UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de Telecomunicación

Evaluación de la influencia del ruido de origen humano en el Parque Natural de la Albufera mediante técnicas de aprendizaje máquina

Trabajo Fin de Máster

Máster Universitario en Ingeniería de Telecomunicación

AUTOR/A: Martínez de Ceano-Vivas, David

Tutor/a: Piñero Sipán, María Gemma

CURSO ACADÉMICO: 2023/2024

Resumen

Este Trabajo Final de Máster tiene como objetivo el entrenamiento y la aplicación de modelos de aprendizaje máquina (Machine Learning, ML) y aprendizaje profundo (Deep Learning, DL) para la detección de eventos sonoros de origen humano en medio natural. Para ello se ha seleccionado información relevante de las grabaciones realizadas durante un año en el Parc de l'Albufera. Con ello se realizó una etapa de evaluación de modelos no supervisados encargados de una primera etapa de agrupación, para facilitar la tarea del etiquetado. Posteriormente se entrenan modelos de aprendizaje profundo y se desarrollará un motor encargado de analizar grabaciones y detectar sonidos de origen humano con objetivo de facilitar el estudio del impacto de estos en la fauna local.

Abstract

This Master's thesis aims to train and apply Machine Learning (ML) and Deep Learning (DL) models for the detection of sound events of human activity in a natural environment. For this purpose, relevant information was selected from the recordings made during a year in the Parc de l'Albufera. An evaluation stage of unsupervised models was carried out, in charge of a first clustering stage, to facilitate the labelling task. Subsequently, deep learning models will be trained and an engine will be developed to analyse recordings and detect sounds of human origin in order to facilitate the study of their impact on local fauna.

RESUMEN EJECUTIVO

La memoria del TFM del Máster Universitario en Ingeniería de Telecomunicación debe desarrollar en el texto los siguientes conceptos, debidamente justificados y discutidos, centrados en el ámbito de la ingeniería de telecomunicación

CONCEPT (ABET)	**CONCEPTO (traducción)**	**¿Cumple? (S/N)**	**¿Dónde? (páginas)**
1. IDENTIFY:	1. IDENTIFICAR:		
1.1. Problem statement and opportunity	1.1. Planteamiento del problema y oportunidad	Si	1
1.2. Constraints (standards, codes, needs, requirements & specifications)	1.2. Toma en consideración de los condicionantes (normas técnicas y regulación, necesidades, requisitos y especificaciones)	Si	4
1.3. Setting of goals	1.3. Establecimiento de objetivos	Si	4
2. FORMULATE:	2. FORMULAR:		
2.1. Creative solution generation (analysis)	2.1. Generación de soluciones creativas (análisis)	Si	5-17
2.2. Evaluation of multiple solutions and decision-making (synthesis)	2.2. Evaluación de múltiples soluciones y toma de decisiones (síntesis)	Si	18-47
3. SOLVE:	3. RESOLVER:		
3.1. Fulfilment of goals	3.1. Evaluación del cumplimiento de objetivos	Si	48
3.2. Overall impact and significance (contributions and practical recommendations)	3.2. Evaluación del impacto global y alcance (contribuciones y recomendaciones prácticas)	Si	48

Índice

Índice de figuras

Índice de tablas

Capítulo 1. Introducción

1.1 Motivación y descripción del problema

La monitorización medioambiental para el análisis de hábitats de especies ha requerido tradicionalmente de un alto grado de participación humana tanto en la adquisición de información como en la supervisión del procedimiento.

Con los avances tecnológicos de los últimos años se han incrementado y mejorado las técnicas de adquisición de audio e imágenes, elevando el volumen de datos disponible. Este incremento de volumen en los datos ha supuesto un reto de cara al procesamiento y a la extracción de valor, es aquí donde entran en juego las técnicas de aprendizaje máquina y aprendizaje profundo, machine learning (ML) y deep learning (DL) en inglés. Estas técnicas permiten extraer información valiosa de grandes volúmenes de datos de forma eficaz, acortando los tiempos necesarios para el procesamiento de datos.

El parque natural de L'Albufera es un entorno natural protegido de gran relevancia biológica, ya que sirve de hábitat para una gran diversidad de aves acuáticas. Este rico ecosistema también presenta actividades humanas, como el cultivo, la caza, la pesca y el turismo. Estas actividades producen contaminación acústica y es esta la que mayor relevancia tiene para este proyecto. Con objetivo de analizar el impacto que tiene esta contaminación en la densidad de población de las aves, se realiza un estudio mediante técnicas de análisis de señal de audio, junto con algoritmos ML no supervisados y DL semi supervisados para procesar las grabaciones realizadas en diferentes puntos de este medio natural y extraer información relevante para los biólogos.

1.2 Estado del arte

1.2.1 Análisis de señales acústicas

El análisis de señales acústicas es un campo que se enfoca en el estudio de los sonidos, utilizando diversas técnicas para extraer información relevante de las grabaciones de audio, lo que permite desde la identificación de especies animales hasta el diagnóstico médico, como el que podría ser la detección de problemas del corazón a partir del ritmo de las pulsaciones.

Existen dos aproximaciones a la hora de realizar esta tarea, dependiendo si el resultado va a tener información temporal o no. Un ejemplo de esta distinción sería por ejemplo si se está analizando un audio extenso y se extrae información a lo largo de este, o sin embargo si se aportan pequeños fragmentos de audio que ya contienen información relevante y se obtiene información de dicho fragmento. Además de tener información temporal, si también se tiene en cuenta que el resultado contenga información de las etiquetas o clases, en este caso estamos ante un problema de detección de eventos sonoros, o como se conoce en inglés Sound Event Detection[1].

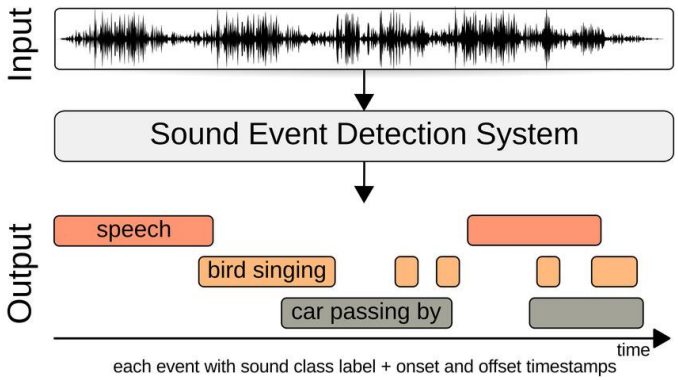

Figura 1. Diagrama de detección de eventos sonoros [1]

La detección de eventos sonoros en entornos naturales supone un gran desafío ya en estos entornos podemos observar una gran diversidad de sonidos, tanto en su origen como en la forma en la que estos aparecen, siendo totalmente impredecibles y presentando duración en intensidad variable, lo que dificulta enormemente el proceso de análisis.

Para llevar a cabo el análisis de señales acústicas, se utilizan diversas técnicas, como la transformada de Fourier o los coeficientes MFCC. Estas técnicas permiten extraer características relevantes de las señales de audio, que posteriormente pueden ser analizadas y clasificadas utilizando algoritmos de machine learning.

1.2.2 Técnicas de aprendizaje máquina

En el procesamiento y la clasificación de señales de audio, las técnicas de machine learning juegan un papel fundamental, ya que permiten la identificación de patrones complejos dentro de los datos acústicos. Estas técnicas engloban una serie de algoritmos y modelos que permiten a los sistemas informáticos aprender a partir de datos y mejorar su desempeño en tareas específicas sin la necesidad de ser programados explícitamente para cada uso particular.

A la hora de hablar estas técnicas podemos hacer una primera distinción en dos tipos de aprendizaje, aunque existen técnicas intermedias.

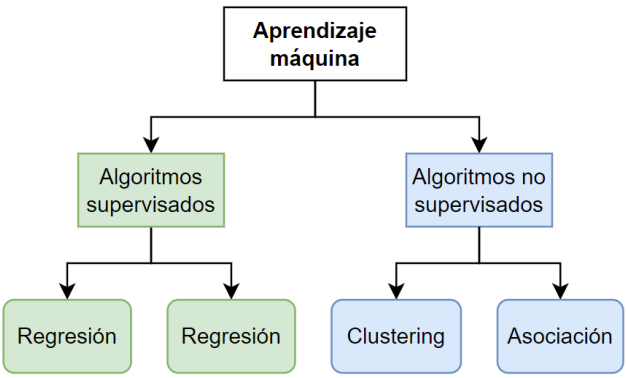

Figura 2. Esquema aprendizaje máquina

Aprendizaje no supervisado: El aprendizaje no supervisado es un tipo de aprendizaje automático que busca patrones no detectados previamente en un conjunto de datos sin etiquetas preexistentes y con un mínimo de supervisión humana [2]. Esta rama del aprendizaje automático trata de utilizar datos que no tienen etiquetas o salidas predefinidas, por lo que el modelo debe encontrar patrones que relacionan los datos entre sí. El objetivo de estos modelos es el de resolver problemas de agrupamiento o de reducción de dimensionalidad.

Aprendizaje supervisado: Esta rama, al contrario que el aprendizaje no supervisado, se caracteriza por proporcionar un conjunto de datos de entrenamiento que incluye directamente la información de salida, es decir, las etiquetas, de forma que el modelo es capaz de ajustarse y aprender los patrones necesarios que permitan relacionar las entradas con las etiquetas ya aportadas. El objetivo del aprendizaje supervisado es el de construir un sistema artificial que sea capaz de aprender a mapear las relaciones entre la entrada y la salida [3]. Algunos de los problemas que se intentan resolver con estas técnicas son de clasificación o de regresión.

1.2.3 Programación y automatización

Para poder aplicar tanto técnicas de análisis acústico como de aprendizaje máquina es necesario encontrar una herramienta que nos facilite tanto el desarrollo como la implantación de dichas técnicas. Existen diversas herramientas y lenguajes de programación que podrían cumplir con estos requisitos; sin embargo, este proyecto se basará en Python.

Python se ha consolidado como uno de los lenguajes de programación más populares en el campo de la ciencia de datos y el aprendizaje automático. Las grandes ventajas que podemos encontrar en este lenguaje son:

- **Sintaxis sencilla y legible**: Python se caracteriza por su sintaxis clara y accesible, muy similar al lenguaje natural, por lo que es ideal para implementaciones tanto para expertos como para principiantes, facilitando su utilización y distribución en sectores con poca base en programación. En el contexto del análisis de señales de audio, esta legibilidad es crucial para asegurar que los algoritmos desarrollados puedan ser fácilmente entendidos, modificados y reutilizados.
- **Variedad de librerías**: El ecosistema de python dispone de una extensa cantidad de librerías que ofrecen herramientas para prácticamente cualquier campo de interés, facilitando tareas y reduciendo la necesidad de desarrollo de funciones propias, algunos ejemplos de las librerías que se utilizarán en este proyecto:
    - Numpy: Operaciones numéricas sobre matrices y vectores.
    - Pandas: Manipulación de datos estructurados.
    - Librosa: Especializada en el análisis de audio [5].
    - Scikit-Learn: Implementaciones de algoritmos de aprendizaje automático. [6].
    - Tensorflow y Pytorch: Implementaciones de algoritmos de aprendizaje automático.
    - Matplotlib: Representaciones gráficas. [4]
- **Gran comunidad**: Python cuenta con una comunidad global muy activa y dispone de gran cantidad de recursos, formación online y documentación, lo que facilita el aprendizaje y adquisición de conocimiento.

En resumen, Python no solo ofrece las herramientas técnicas necesarias para llevar a cabo la automatización del proceso de análisis de audio y la implantación de los modelos de aprendizaje automático, sino que, al ser un lenguaje amigable y fácil de entender, también facilitará la comprensión de los procesos seguidos durante el desarrollo de este trabajo.

1.3 Objetivos

El objetivo principal de este proyecto de análisis de las grabaciones obtenidas del Parc de l'Albufera, para detectar, clasificar y medir la densidad de sonidos de origen no humano mediante técnicas de aprendizaje máquina con el objetivo de facilitar los estudios medioambientales del impacto de estos sonidos en la fauna local.

Para ello será necesario diseñar un motor, basado en el lenguaje de programación Python, capaz de analizar y reportar la información obtenida de las grabaciones de forma que se pueda realizar el posterior análisis de los datos extraídos, un proceso que requiere tanto de investigación y documentación como desarrollo de los algoritmos necesarios.

Por lo tanto, se puede definir una serie de objetivos secundarios que llevarán al desarrollo final de dicho motor:

1. Análisis exploratorio de los datos disponibles y procesado de señales de audio.
2. Separación y clasificación mediante técnicas de machine learning no supervisado, evaluando y validando diferentes modelos.
3. Diseño y entrenamiento de redes neuronales para la detección.
4. Evaluación de resultados.

Condicionantes.

Para el objeto de este proyecto no se conocen condicionantes, normas o especificaciones que marquen las pautas a seguir para su desarrollo. El único condicionante que podría estar asociado al proyecto es la metodología de grabación y obtención de la base de datos, así como las especificaciones técnicas de los micrófonos que puedan afectar al formato de los datos, como puede ser el formato de archivo y la frecuencia de muestreo de la grabación. Aunque su impacto es mínimo ya que los propios ficheros contienen la información necesaria para procesar adecuadamente las grabaciones. Esta información técnica se menciona en el punto dedicado al origen de los datos.

Capítulo 2. Metodología empleada

En este apartado se describe toda la teoría que ha formado parte del desarrollo de este trabajo, explicando en primera instancia todos los conceptos necesarios en el análisis acústico, para después introducir los diferentes algoritmos y modelos de aprendizaje máquina que serán relevantes durante el desarrollo.

Para una reproducción detallada de los experimentos y una exploración más profunda de los resultados, se invita al lector a consultar los notebooks de Jupyter alojados en el siguiente repositorio: https://github.com/Damarde/TFM_Clasificador_Sonidos_con_ML

Estos notebooks contienen el código fuente, las salidas que se pueden observar en este documento.

Generación de soluciones creativas (Análisis)

El planteamiento para dar solución al problema, tal y como se describe en este capítulo, pasa por el estudio de las técnicas de análisis de audio y la selección de las características, luego escoger las más adecuadas para cada modelo de aprendizaje máquina. El siguiente paso es comprobar qué modelo es el más adecuado, si los relacionados con el aprendizaje no supervisado y su ventaja de no necesitar un etiquetado previo, o una combinación de la aplicación de estos modelos, para facilitar el etiquetado, y las técnicas de aprendizaje profundo que ofrecen una mayor precisión.

2.1 Análisis de audio

Las señales de audio presentan gran cantidad de información que en primera instancia es difícil de observar e interpretar. Estas señales se componen de un conjunto de ondas a diferentes frecuencias superpuestas entre sí.

Para explorar las técnicas que se van a aplicar vamos a analizarlas utilizando un fragmento de grabación obtenido de los micrófonos de la albufera. Como se puede observar en la figura 3, es difícil identificar cualquier tipo de información más allá de picos de amplitud en ciertos momentos, pero ¿qué significan estos picos?

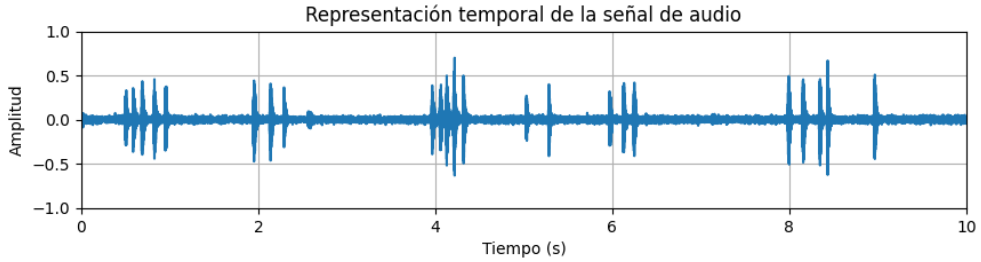

Figura 3. Representación de fragmento de audio

Para dar respuesta a esta pregunta entran las técnicas de procesado de señal que facilitarán la extracción de información.

2.1.1 Transformada de Fourier

Una forma de poder observar en mayor detalle la información contenida dentro de una señal de audio es separando las señales superpuestas por frecuencias, esto se puede conseguir mediante la aplicación de la Transformada de Fourier (TF). La Transformada de Fourier es una fórmula matemática que transforma una señal en el dominio del tiempo a una señal en el dominio de la frecuencia, facilitando de esta forma el análisis del contenido espectral de la señal.

En nuestro caso de aplicación contamos con una señal discreta por lo que se utiliza una variación, la Transformada Discreta de Fourier, o Discrete Fourier Transform (DFT) en inglés. Esta fórmula se representa de la siguiente manera:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi k}{N} n} \quad \textbf{(2.1)}$$

donde x[n] es la señal muestreada tal que x[n] = x(n·Ts) siendo fs = 1/Ts la frecuencia de muestreo, N el tamaño de la DFT y k el índice tal que k=0,…, N-1. Si la señal x[n] es real, su contenido frecuencial puede ser visualizado mediante la mitad de los valores de X[k], k=0,…, N/2. El rango de frecuencia a representar será entonces f = (0, k/N, …, k/(N/2)) · fs, en Hz.

El uso de esta ecuación amplia la información, pero sigue sin aporta la suficiente como para saber exactamente que está sucediendo, tal y como se puede observar en la siguiente figura.

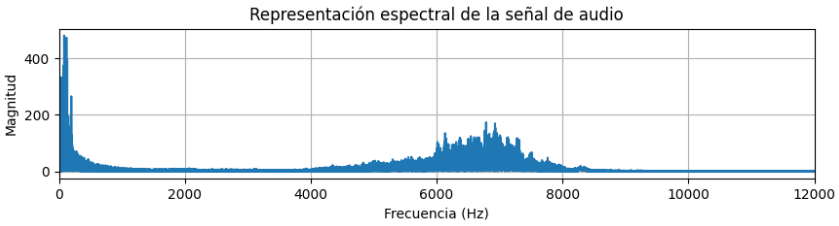

Figura 4. Resultado de la DFT de la señal

Algo que sí se puede distinguir en este caso específico es que la mayor parte de la señal se concentra en frecuencias bajas, y que hay cierta información presente entre los 4000 y los 8000 Hz, aunque dada la evolución temporal esto tampoco refleja la realidad con precisión, ya que es imposible discernir la evolución temporal de dichas señales.

Es por ello por lo que se define otra versión de la TF para obtener información de mayor precisión, más apta para el uso en señales no estacionarias, como son las señales digitales de audio, esta es la Transformada de Fourier de Tiempo Corto, o Short Time Fourier Transform (STFT) [8]. Su uso se basa en la división de la señal en segmentos cortos para calcular la TF para cada segmento, permitiendo observar la evolución en el contenido espectral de la señal a lo largo del tiempo. Su ecuación es la siguiente:

$$S[k, m] = \sum_{n=0}^{N-1} w[n] x[n + m \cdot L] e^{-\frac{j2\pi k}{N} n} \quad \textbf{(2.2)}$$

Siendo w[n] la ventana, cuyo tamaño influye tanto en el coste computacional como en la resolución en el eje frecuencia, y L el salto de muestras que permite ir recorriendo la señal de entrada a lo largo del tiempo. El resultado es una matriz S[k,m] que contiene información de x[n] en frecuencia y tiempo.

2.1.2 Espectrograma

A partir de la STFT se puede obtener el espectrograma, una representación visual de la frecuencia en función del tiempo de una señal, es decir, muestra la evolución del contenido espectral a lo largo del tiempo. Esta representación se obtiene a partir del módulo de la STFT, y se suele representar en dB.

En la siguiente figura se puede observar la representación del espectrograma de la señal mostrada hasta ahora:

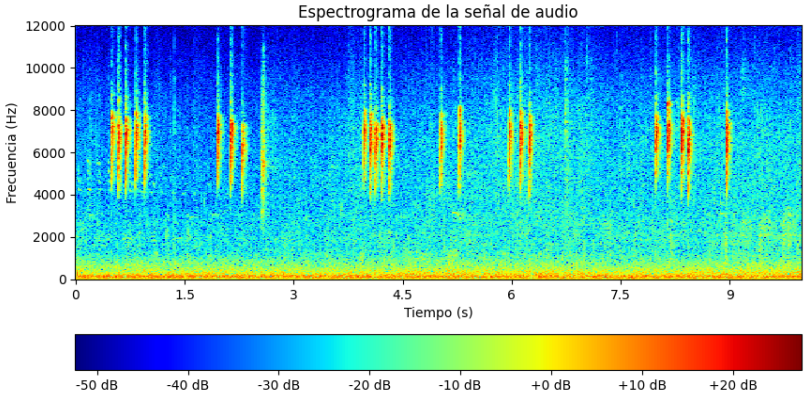

Figura 5. Representación del espectrograma de la señal

Gracias al espectrograma ahora podemos observar cómo evoluciona la señal tanto en el tiempo y según frecuencia, obteniendo de esta manera más información.

2.1.3 Espectrograma de Mel

El espectrograma de Mel es una forma avanzada, comúnmente utilizada en el campo del análisis de audio, para visualizar una señal en el eje tiempo-frecuencia. A diferencia del espectrograma tradicional, el espectrograma de Mel ajusta el eje de frecuencias según la escala de Mel, la cual refleja cómo el oído humano percibe los sonidos.

El oído humano no percibe las frecuencias de manera lineal. A bajas frecuencias es capaz de distinguir cambios en frecuencia con mayor precisión, mientras que, a altas frecuencias, las diferencias perceptuales son menores, tal y como se ilustra en la figura 6.

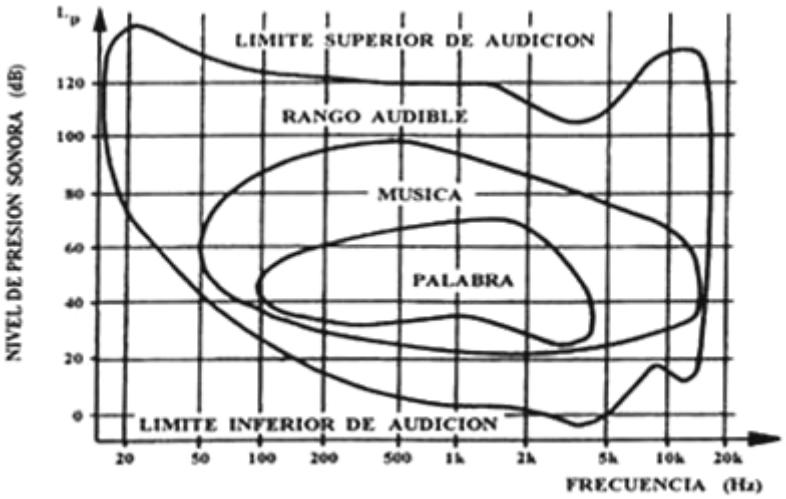

Figura 6. Rangos de audición en humanos. [9]

Para ajustar la representación de la señal a nuestra percepción auditiva, se utiliza la escala de Mel, calculada mediante la siguiente ecuación [12]:

$$f_{mel} = 2595\, log_{10}(1 + \frac{f}{700}) \quad \textbf{(2.3)}$$

El proceso para calcular el espectrograma de Mel es similar al que se seguiría para una FFT, pero teniendo un paso intermedio adicional que involucra los filtros de Mel. Para ello primero se fragmenta la señal en segmentos más pequeños mediante un proceso de enventanado. Se obtiene la transformada rápida de Fourier para cada uno de los segmentos, que, como se comentó anteriormente pasa la señal al dominio espectral.

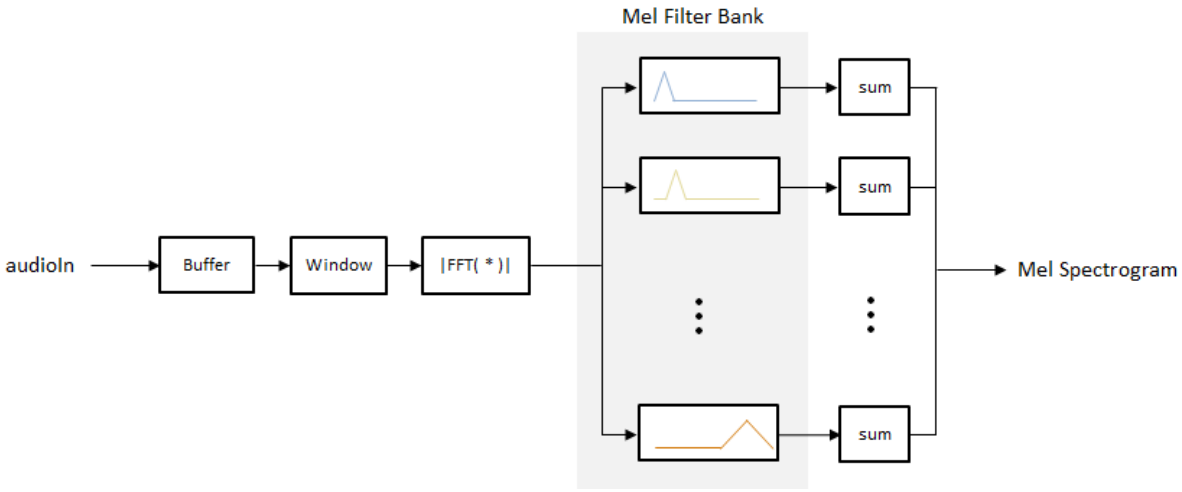

Figura 7. Esquema cálculo espectrograma Mel [10]

Una vez obtenida la representación en frecuencia se aplica un banco de filtros definido por el número de bandas de Mel. Este banco de filtros está diseñado utilizando filtros de banda estrecha que se distribuyen simulando la escala de Mel. Son filtros típicamente de forma triangular y están espaciados a lo largo de dicha escala, cubriendo un rango específico de frecuencias.

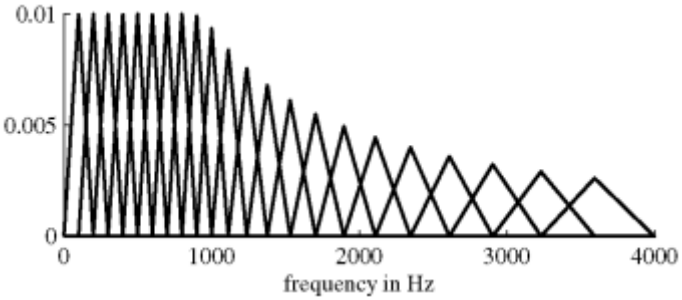

Figura 8. Representación de banco de filtros Mel [11]

Tras este banco de filtros se suma la energía obtenida por cada banco para reconstruir la señal, reflejando la distribución de energía en cada frecuencia, dando como resultado un espectrograma en el que tiene mayor presencia las señales a bajas frecuencias.

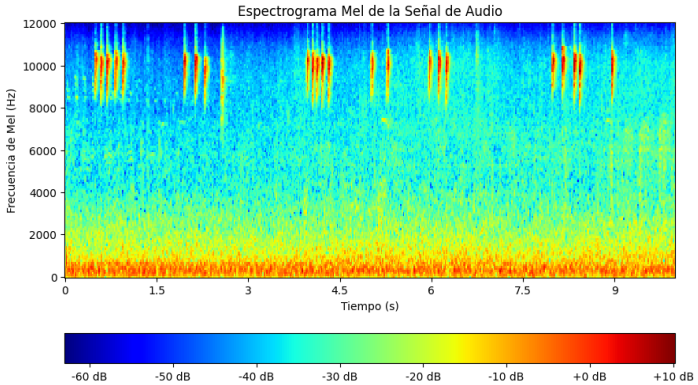

Figura 9. Representación del Espectrograma Mel de la señal

2.2 Características

Para poder entrenar los modelos de machine learning, es fundamental transformar las señales a estudiar en información que los modelos puedan interpretar. Esta información, que captura los aspectos más relevantes y representativos de las señales, se conoce como características o features. La selección de las características idóneas es un proceso que requiere un conocimiento profundo tanto del problema a abordar como de la naturaleza de las características en sí mismas. Es fundamental identificar aquellas características que aporten información valiosa y evitar la inclusión de aquellas que presenten una alta correlación entre sí, con el objetivo de maximizar la cantidad de información útil disponible para los modelos.

A continuación, se describen las principales características acústicas que se han tenido en cuenta para este proceso, teniendo en consideración su información y también la su disponibilidad en librerías de python que faciliten su implantación. Para ello se ha utilizado la librería Librosa, ampliamente usada en el análisis y procesamiento de señales de audio, y que contiene gran cantidad de operaciones para la extracción de características [5].

2.2.1 Flujo espectral

El flujo espectral es una medida que cuantifica la variación en el contenido espectral de una señal a lo largo del tiempo. En esencia mide cómo cambia la energía en las frecuencias entre segmentos temporales consecutivos.

El proceso para calcularlo pasa por obtener el espectro de potencia a partir de la STFT, obteniendo así la energía de la muestra en cada tiempo y frecuencia. Entonces se calcula la diferencia entre tramas consecutivas. El resultado de este flujo se puede calcular con la siguiente expresión [13]:

$$\boldsymbol{Flux(m)} = \sqrt{\sum_f \big(\boldsymbol{S(k,m)} - \boldsymbol{S(k,m-1)}\big)^2} \tag{2.4}$$

Obteniendo una evolución de la variación del espectro en el tiempo, tal y como puede observarse en la siguiente gráfica.

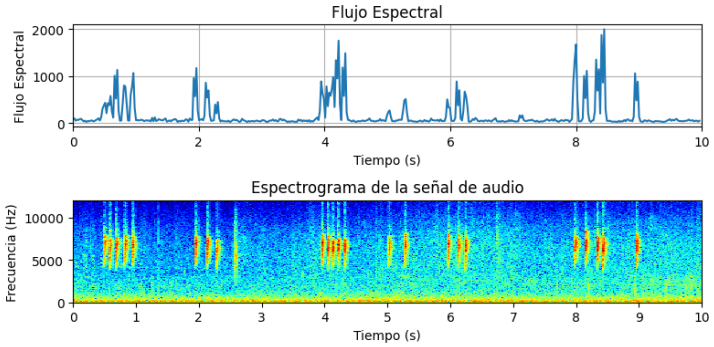

Figura 10. Flujo espectral y espectrograma de la señal.

2.2.2 Centroide espectral

El centroide espectral es una medida acústica que se utiliza para indicar el "centro de masa" espectral. Es una característica espectral comúnmente utilizada en el análisis de audio, ya que sirve de base para calcular otras características relacionadas con el mismo.

El centroide espectral a lo largo del tiempo puede calcularse con la siguiente ecuación:

$$\boldsymbol{C(m)} = \frac{\sum_f \boldsymbol{f} * \boldsymbol{S(k,m)}}{\sum_f \boldsymbol{S(k,m)}} \tag{2.5}$$

Esta medida indica cuán grave o aguda suena una señal. Valores bajos significa que la energía espectral está concentrada en frecuencias más bajas, lo que supone un sonido más grave. Por contra, valores más altos indican un sonido más agudo

Podemos encontrar un método en python que nos permite calcular de forma agil el centroide espectral de una señal dentro de la librería Librosa*: librosa.feature.spectral_centroid()*[5]

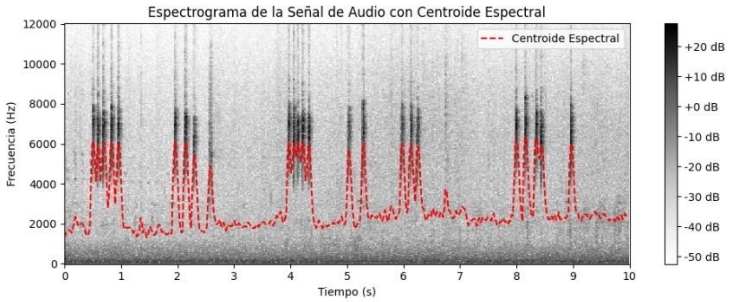

Figura 11. Centroide Espectral superpuesto al espectrograma de la señal.

2.2.3 Ancho de banda espectral

El ancho de banda espectral, o dispersión espectral, es una medida que describe cómo se concentra la energía espectral de una señal de audio alrededor de su centroide espectral. Este valor no es un reflejo de la cantidad de energía, sino que es forma de cuantificar cómo se distribuye a lo largo ancho de banda de la señal. Matemáticamente la dispersión espectral en un instante de tiempo se define como la desviación estándar de las frecuencias alrededor del centroide, expresada en la siguiente ecuación:

$$\sigma(m) = \sqrt{\frac{\sum_f (f - C(m))^2 S(k,m)}{\sum_f S(k,m)}} \quad \textbf{(2.6)}$$

Un valor bajo indica que la energía espectral está concentrada, o cerca del centroide, mientras que un valor elevado indica que la energía está más dispersa o distribuida, esto podemos observarlo en la figura 12.

De forma similar al centroide espectral, es posible calcular directamente el ancho de banda espectral utilizando la librería librosa con la función: *librosa.feature.spectral_bandwidth()* [5].

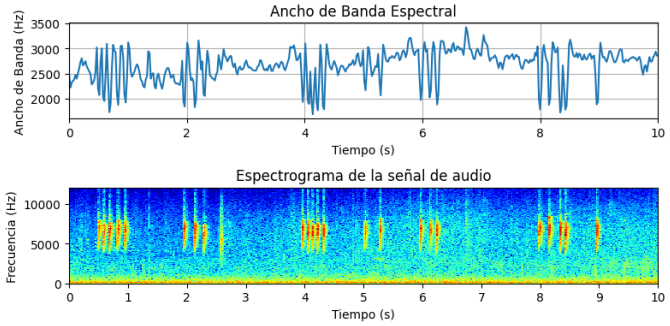

Figura 12. Ancho de banda espectral y espectrograma de la señal.

2.2.4 Planitud espectral

La planitud espectral, o espectral flatness, mide cuan plana o uniforme es la distribución del espectro de una señal en el dominio de la frecuencia.

Se calcula midiendo la proporción entre la media geométrica y la media aritmética de la magnitud del espectro de una señal tal y como se describe en la siguiente ecuación:

$$SF(m) = \frac{MG(S(k,m))}{MA(S(k,m))} \quad \textbf{(2.7)}$$

La función de librosa encargada del cálculo en python es: *librosa.feature.spectral_flatness()* [5].

Un valor alto indica una señal con una distribución más uniforme a lo largo del espectro, lo que se puede traducir en una señal más ruidosa, mientras que al contrato un valor bajo indica componentes con tonos fuertes, como por ejemplo notas musicales con armónicos definidos.

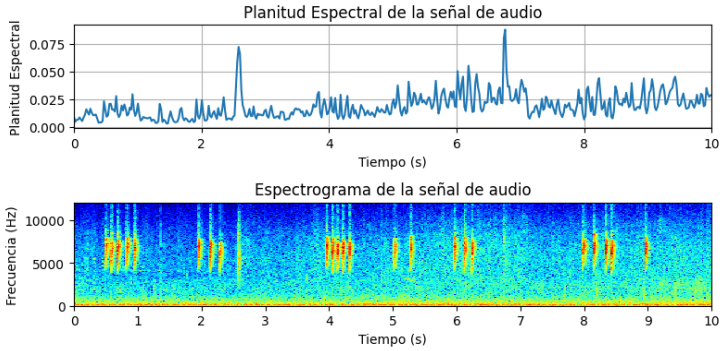

Figura 13. Planitud espectral y espectrograma de la señal.

2.2.5 Asimetría espectral

La asimetría espectral (también conocida como skewness espectral) es una medida que describe la simetría de la distribución del espectro de frecuencias alrededor de su centroide. Específicamente, mide el grado de asimetría de la distribución espectral, lo que proporciona información sobre la tendencia de la energía espectral hacia frecuencias más altas o bajas. Es posible calcular la asimetría espectral en un instante de tiempo mediante la siguiente ecuación:

$$\gamma(m) = \frac{\sum_f \left(\frac{f - C(m)}{\sigma(m)}\right)^3 S(k,m)}{\sum_f S(k,m)} \quad \textbf{(2.8)}$$

Donde:

- γ(m) es la asimetría espectral en la trama o fragmento m.
- C(m) es el centroide espectral en el tiempo t.
- σ(m) es la dispersión espectral (o ancho de banda espectral) en el tiempo t.
- S(k,m) es la magnitud o energía del espectro en la frecuencia f=k/N·fs y en el tiempo t.

Valores positivos indican que la energía espectral tiende a concentrarse en frecuencias más altas que el centroide, mientras que valores negativos indican una concentración a frecuencias más bajas. Un valor cercano a cero es indicativo de una concentración simétrica.

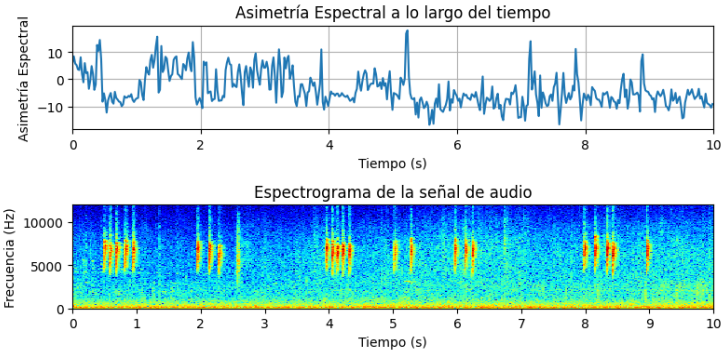

Figura 14. Asimetría espectral y espectrograma de la señal.

2.2.6 Roll-off espectral

El roll-off espectral es una medida que indica el punto en el espectro de frecuencias que contiene la mayor parte del porcentaje, normalmente entre un 85% y un 95%, de la energía total de la señal está contenida por debajo de dicho punto. Esta puede calcularse con la siguiente expresión:

$$R(m) = min\left\{f: \sum_{k=0}^{f} S(k,m) \geq P \sum_{k'=0}^{f-1} S(k',m)\right\} \quad \textbf{(2.9)}$$

Donde:

- R(m) es la frecuencia de roll-off espectral en el tiempo t.
- S(k,m) es la magnitud o energía del espectro en la frecuencia f=k/N·fs y en el tiempo t.
- P es el porcentaje de energía acumulada, que normalmente se establece en 85% o 95%.
- f es la frecuencia máxima representada en el espectro.

Librosa dispone de una función específica para el cálculo del roll-off espectral, esta se describe de la siguiente manera: *librosa.feature.spectral_rolloff()* [5].

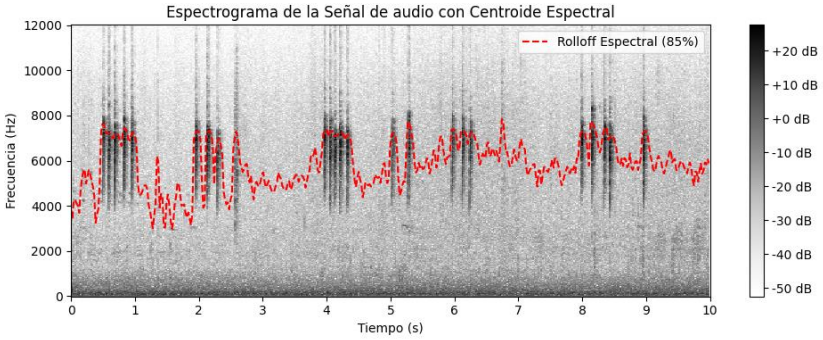

Figura 15. Roll-off espectral superpuesto al espectrograma de la señal.

2.2.7 Contraste espectral

El contraste espectral es una característica acústica que mide la diferencia en amplitud entre picos y valles en el espectro de una señal de audio. Es útil para caracterizar la textura de una señal e identificar cuando una señal presenta gran variedad de frecuencias o una distribución más plana y uniforme. La expresión que define el contraste espectral es la siguiente:

$$Ct(m) = \frac{max(S(k,m)) - min(S(k,m))}{mediana(S(k,m))} \quad \textbf{(2.10)}$$

Su cálculo está contemplado en la librería librosa mediante: *librosa.feature.spectral_contrast()* [5].

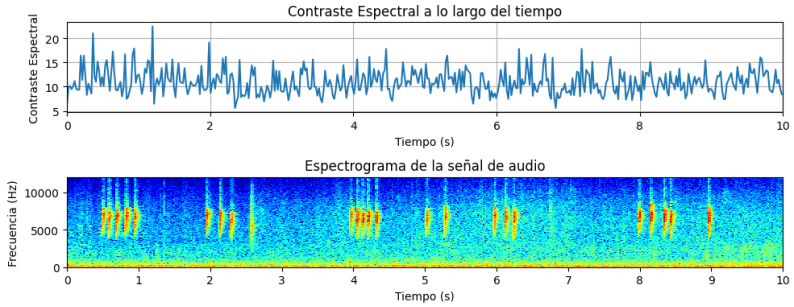

Figura 16. Contraste espectral y espectrograma de la señal.

2.2.8 Coeficientes cepstrales de frecuencia Mel (MFCC)

Como se ha mencionado con anterioridad, el espectrograma de Mel nos ofrece una representación visual del contenido frecuencial de una señal de audio según la capacidad auditiva del ser humano, pero para que una máquina pueda analizar de forma eficiente esta información es necesario una forma más compacta y numérica de capturar esta información. Aquí es donde entran en juego los coeficientes cepstrales de la frecuencia mel (MFCC). Estos coeficientes proporcionan una

descripción más detallada y robusta de la envolvente espectral, condensando la información clave en un conjunto de valores numéricos. Siendo una técnica muy extendida en el análisis de sonidos, especialmente, aunque no limitado, en el análisis del habla.

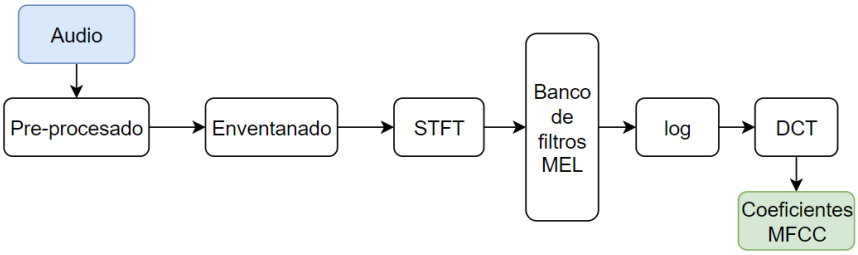

Figura 17. Proceso de obtención de coeficientes MFCC

Igual que para las otras características espectrales, librosa nos proporciona una función específica para su cálculo: *librosa.feature.mfcc()* [5].

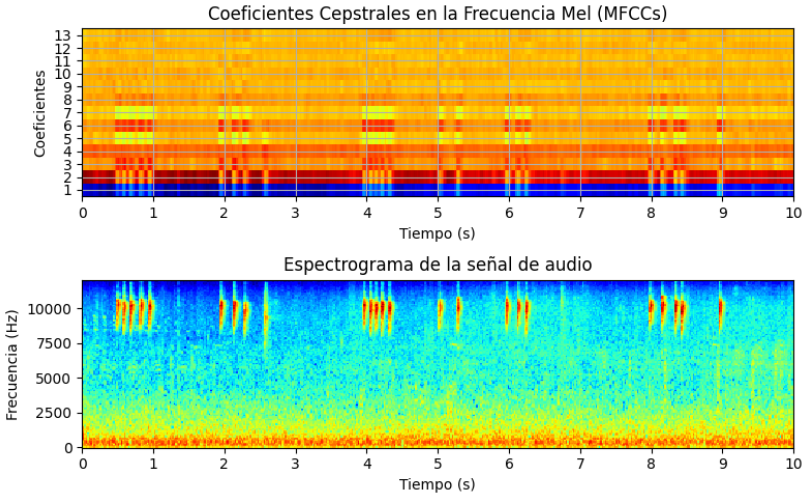

Figura 18. MFCC y espectrograma Mel de la señal.

2.2.9 Delta MFCC

Los coeficientes delta de los MFCCs (también conocidos como deltas de MFCC) proporcionan información sobre la dinámica temporal de los coeficientes cepstrales en la frecuencia Mel. Mientras que los MFCCs capturan la estructura espectral estática de la señal de audio, los coeficientes delta miden cómo cambia a lo largo del tiempo. Esto ayuda a capturar la dinámica y la evolución temporal de la señal de audio.

$$\boldsymbol{\Delta MFCC(j,m) = MFCC(j,m) - MFCC(j,m-1)} \quad \textbf{(2.11)}$$

Donde:

- MFCC(j,m) es el valor del coeficiente cepstral j en el tiempo t.

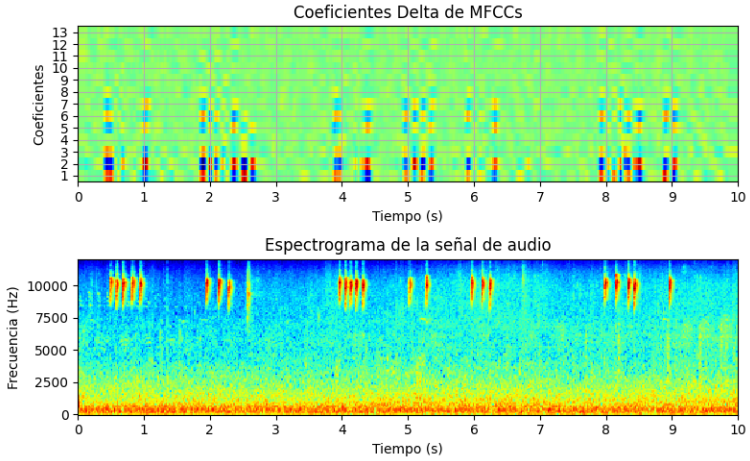

Figura 19. Coeficientes delta MFCC y espectrograma Mel de la señal.

2.3 Otras técnicas de procesado

Una vez obtenidas las características es importante tener en consideración que no todos los valores se trabajan en la misma escala, por lo que es importante aplicar un escalado. Esto implica transformar los datos para que todas las características se encuentren en un rango similar o tengan una distribución comparable. Esto es importante dado que muchos algoritmos de machine learning, como los basados en distancias o las redes neuronales son sensibles a la magnitud de las características. Tipos comunes de escalado:

- Normalización (Min-Max Scaling): Escala las características a un rango específico, generalmente [0, 1].
- Estandarización (Standard Scaling): Transforma las características para que tengan media 0 y desviación estándar 1.
- Escalado Robusto (Robust Scaling): Usa estadísticas como la mediana y los rangos intercuartílicos para escalar características, haciéndolo menos sensible a outliers.

2.4 Aprendizaje Máquina

El aprendizaje máquina es una disciplina dentro de la inteligencia artificial que permite a los sistemas informáticos aprender y mejorar a partir de una serie de datos de entrada, sin una necesidad explícita de ser programados para la tarea específica.

Estos modelos de aprendizaje, en vez de seguir un conjunto de instrucciones fijas, identifican patrones, realizan predicciones o toman decisiones en función de estas, basado en los datos de entrada. Este enfoque ha demostrado ser de extrema utilidad en gran variedad de aplicaciones modernas, desde la recomendación de productos en plataformas de comercio hasta el diagnóstico médico asistido.

En general el proceso de funcionamiento de estos algoritmos sigue el mismo patrón:

- Recolección de datos. Se recopilan los datos y se organizan de forma que puedan ser procesados informáticamente.
- Preprocesamiento de datos: Se hace un tratamiento previo, limpiando y transformando los datos en caso de ser necesario.
- División de datos. A la hora de entrenar los datos es importante dividir el conjunto de datos en conjunto de entrenamiento y conjunto de prueba, también es común apartar un

nuevo conjunto que sería utilizado para la validación. Esto se realiza para evitar probar y validar el modelo sobre datos que hayan sido utilizados durante el entrenamiento, ya que estos serían conocidos.

- Selección del modelo. Es la parte más importante, la elección del modelo es un proceso clave, que requiere conocer el problema, ya que no todos los modelos son efectivos para todos los escenarios.
- Entrenamiento. En esta etapa se entrena el modelo con los datos de entrenamiento, ajustando los parámetros de este.
- Evaluación. Dependiendo del modelo empleado existen una serie de métricas que pueden servir para medir el rendimiento del modelo utilizando datos no conocidos por este, por ejemplo, los de validación mencionados anteriormente.
- Ajuste y optimización. Tras el proceso de evaluación es posible que se quieran hacer modificaciones y ajustes, cambiando características o hiper parámetros. Este proceso implica repetir las etapas de entrenamiento y evaluación y puede repetirse las veces que sean necesarias hasta obtener un rendimiento deseado.

2.4.1 Métodos no supervisados

Como ya se ha mencionado anteriormente, los métodos no supervisados se utilizan cuando los datos disponibles no están etiquetados. Este es el caso más común en los datos, especialmente cuando se trata de grabaciones en medio natural y es uno de los principales problemas que nos encontramos en este proyecto. En nuestro caso las grabaciones no han sido tratadas por lo que no hay ninguna etiqueta.

Clustering

Las técnicas de clustering o agrupamiento, son técnicas clave en el aprendizaje no supervisado. Como su nombre indica el objetivo de estos algoritmos es el de agrupar los datos que compartan similitudes, esto se hace detectando estructuras o patrones ocultos en los datos y es útil para segmentar grandes volúmenes de datos.

Alguna de las técnicas que se exploran en este proyecto son las siguientes:

- K-Means: Se trata de un algoritmo que divide el conjunto de datos en un numero predefinido de clústeres (k). El objetivo es minimizar la suma de las distancias al cuadrado que hay entre la muestra y el centroide del clúster al que pertenecen.

  Se trata de un proceso iterativo en el que se asignan una serie de centroides y se buscan los elementos que están más cercanos a él. Una vez hecha esta agrupación se recalcula el centroide y se vuelven a asignar los grupos según la distancia al centroide. Este proceso se repite hasta que la asignación y los centroides convergen. Es un proceso eficiente a nivel computacional y especialmente adecuado para grupos de datos de tamaño similar.[14].
- DBSCAN: Algoritmo basado en densidad que agrupa los puntos de datos cercanos y densamente conectados, identificando puntos de baja densidad como ruido. Este algoritmo requiere de dos parámetros: “eps”, que define el radio de búsqueda de vecinos, y “min_samples”, que hace referencia al número de mínimo de puntos necesarios para formar un clúster. Al contrario de k-Means, no requiere de predefinir el número de clústeres, y es capaz de detectar agrupaciones de forma arbitraria y clasificar los puntos que no cumplen con el umbral de densidad como ruido.[15]
- GMM: Se trata de modelos probabilísticos que parten de la suposición de que los datos son una mezcla de distribuciones gaussianas. Cada componente gaussiana en el modelo representa un clúster. GMM es flexible y puede modelar clústeres elípticos y de tamaños

variados, y proporciona una estimación probabilística de las pertenencias a clusters. Sin embargo, puede ser sensible a la inicialización y puede converger a óptimos locales, además de requerir la especificación del número de componentes a priori.

Reducción de dimensionalidad

Otra de las técnicas más conocidas es la Reducción de dimensionalidad. Son comúnmente usadas en conjunto con el resto de modelos de machine learning, ya que permiten transformar grandes volúmenes de datos a dimensiones más reducidas manteniendo la mayor cantidad posible de información.

Algunos de los modelos de reducción de dimensionalidad que se han valorado en este proyecto son:

- Análisis de componentes principales (PCA en inglés): PCA es ampliamente utilizado debido a su simplicidad y efectividad, especialmente cuando se busca reducir la complejidad de los datos antes de aplicar modelos de análisis o clasificación. Es una técnica que transforma un conjunto de datos de alta dimensión en un menor al identificar las direcciones que captan la mayor varianza posible en los datos, proyectándolos sobre los nuevos ejes y maximizando la información retenida.

- t-SNE (t-distributed Stochastic Neighbor Embedding) es un método no lineal que se utiliza principalmente para la visualización de datos de alta dimensionalidad. A diferencia de PCA, se centra en preservar la estructura local de los datos, de manera que los puntos son similares permanezcan cerca en el nuevo espacio reducido.
  Presenta un mayor coste computacional y es más utilizado para realizar análisis exploratorios, no siendo muy recomendado previo a modelos de clasificación.

2.4.2 Métodos supervisados

En los métodos supervisados, los datos de entrenamiento vienen etiquetados, es decir, cada entrada tiene asociada una salida conocida. Estos modelos son capaces de aprender patrones a partir de los datos etiquetados, lo que les permite realizar predicciones sobre nuevas entradas desconocidas. Dependiendo del tipo de problema, los modelos supervisados pueden abordar tareas de clasificación o regresión.

- Clasificación: El objetivo es asignar a cada entrada una etiqueta discreta, es decir, se trata de predecir a qué categoría o clase pertenece un nuevo dato. Ejemplos comunes incluyen la identificación de imágenes o la clasificación de correos electrónicos como spam o no spam.
- Regresión: En este caso, el objetivo es predecir un valor continúo basado en las características de entrada. Un ejemplo típico es la predicción de precios de viviendas o la estimación de la temperatura en función de diversas variables.

Dentro de este marco, nos centraremos en un tipo específico de modelo de clasificación: las redes neuronales convolucionales, que son particularmente efectivas para el procesamiento de datos con estructura espacial, como las imágenes o, en el contexto de este proyecto, las secuencias de audio, así como para su clasificación.

Redes neuronales convolucionales

Las redes neuronales son una clase de modelos supervisados que pertenecen al campo del aprendizaje profundo, una subdisciplina dentro del aprendizaje automático. Estas redes están formadas por múltiples capas de neuronas interconectadas que imitan el funcionamiento del cerebro humano, lo que les permite aprender representaciones complejas y abstractas a partir de los datos de entrada. A través del proceso de entrenamiento, las redes neuronales ajustan sus parámetros internos para captar patrones ocultos en los datos, lo que les permite realizar predicciones precisas en tareas complejas. En la figura 20 podemos ver un ejemplo de red neuronal convolucional centrada en la separación de fuentes de sonidos.

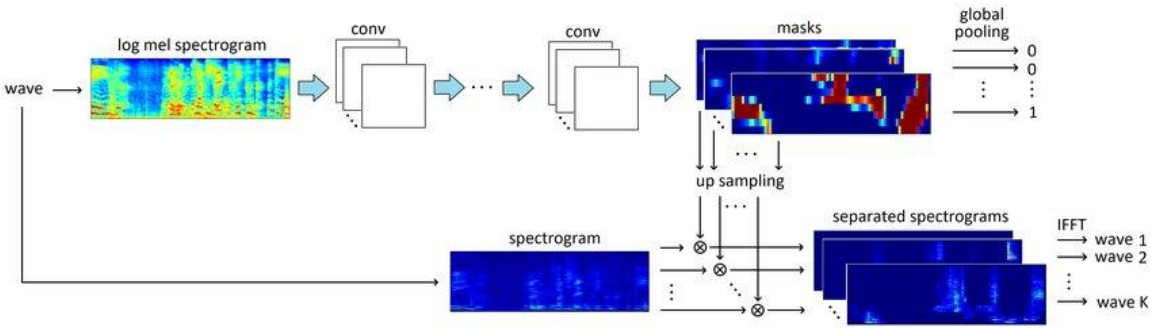

Figura 20. Esquema red neuronal convolucional. [17]

Aunque las redes neuronales son ampliamente reconocidas por su eficacia en el procesamiento de imágenes, también han demostrado ser extremadamente versátiles en otras aplicaciones, como el análisis de señales de audio, procesamiento de lenguaje natural y series temporales. En este proyecto, utilizaremos redes neuronales convolucionales (CNNs) para aprender y hacer predicciones a partir de la información contenida en los espectrogramas, que son representaciones visuales de señales de audio. Al analizar los espectrogramas, estas redes pueden capturar características temporales y frecuenciales cruciales, lo que las hace especialmente adecuadas para la clasificación y análisis de sonidos en entornos naturales.

Capítulo 3. Clasificador de sonidos de origen humano.

En esta sección se detalla el proceso seguido para el desarrollo del clasificador de sonidos de origen humano. Comienza con una puesta en contexto del proyecto, luego se divide el desarrollo en diferentes etapas, cada una con sus correspondientes metodologías y resultados.

La primera etapa comienza con una exploración de los datos disponibles. En esta etapa se incluye la evaluación y selección tanto de modelos de reducción de dimensionalidad como de técnicas de agrupamiento, analizando métricas y resultados.

La segunda etapa se centra en el desarrollo y evaluación de una red neuronal convolucional. Con apoyo de los resultados de la primera fase se realizarán las pruebas y ajustes necesarios para el entrenamiento y validación de la red neuronal.

3.1 Origen de datos

Este proyecto se centra en estudiar los sonidos captados por una serie de micrófonos repartidos por el Parque Natural de la Albufera de Valencia, situado a tan solo 10 km al sur de la ciudad de Valencia. Estos micrófonos están distribuidos en diez nodos acústicos, situados en la zona de la laguna tal y como se muestra en el siguiente mapa.

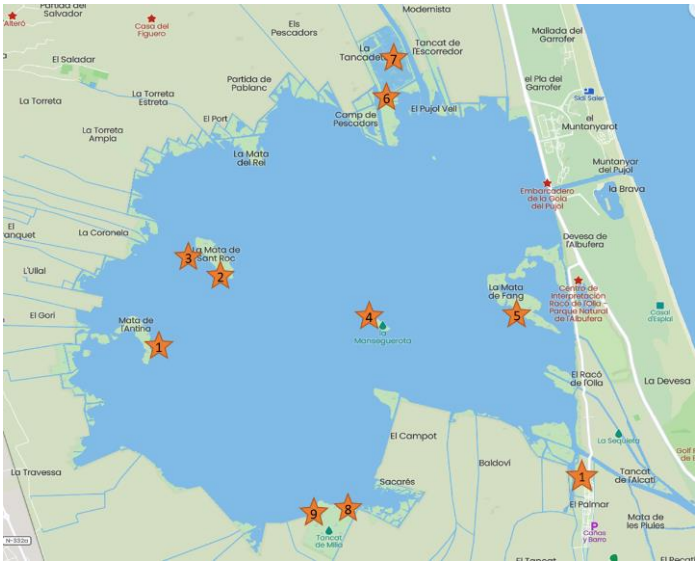

Figura 21. Ubicación de los nodos acústicos en la albufera. [18]

Tal y como puede observarse se dispone de 5 nodos situados en islas dentro de la propia albufera y los otros cinco nodos están instalados por los terrenos que rodean la laguna. Esta distribución ofrece un amplio abanico de sonidos, algunos situados en zonas más transitadas y con alta actividad humana y otros en zonas más apartadas con gran riqueza de sonidos de origen natural.

Los datos disponibles se tratan de grabaciones realizadas entre 2022 y 2023, en ficheros de audio de 60 minutos de duración y en formato wav. Todas las grabaciones se realizan con una frecuencia de muestreo de 24000 Hz.

3.2 Fase 1: Exploración y técnicas no supervisadas.

3.2.1 Base de datos

Para empezar a trabajar es importante escoger y definir un conjunto de datos iniciales que contenga muestras de todos los sonidos de interés. Para la elaboración de esta base de datos es importante tener en cuenta el objetivo principal del estudio: Detectar sonidos de origen humano. Dada la naturaleza de la actividad humana nos centramos en las grabaciones que se realizan entre las 10:00 h y las 13:00 h, ya que en estas horas son las de mayor actividad humana. Se analizan las grabaciones de dos nodos diferentes en estas franjas y se crea un dataset inicial teniendo en cuenta que el contenido presente debe contener tanto los sonidos de interés como otros sonidos para comprobar que el proceso de agrupamiento es capaz de separar las diferentes fuentes de sonidos de forma adecuada. Con esta premisa se genera un conjunto de datos que contiene la información descrita en la tabla 1.

Tipo de sonido	**Duración (s)**
Paisaje acústico	45
Paisaje acústico	30
Avión	45
Avión	45
Motor de Barco	30
Motor de Barco	30
Motor de Barco y voces	45
Voces	30
Voces	30

Tabla 1. Distribución de grabaciones en base de datos inicial.

En total el dataset inicial constará de una duración de 330 segundos. Representando la información contenida con las técnicas descritas.

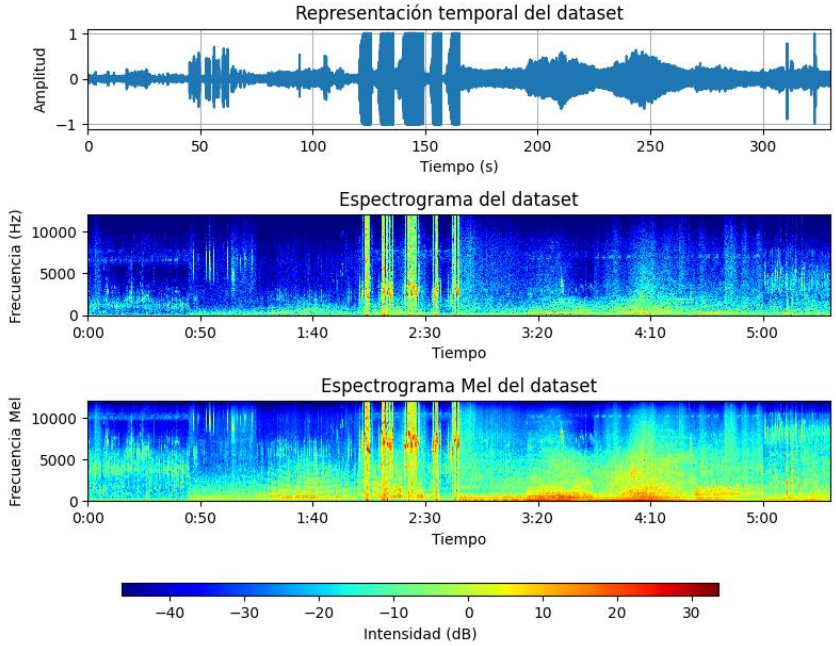

Figura 22. Representación de los audios de la base de datos inicial.

3.2.2 Procesamiento y características

Antes de pasar los datos a los modelos es necesario realizar un procesado previo para preparar los datos. Uno de los primeros pasos es segmentar la base de datos para generar muestras más pequeñas y para las cuales se extraerán las características necesarias. Definir la duración de las muestras es un proceso delicado y que depende del contexto de estudio. Dado que la duración de los sonidos que queremos detectar es relativamente elevada, se ha optado por tomar **muestras de tres segundos**. Esto divide el dataset en un total de 110 muestras, para las cuales se calcularán las diferentes características ya mencionadas.

A la hora de construir la matriz de características para un modelo de aprendizaje máquina, es fundamental obtener tanto la tendencia central como la dispersión de las características a lo largo del tiempo en los diferentes segmentos de la señal. Para ello, se utilizarán la media y la varianza de cada una de las características calculadas. La media proporciona una medida del valor típico de la característica, lo cual es esencial para entender el comportamiento general de la señal. Por otro lado, la varianza ofrece una visión sobre la variabilidad o consistencia de esa característica, reflejando cómo fluctúa alrededor de su media. Estas dos estadísticas juntas permiten representar de manera compacta y efectiva la información contenida en cada una de las características, facilitando su uso en modelos de clustering al capturar tanto la estabilidad como la dinámica de los patrones presentes en los datos.

Antes de pasar a los modelos de aprendizaje automático hay que tener en cuenta la variabilidad de los valores de las características, como se comentaba anteriormente es importante que se encuentren en un rango comparable, es por ello que se aplica un escalado. A continuación, se calcula el histograma de cada una de las características para comprobar si contienen la suficiente información como para ser valiosas para el clasificador. El objetivo es buscar características que presenten un histograma con variabilidad, si el histograma presenta una distribución con valores muy concentrados en un rango muy pequeño significa que esta característica aporta poca información.

Tras observar todos los histogramas es fácilmente observable que tanto la planitud espectral como el flujo espectral presentan muy poca información tanto su media como su varianza.

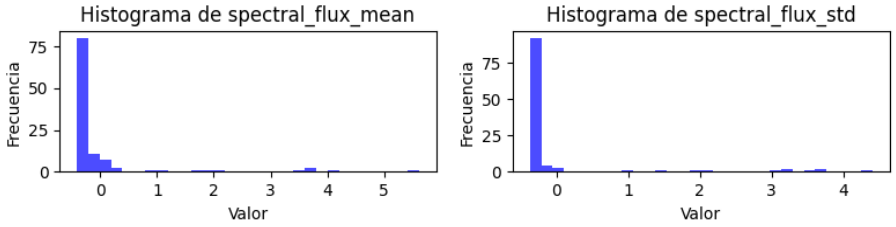

Figura 23. Histograma de media y varianza del flujo espectral.

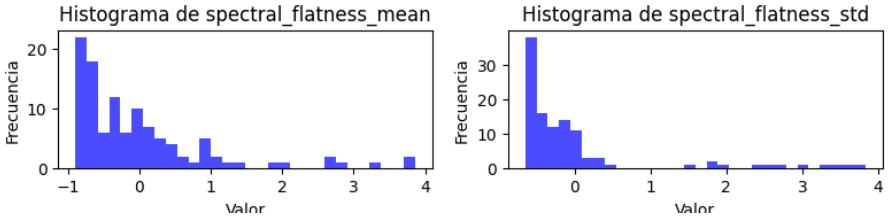

Figura 24. Histograma de media y varianza de la planitud espectral.

Sin embargo, las características como el centroide, el ancho de banda y el roll-off espectral presentan un histograma mucho más rico.

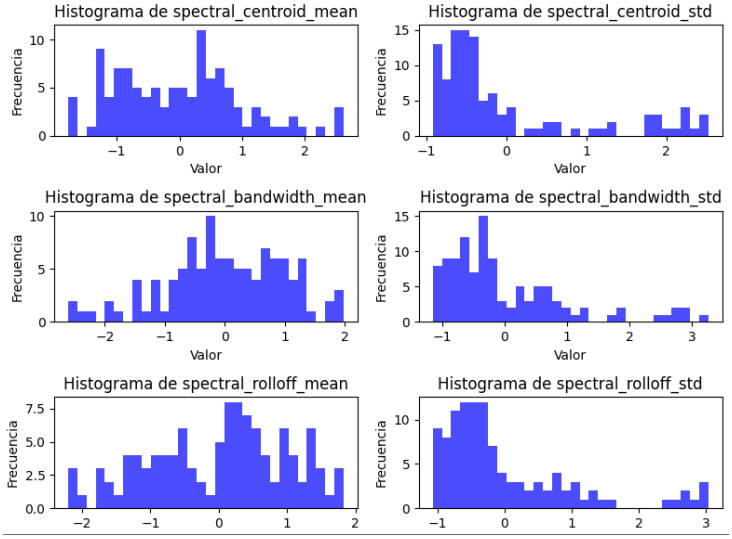

Figura 25. Histograma de múltiples características.

Tras eliminar las características innecesarias, se vuelve a calcular la matriz de características y se normaliza, de esta forma nos quedamos con una matriz con las siguientes características calculadas:

- Centroide espectral: Descrito en el apartado 2.2.2
- Ancho de banda espectral: Descrito en el apartado 2.2.3
- Asimetría espectral: Descrito en el apartado 2.2.5
- Roll-off espectral: Descrito en el apartado 2.2.6
- Contraste espectral: Descrito en el apartado 2.2.7
- Coeficientes MFCC: Descrito en el apartado 2.2.9

3.2.3 Selección de modelos

En este apartado se describen las pruebas realizadas para la elección de los distintos modelos ya mencionados.

Reducción de dimensionalidad

Como ya se mencionaba en la descripción de los modelos de reducción de dimensionalidad, el algoritmo t-SNE es más común para la visualización, además su utilización requiere de la configuración y ajuste de varios parámetros, por lo que aumenta la complejidad y dificulta su uso, es por ello por lo que he optado por la técnica de análisis de componentes principales (PCA), debido a su facilidad de uso.

El número de componentes principales se basa en el acumulado de varianza explicada. Cada componente principal explica una cierta cantidad de la varianza total de los datos. El objetivo es seleccionar un número de componentes que capture una proporción significativa de la varianza total con el fin de reducir la dimensionalidad del conjunto de datos sin perder información crucial. Esto se puede ver reflejado en la figura 26.

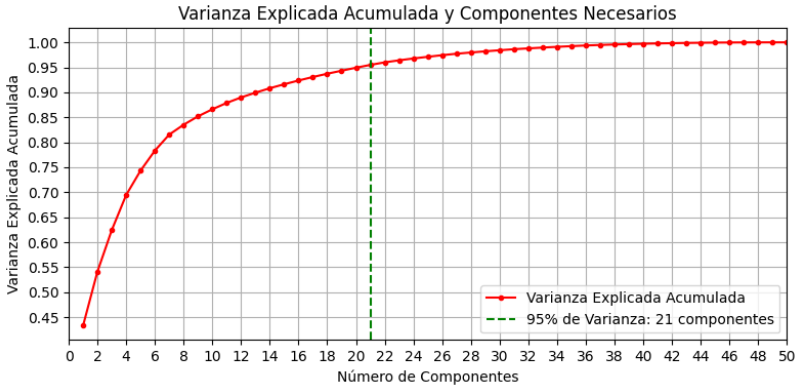

Figura 26. Evolución de la varianza explicada frente al número de componentes PCA

Aplicando este algoritmo podemos proyectar los puntos que componen nuestro dataset inicial utilizando las 2 primeras componentes, ya que estas capturan gran parte de la varianza y permiten una visualización sencilla y efectiva, tal y como se representa en la figura 27.

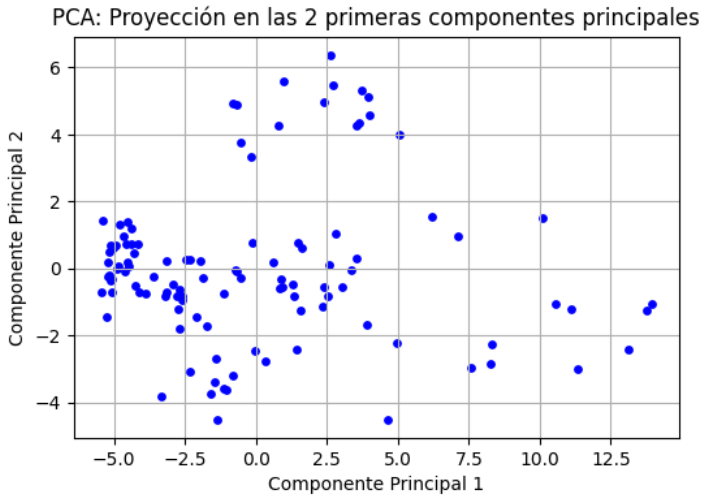

Figura 27. Proyección de las 2 primeras componentes principales.

Clustering

Para escoger el mejor modelo se van a realizar diferentes pruebas aplicando cada modelo y obtener métricas para comparar los resultados.

Hay que tener en cuenta que existe una variable aleatoria, random_state, que afecta a la inicialización de los modelos, y cambia cada vez que se ejecuta. Es por ello que se selecciona un estado inicial fijo para simplificar el proceso.

- **KMEANS**

La primera decisión que tomar a la hora de utilizar el modelo K-Means es cuántos grupos o clúster necesitamos. Para ello, existe una técnica denominada el método del codo, que se utiliza para determinar el número óptimo de clúster. Esta técnica consiste en calcular la suma de las distancias al cuadrado dentro de los clústeres (inercia) para diferentes valores de k (número de clústeres) y graficar estos valores. El punto donde la disminución de la inercia comienza a volverse menos pronunciada, formando un "codo" en la curva, sugiere el número óptimo de clústeres a utilizar en el modelo. Hay que tener en cuenta que este método es una guía para ayudar a encontrar el valor óptimo de k y depende mucho de la naturaleza de los datos

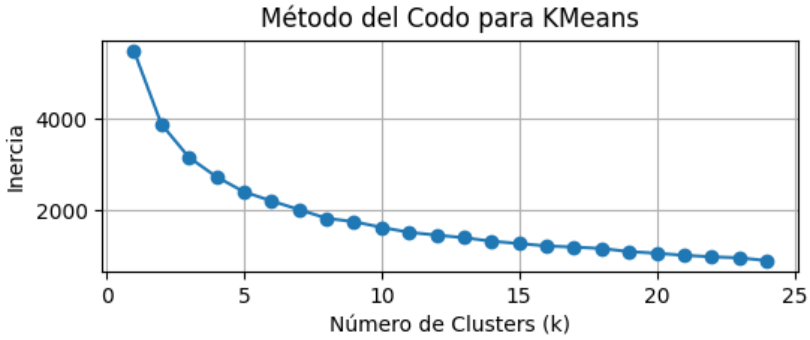

Figura 28. Método del codo para encontrar valor k.

En la figura 28 podemos ver un cambio de pendiente a partir del k=3, este podría ser un valor ideal para agrupar los sonidos utilizando una etiqueta para el paisaje acústico y las otras 2 para sonidos de origen humano, pero dada la naturaleza aleatoria de los sonidos que podemos encontrar parece poco factible una clasificación tan precisa, es por ello que hay que verificar el desempeño del modelo a la hora de clasificar. Para ello se realiza una serie de pruebas, utilizando el K-Means un número de clústeres desde 3 hasta 8. Para comprobar los resultados se realiza un agrupamiento de los sonidos por su etiquetado resultante y se reproducen, escuchando qué tipo de sonidos están presentes y si están entremezclados.

En el K-Means con 3,4 y 5 clústeres los sonidos están demasiado entrelazados, es decir, el modelo no es capaz de separar y agrupar correctamente los sonidos, existe alguna etiqueta que parece agrupar un tipo de sonido que es muy característico, pero el resto de los sonidos no los separa adecuadamente.

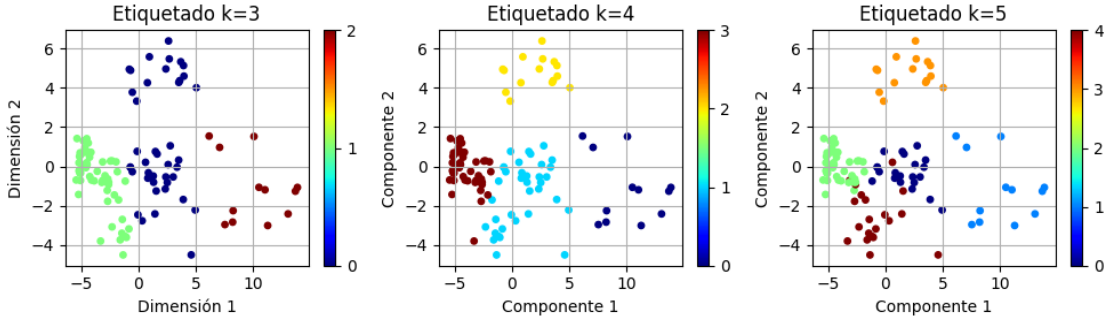

Figura 29. Proyección con el etiquetado K-Means con k=3, 4 y 5

Con k=6 empieza a apreciarse cierta separación, pero aunque parece acercarse el modelo parece que siguen mezclados sonidos, k=7 parece hacer una buena separación y con k=8 empieza a separar sonidos muy similares en grupos más pequeños.

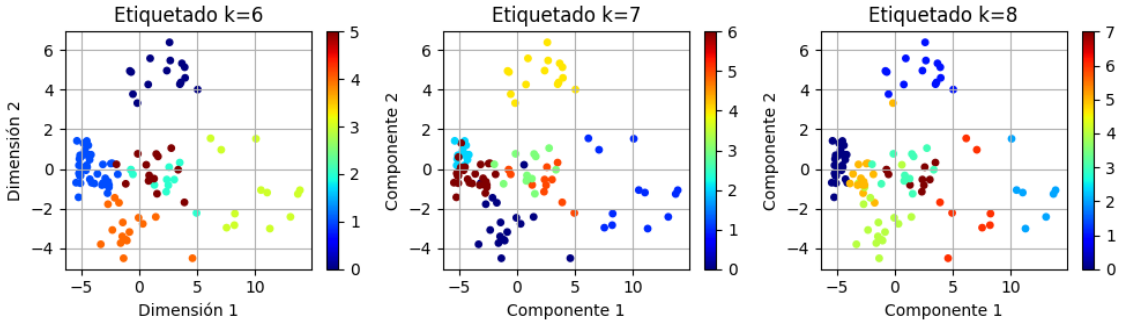

Figura 30. Proyección con el etiquetado K-Means con k=6, 7 y 8

A partir de k=6 parece ser el número de clústeres con el que el modelo empieza a distinguir sonidos de animales de las voces humanas. Para visualizar los resultados se realiza un etiquetado grosso modo del dataset inicial, asignando etiquetas a las distintas muestras según la percepción inicial, aunque es un método sesgado y no cubre todas las muestras al detalle nos puede servir para hacer una comparación aproximada de cómo se comporta el modelo.

Tipo de sonido	**Duración (s)**	**Etiqueta manual**
Paisaje acústico	45	4
Paisaje acústico - Más ruidoso	30	0
Avión	45	2
Avión - con mucho ruido de pájaros	45	3
Motor de Barco	30	1
Motor de Barco	30	1
Motor de Barco	45	1
Voces – con ruido de motor y pájaros	30	1
Voces – con ruido de pájaros	30	5

Tabla 2. Distribución de grabaciones en base de datos inicial.

Aplicando estas etiquetas podemos representar ahora ambas proyecciones tal y como se ve en la figura 30,

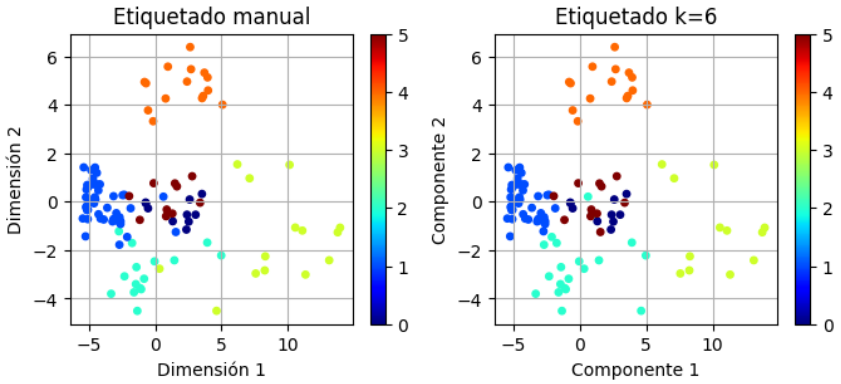

Figura 31. Comparación del etiquetado a mano con el K-Means

Una forma de analizar mejor estos resultados cuando se tiene el etiquetado es utilizando la matriz de confusión. Una matriz de confusión es una tabla que se utiliza en el aprendizaje automático para evaluar el rendimiento de un modelo de clasificación. Esta herramienta nos permite visualizar de forma clara y concisa los aciertos y errores que comete un modelo al clasificar nuevos datos.

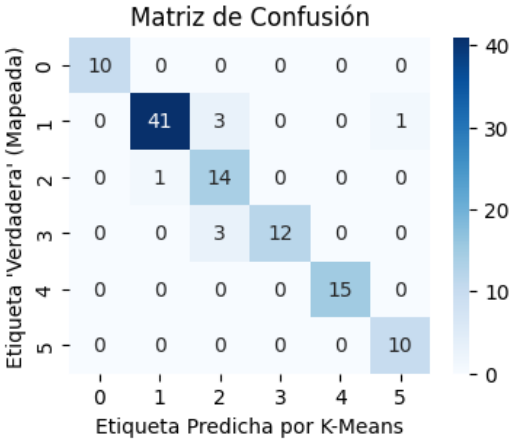

Figura 32. Matriz de confusión

Tal y como vemos en la matriz de confusión, utilizando el etiquetado grosso modo que hemos realizado parece tener una precisión muy elevada, ya que los elementos están distribuidos principalmente en la diagonal, a excepción de 8 muestras. El fallo en estas muestras puede deberse a un fallo en el etiquetado, ya que al etiquetar de forma general el conjunto entero es muy probable que en algún momento una muestra sea claramente de otro grupo.

- **DBSCAN**

Como ya se explicaba, DBSCAN es un algoritmo basado en la densidad de los datos. A la hora de implementarlo es necesario configurar dos parámetros:

- EPS: Es el radio de vecindad, hace relación a la distancia entre los puntos vecinos.
- Min_sample: Número mínimo de puntos necesario para hacer un grupo.

Estos parámetros son muy complejos de ajustar, un valor muy pequeño de EPS podría fragmentar los clústeres, mientras que uno muy grande podría fusionarlos, además definir el número mínimo de muestras es algo complejo, ya que los diferentes tipos de sonidos pueden requerir diferentes valores de muestras mínimas, lo que complica en análisis.

Teniendo en cuenta la naturaleza arbitraria de los sonidos presentes en un medio como el Parc de l'Albufera dificulta aún más el proceso, ya que son extremadamente variados en amplitud, frecuencia y duración, además de presentar densidades diferentes.

Pese a esto se realizan una serie de pruebas ajustando los parámetros, pero los resultados no parecen ser capaces de distinguir correctamente algunos de los grupos que K-Means sí que consigue clasificar.

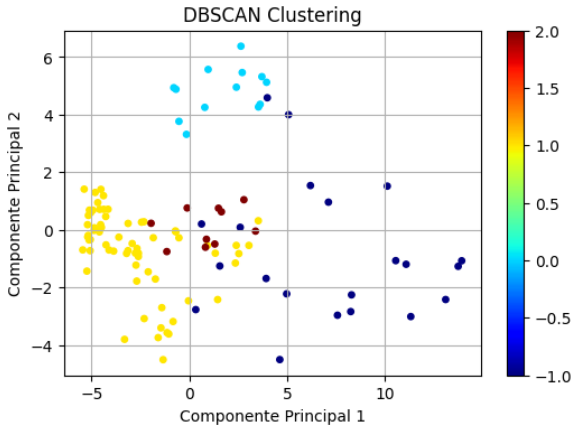

Figura 33. Proyección del etiquetado DBSCAN con EPS 5 y 5 muestras

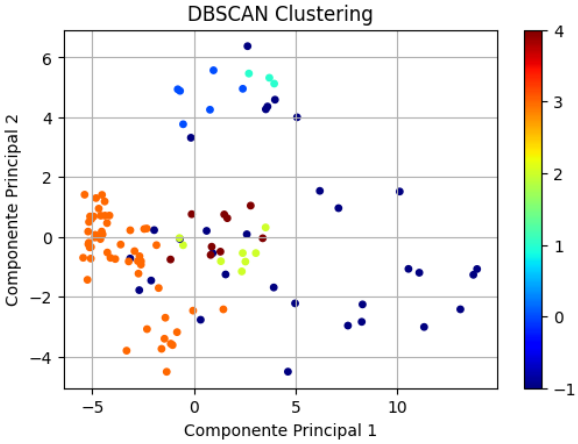

Figura 34. Proyección del etiquetado DBSCAN con EPS 4 y 5 muestras

3.2.4 Conclusiones de fase 1.

Tal y como se ha podido comprobar las técnicas no supervisadas son muy útiles para poder hacer un análisis previo de los diferentes sonidos, preparar los modelos adecuados requiere de un conocimiento previo de los datos, pero agiliza enormemente la labor. Las técnicas de reducción de dimensionalidad son prácticas tanto individualmente como en conjunto con las técnicas de agrupación. Hemos comprobado como en conjunto un algoritmo PCA con 21 componentes y un clasificador K-Means con k=6 son capaces de separar y detectar matices en este primer grupo de datos.

3.3 Fase 2: Aprendizaje profundo.

Teniendo en cuenta el objetivo del proyecto, buscamos poder detectar y clasificar los diferentes sonidos de origen humano, en el dataset se localizan en esencia 3 sonidos de interés: Motores de barca, avión y conversaciones. Para poder afinar este proceso se plantea utilizar Redes Neuronales Convolucionales. Estas redes neuronales requieren que la entrada esté etiquetada, por lo que se han utilizado los resultados del clasificador no supervisado de la primera etapa del proyecto, aunque antes de ello se realiza un ejercicio de revisión y ajustado de las etiquetas que puedan no ser correctas, además de reducir las etiquetas posibles a 4: Paisaje acústico, voces, avión y motor.

3.3.1 Ampliación de base de datos

Otro de los planteamientos clave para garantizar un correcto entrenamiento de la red neuronal es la ampliación del dataset, ya que necesitaremos información tanto para entrenar como para validar, es por ello que se hacer un nuevo proceso de búsqueda y selección de fragmentos para ampliar el contenido a un total de 35 minutos de grabación. Dejando 690 muestras de 3 segundos.

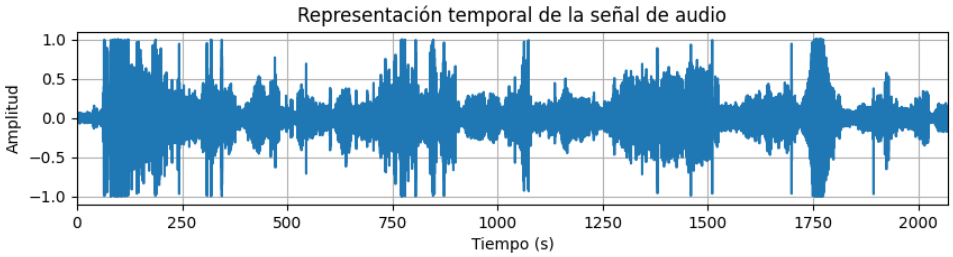

Figura 35. Representación del dataset ampliado.

Distribuidas las muestras de la siguiente manera:

Sonido	**Duración (s)**
Paisaje acústico	480
Avión	495
Motor	480
Voces	615

Tabla 3. Distribución de sonidos del nuevo dataset ampliado

3.3.2 Procesamiento y características

Como ya se ha indicado anteriormente, las redes neuronales funcionan especialmente bien a la hora de analizar imágenes, es por ello que el espectrograma es una elección particularmente efectiva como entrada, en especial se utiliza el espectrograma Mel, ya que nuestro objetivo es distinguir principalmente la información a bajas frecuencias, que es donde se concentran la mayoría de los sonidos de origen humano, y esta representación realza la información a esas frecuencias.

El proceso de obtención del espectrograma Mel será similar al planteado en la fase anterior, se procesa la señal en fragmentos de tres segundos de duración y se calcula el espectrograma de Mel para cada fragmento y esto se pasará a la red Neuronal.

A la hora de realizar el cálculo de la STFT se utiliza una ventana de 2048 con un salto entre de 512, el resultado se pasa por un el banco de filtros Mel con 128 bandas, con estos valores se busca un equilibrio entre capturar detalles y carga computacional. De esta forma el resultado es el espectrograma Mel de cada fragmento, una matriz de datos con dimensiones (128, 141), este dato es muy importante ya que la primera capa de la red neuronal debe diseñarse para el tamaño de entrada de los datos.

Una vez obtenido la base de datos de espectrogramas dividimos el contenido en dataset de entrenamiento y dataset de validación, siendo un 10% de los datos reservados para poder validar el rendimiento de la red neuronal.

3.3.3 Diseño de la red neuronal

Al diseñar una CNN, es fundamental comprender la función de cada componente y cómo interactúan entre sí.

Las capas convolucionales son la entidad central de una CNN. Estas capas aplican filtros a las imágenes de entrada para extraer características locales. Es importante realizar una elección adecuada de los parámetros para garantizar un correcto desempeño de la red:

- Número de capas: La elección del número de capas es crucial, más capas permiten capturar características más complejas y abstractas pero a su vez incrementa la complejidad del modelo, elevando también el riesgo de sobreajuste, es importante encontrar un equilibrio entre profundidad y generalización.
- Número de filtros: Determina la cantidad de características que se extraen en cada capa. De forma similar al número de capas, mayor cantidad puede generar sobreajuste y sobre todo incrementa el coste computacional.
- Tamaño del filtro: O Kernel, define la región de la imagen observada por cada filtro. Generalmente de tamaños 3x3 o 5x5. Un filtro mayor capta patrones de mayor tamaño, por otro lado filtros más pequeños capturan detalles más finos.
- Funciones de activación: Introduce no linealidad en la red, permitiendo que la CNN aprenda representaciones más complejas. Las funciones de activación más comunes son ReLU (Rectified Linear Unit), sigmoid y tanh. Siendo ReLU la más utilizada en la CNN.

Además de las capas convolucionales, otras capas juegan un papel importante en la arquitectura de una CNN:

- **Max Pooling:** Reduce la dimensionalidad de la representación al seleccionar el valor máximo en una región de la capa anterior. Esta operación ayuda a reducir el sobreajuste y a hacer la red más invariante a pequeñas traslaciones.
- **Flatten:** Aplana la salida de las capas convolucionales en un vector unidimensional, preparando los datos para ser procesados por las capas densas.
- **Dense:** Capas totalmente conectadas que realizan una combinación lineal de las entradas y aplican una función de activación. Estas capas son responsables de la clasificación final en muchas arquitecturas de CNN.
- **Dropout:** Una técnica de regularización que consiste en desactivar aleatoriamente un porcentaje de neuronas durante el entrenamiento. Esto ayuda a prevenir el sobreajuste al reducir la codependencia entre las neuronas.

Modelos.

Para el diseño de la red neuronal se van a realizar diferentes pruebas teniendo en cuenta varias estructuras ya conocidas:

Estructura	Capas Conv.	Capas Densas	Función de Activación	Otras Técnicas Incluidas
LeNet	2	2	Sigmoid/Tanh	Subsampling (MaxPooling), Data Normalization
AlexNet	5	3	ReLU	Dropout, MaxPooling, Data Augmentation
ResNet	34/50	1	ReLU	Skip Connections (Residuals), Batch Normalization
VGG	8/10/13	3	ReLU	MaxPooling, Batch Normalization

Tabla 4. Estructuras de redes neuronales conocidas.

Las funciones de activación para las capas convolucionales más utilizadas son ReLU, a excepción de LeNet, que utiliza sigmoid, en la capa densa de salida se utilizará SoftMax.

La estructura de estas redes es muy similar, concatenando capas convolucionales al principio con capas MaxPooling, y finalizando en las capas densas, este tipo de estructuras se puede observar en la representación de una VGG-16 en la figura 36.

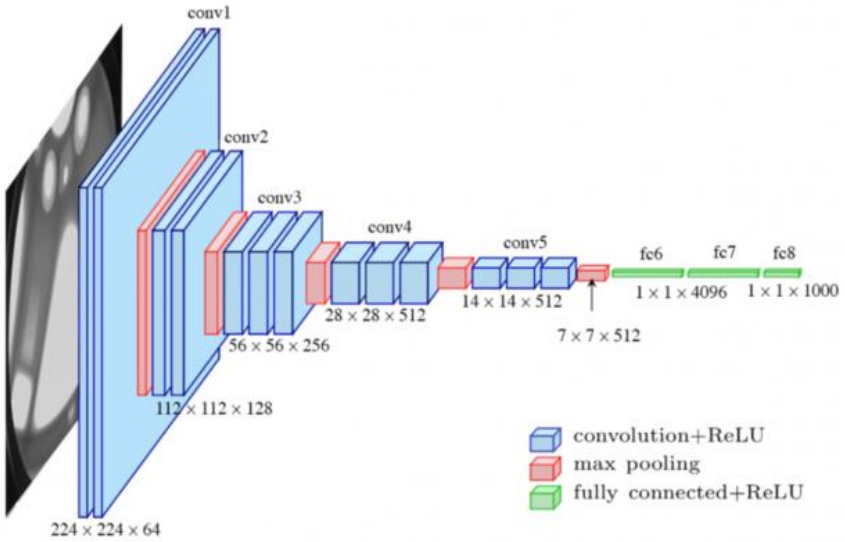

Figura 36. Representación de las capas de una red VGG-16 [19]

Para cada una de las redes se realiza una serie de pruebas cambiando los hiperparametros batch size y learning rate, además se realizan ajustes en algunas de las capas, añadiendo o ajustando principalmente las capas dropout. Para el entrenamiento se establecen 50 epochs con un batch-size de 16 a 32, utilizando el optimizador Adam y una tasa de aprendizaje del 0,001 o 0,0001, dependiendo de la prueba. También se establece una técnica de parado temprano para evitar el sobreajuste, esta técnica detiene el entrenamiento si las pérdidas de validación incrementan durante una serie de épocas, definida por el parámetro “paciencia”, volviendo a los mejores pesos obtenidos.

En resumen se realizan las siguientes pruebas:

Modelo	v.	Batch	Learning Rate	Dropout	Parámetros
LeNet-5	A	16	0,0001	No	8.768.300
LeNet-5	B	32	0,0001	No	8.768.300
AlexNet	A	32	0,0001	0,25	26.817.220
AlexNet	B	32	0,0001	0,5	26.817.220
Resnet_50	A	32	0,0001	No	13.903.044
Resnet_50	B	32	0,0001	0,5	14.953.156
VGG_11	A	32	0,0001	No	59.575.556
VGG_11	B	32	0,0001	0,25	59.575.556
VGG_11	C	32	0,0001	0,5	59.575.556
VGG_13	A	32	0,0001	No	59.760,068
VGG_13	B	32	0,0001	0,25	59.760,068
VGG_13	C	32	0,0001	0,5	59.760,068
VGG_16	A	32	0,0001	No	65.069.764
VGG_16	B	32	0,0001	0,25	65.069.764
VGG_16	C	32	0,0001	0,5	65.069.764

Tabla 5. Tabla de estructura de los modelos de pruebas.

3.3.4 Evaluación.

Las métricas empleadas para evaluar el desempeño del modelo son ampliamente reconocidas y aceptadas en el ámbito de la ciencia de datos y el aprendizaje automático. En particular, se han utilizado las métricas de precisión, sensibilidad (también conocida como recall), y el F1-score.

La precisión mide la proporción de verdaderos positivos sobre el total de predicciones positivas, proporcionando una indicación de la exactitud de las predicciones positivas realizadas por el modelo.

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \quad \textbf{(3.1)}$$

La sensibilidad, por su parte, refleja la capacidad del modelo para identificar correctamente todas las instancias positivas, es decir, cuán efectivo es en la detección de verdaderos positivos en relación con el total de casos que deberían haber sido identificados como positivos.

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \quad \textbf{(3.2)}$$

Por último, el F1-score combina ambas métricas en una única medida armónica, equilibrando la precisión y la sensibilidad, especialmente en contextos donde existe un desequilibrio de clases.

Estas métricas, conjuntamente, ofrecen una visión integral del comportamiento del modelo, permitiendo evaluar su rendimiento de manera más completa y contextualizada.

$$F1Score = \frac{2\ Precision\ Recall}{Precision + Recall} \quad (3.3)$$

Con estas métricas definidas, se procede al entrenamiento y la evaluación de las redes neuronales diseñadas. La aplicación de estas métricas permite la evaluación del rendimiento y las capacidades de cada modelo a partir del análisis de las predicciones utilizando el dataset de validación. A continuación se muestran la media de estas métricas entre las 4 etiquetas indicadas.

Modelo	v.	Accuracy	Precisión	Recall	F1-score
LeNet-5	A	0,855	0,85	0,86	0,85
LeNet-5	B	0,884	0,88	0,89	0,89
AlexNet	A	0,884	0,88	0,89	0,88
AlexNet	B	0,927	0,93	0,93	0,93
Resnet_50	A	0,913	0,91	0,91	0,91
Resnet_50	B	0,826	0,86	0,80	0,79
VGG_11	A	0,913	0,91	0,91	0,91
VGG_11	B	0,942	0,94	0,95	0,94
VGG_11	C	0,898	0,90	0,90	0,89
VGG_13	A	0,913	0,91	0,91	0,91
VGG_13	B	0,898	0,89	0,90	0,90
VGG_13	C	0,927	0,94	0,92	0,93
VGG_16	A	0,826	0,83	0,83	0,83
VGG_16	B	0,913	0,91	0,91	0,91
VGG_16	C	0,841	0,85	0,85	0,84

Tabla 6. Tabla de resultados de entrenar las distintas CNNs

De todos los modelos entrenados los que a primera vista presentan un mejor rendimiento, basándonos en la métrica *Accuracy*, son los listados en la tabla 7:

Modelo	v.	Accuracy
VGG_11	B	0,942
VGG_13	C	0,927
AlexNet	B	0,927
VGG_16	B	0,913
Resnet_50	A	0,913

Tabla 7. Redes con mejor A*ccuracy*.

Para estos modelos se representa el progreso del entrenamiento a través del histórico de pérdidas y *accuracy* a lo largo de los distintos ciclos de entrenamiento, además se muestra las tablas con los valores de las métricas indicadas anteriormente para las distintas etiquetas, obtenidos al evaluar el proceso de entrenamiento de cada modelo con las muestras de validación.

- **Resultados iniciales del entrenamiento del modelo VGG11_b**

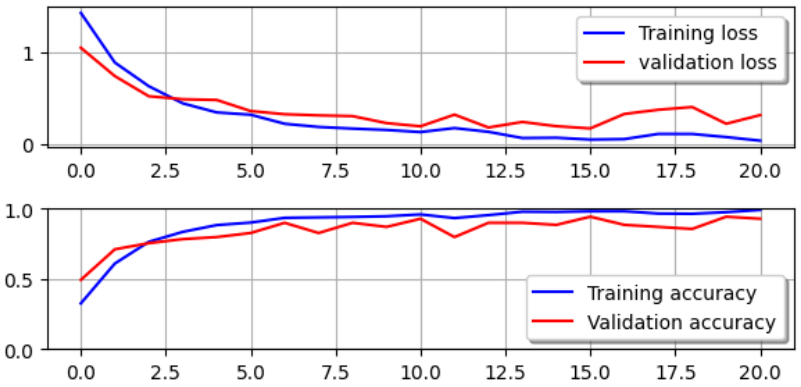

Figura 37. VGG_11b: Evolución de pérdidas y precisión a lo largo de cada Epoch en fase entrenamiento

	Precisión	Recall	F1-Score
Paisaje	0,93	1,00	0,96
Avión	0,93	1,00	0,97
Motor	0,88	0,82	0,88
Voces	1,00	0,90	0,95

Tabla 8. VGG_11b: Resultados de métricas de validación por etiqueta.

- **Resultados iniciales del entrenamiento del modelo VGG13_c**

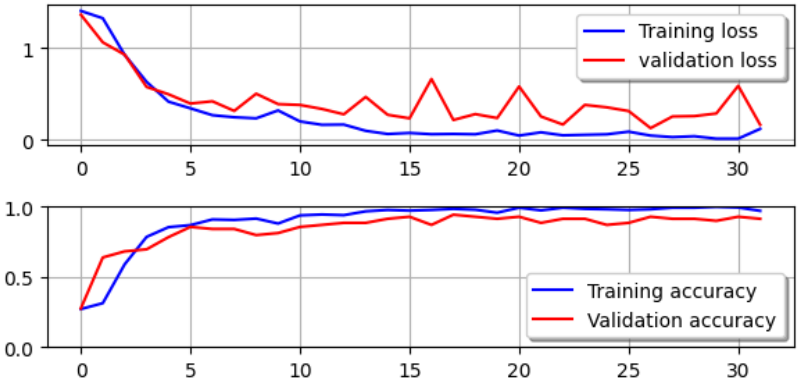

Figura 38. VGG13_c: Evolución de pérdidas y precisión a lo largo de cada Epoch en fase entrenamiento

	Precisión	Recall	F1-Score
Paisaje	1,00	0,92	0,96
Avión	0,90	1,00	0,95
Motor	0,93	0,82	0,88
Voces	0,9	0,95	0,93

Tabla 9. VGG13_c: Resultados de métricas de validación por etiqueta.

- **Resultados iniciales del entrenamiento del modelo AlexNet_b**

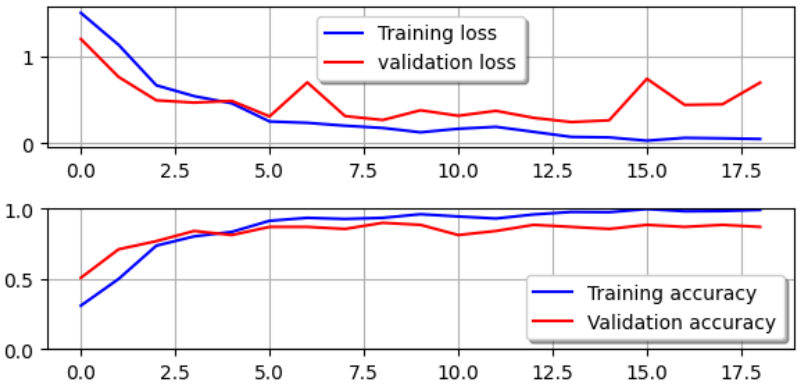

Figura 39. AlexNet_b: Evolución de pérdidas y precisión a lo largo de cada Epoch en fase entrenamiento

	Precisión	Recall	F1-Score
Paisaje	0,75	0,92	0,83
Avión	0,86	0,95	0,90
Motor	0,87	0,76	0,81
Voces	1,00	0,85	0,92

Tabla 10. AlexNet_b: Resultados de métricas de validación por etiqueta.

- **Resultados iniciales del entrenamiento del modelo VGG16_b**

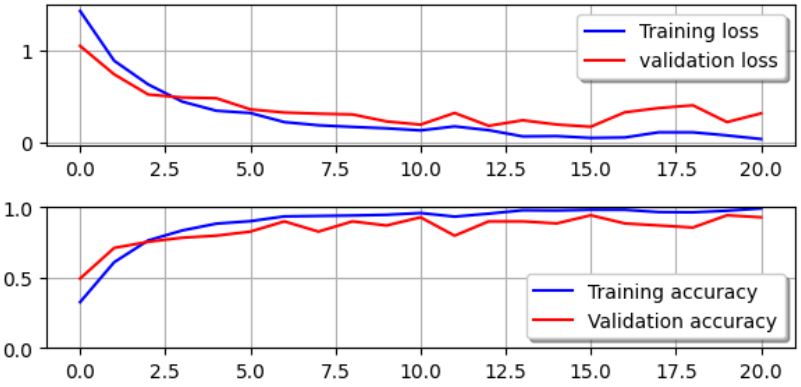

Figura 40. VGG16_b: Evolución de pérdidas y precisión a lo largo de cada Epoch en fase entrenamiento

	Precisión	Recall	F1-Score
Paisaje	0,83	0,77	0,80
Avión	0,95	1,00	0,97
Motor	0,71	0,71	0,71
Voces	0,75	0,75	0,75

Tabla 11. VGG16_b: Resultados de métricas de validación por etiqueta.

- **Resultados iniciales del entrenamiento del modelo ResNet50_b**

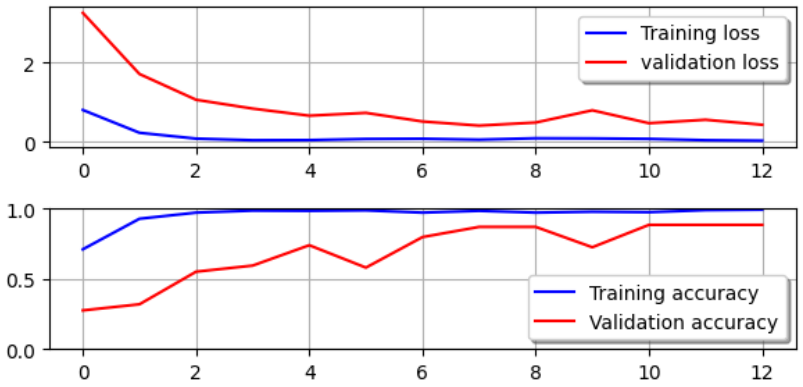

Figura 41. ResNet50_b: Evolución de pérdidas y precisión a lo largo de cada Epoch en fase entrenamiento

	Precisión	Recall	F1-Score
Paisaje	0,89	0,62	0,73
Avión	0,82	0,95	0,88
Motor	0,84	0,94	0,89
Voces	0,95	0,90	0,92

Tabla 12. ResNet50_b: Resultados de métricas de validación por etiqueta.

Para validar estos resultados se han realizado comprobaciones adicionales con muestras no conocidas por los modelos. Primero se ha utilizado el dataset primera fase del proyecto, un conjunto de muestras variadas que nos resulta fácil de reconocer. Luego procesaron 2 grabaciones de 1h en las que solo hay sonidos del paisaje y de forma puntual aparece algún avión, ya que estas son fácilmente reconocibles y el resultado es fácil de interpretar. El etiquetado está asociado de la siguiente manera:

0- Paisaje acústico
1- Avión
2- Motor de barca
3- Voces humanas

A continuación se muestran los resultados obtenidos al predecir cada una de las muestras descritas de los modelos que han tenido los mejores resultados en las pruebas que se describen.

- **Validación VGG11 con dropout 0.25**

Primero se carga el dataset de la primera etapa, que recordando está formado por 2 fragmentos de ruido natural, 2 fragmentos que contienen aviones, el segundo de ellos con ruidos de pájaro muy cercanos al micrófono, luego 3 muestras de ruido de motor de barca y por último 2 fragmentos de sonido con voces humanas, que tambien contiene algo de sonido de motor.

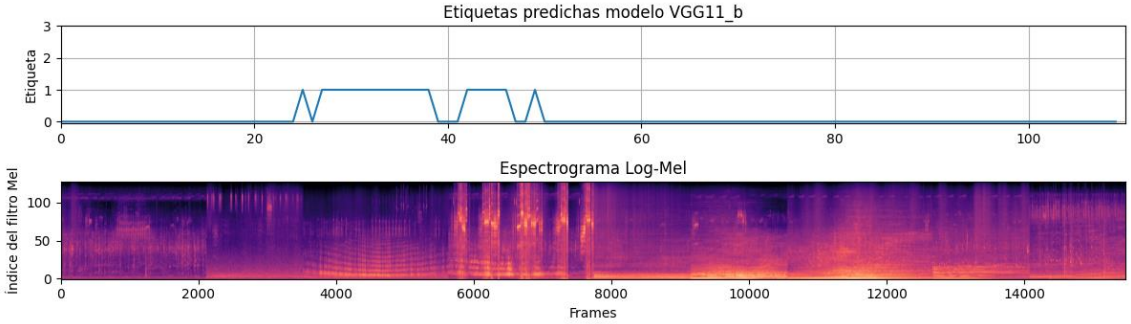

Figura 42. VGG11b, prueba con dataset variado.

Tal y como podemos observar en la figura 39, la red neuronal es capaz de identificar correctamente el sonido de avión, a excepción de algunas muestras con falso positivo y falso negativo, pero es completamente incapaz de separar el ruido del motor y las voces del paisaje acústico.

En la segunda parece que es capaz de detectar sin grandes problemas casi todos los sonidos de aviones a excepción de la segunda muestra que no es capaz de distinguirla del paisaje acústico.

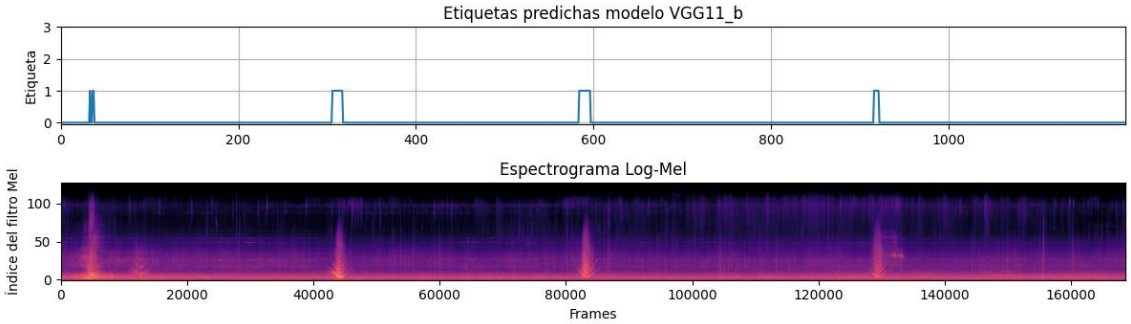

Figura 43. VGG11b, segunda prueba con grabación de aviones y paisaje.

Y por último en la tercera prueba de nuevo es capaz de distinguir el sonido de aviones frente a una fuente sonora constante a altas frecuencias.

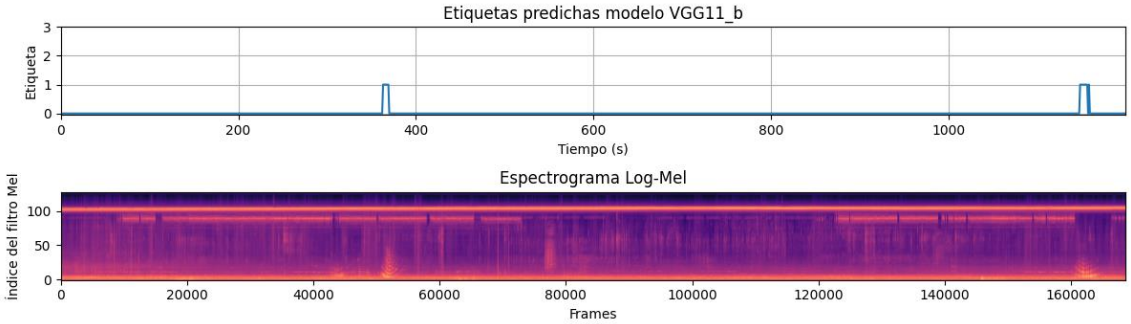

Figura 44. VGG11b, tercera prueba con grabación de aviones y sonido de fondo.

En resumen este modelo parece detectar de forma consistente el ruido de aviones pero no es capaz de distinguir sonidos de motor o voces del ruido de fondo.

- **Validación VGG13 con dropout 0.5**

De igual forma que el modelo anterior se realizan las pruebas con las 3 grabaciones escogidas. En la primera prueba con el dataset variado se aprecia que, igual que VGG11b es capaz de detectar el avión aunque aparece algún falso positivo en la parte final.

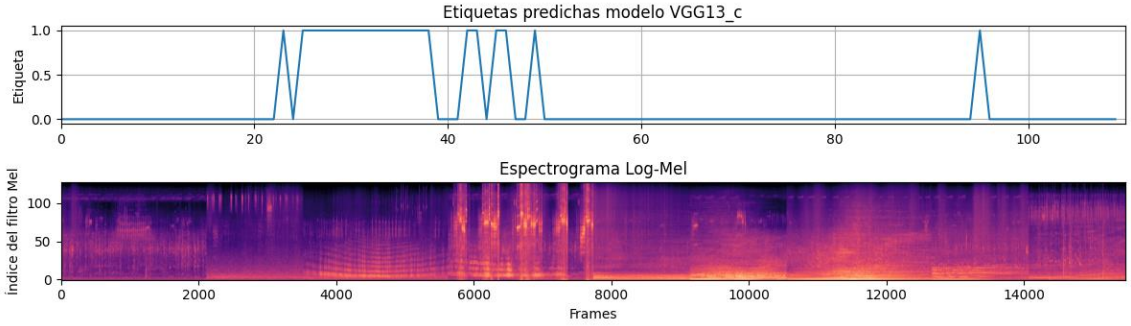

Figura 45. VGG13c, prueba con dataset variado.

En la segunda prueba parece detectar algunas muestras de avión que el anterior modelo no fue capaz de detectar pero aparecen muchos falsos positivos.

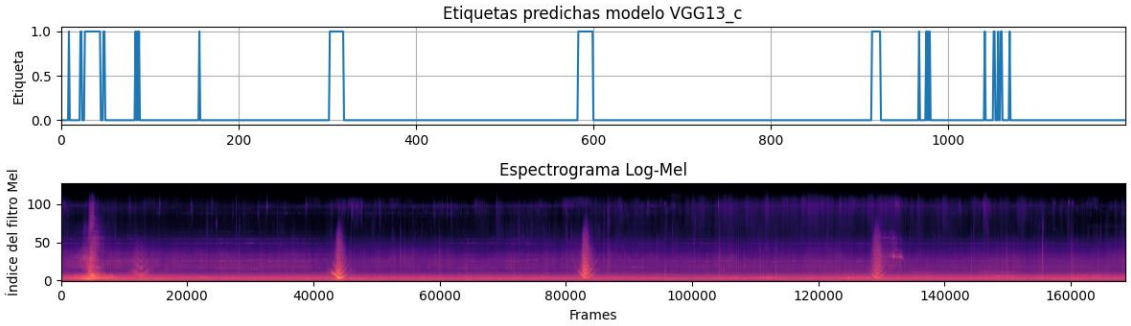

Figura 46. VGG13c, segunda prueba con grabación de aviones y paisaje.

En la última prueba detecta sin problema los aviones, clasificando el resto de sonido como paisaje acústico.

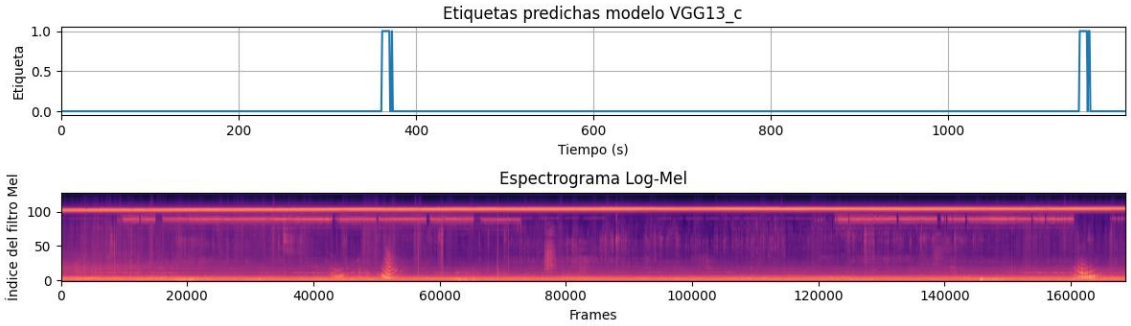

Figura 47. VGG13c, tercera prueba con grabación de aviones y sonido de fondo.

Presenta mejoras al etiquetar la muestra de avión incluso en los momentos en los que se acerca y se aleja, pero genera más falsos positivos. De nuevo es un modelo insuficiente ya que no es capaz de detectar el resto de etiquetas en la primera prueba.

- **Validación AlexNet**

Con el modelo AlexNet, aunque las métricas iniciales durante el entrenamiento parecían prometedoras, al comprobar con el dataset variado se ha observado que no es capaz de identificar ninguna clase, asignando a todo el conjunto la etiqueta de paisaje acústico, o "0", tal y como se muestra en la figura 48.

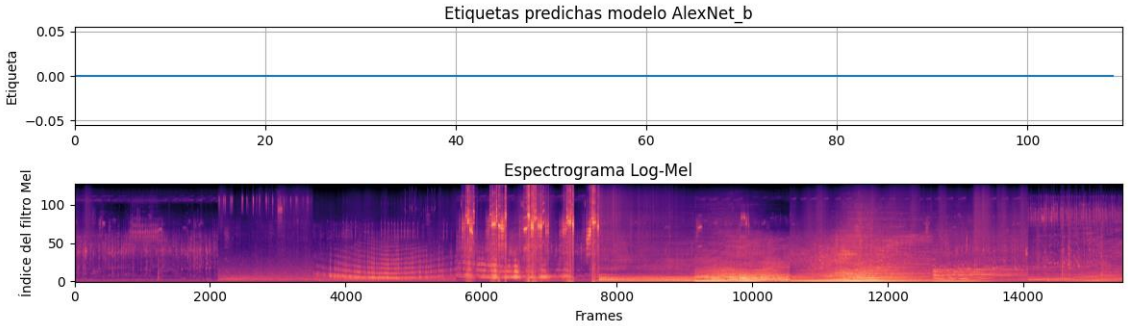

Figura 48. AlexNet, prueba con dataset variado.

Este resultado sucede de igual manera con las otras grabaciones probadas, por lo que este modelo no parece funcionar adecuadamente.

- **Validación VGG16 con dropout 0.25**

En esta ocasión, los resultados parecen mejorar con la red neuronal VGG16. En la primera prueba se aprecia que detecta adecuadamente la muestra de avión, aunque genera varios falsos positivos. Pero de nuevo no es capaz de detectar las muestras de motor o voces en este dataset.

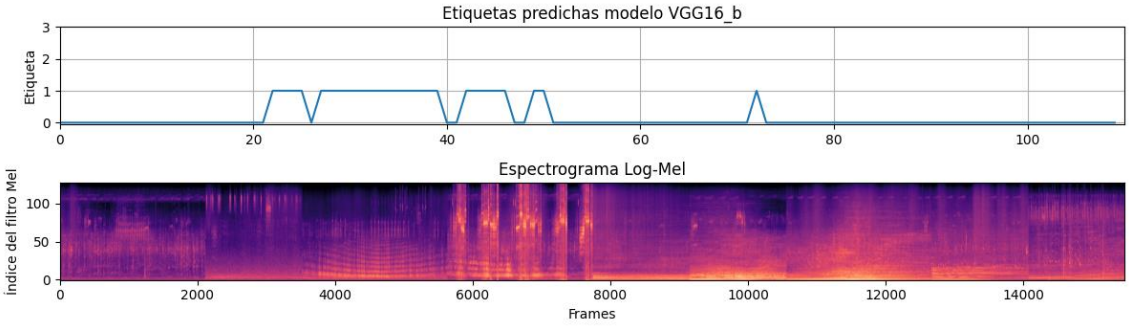

Figura 49. VGG16b, prueba con dataset variado.

Con la segunda prueba parece detectar adecuadamente las muestras de avión, pero genera muchos falsos positivos con la etiqueta de personas, "3", tal y como se ve en la figura 49.

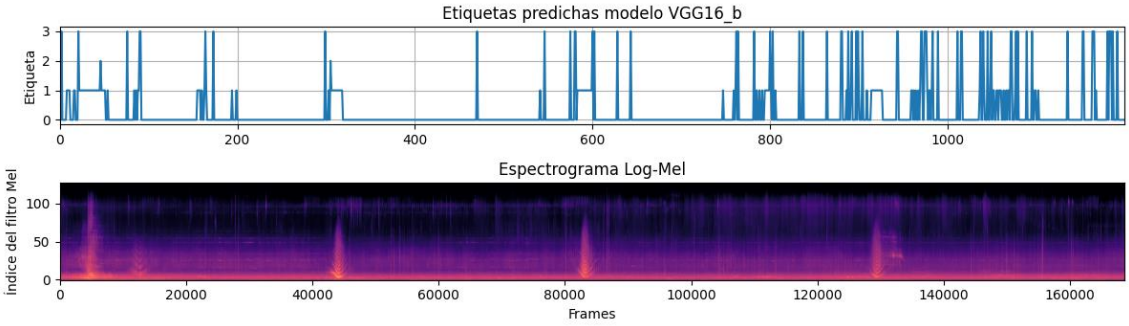

Figura 50. VGG16b, segunda prueba con grabación de aviones y paisaje.

En la última grabación se repiten los resultados de la prueba anterior, buena detección del ruido de avión pese a tener un ruido de fondo, pero tambien falsos positivos para otras etiquetas.

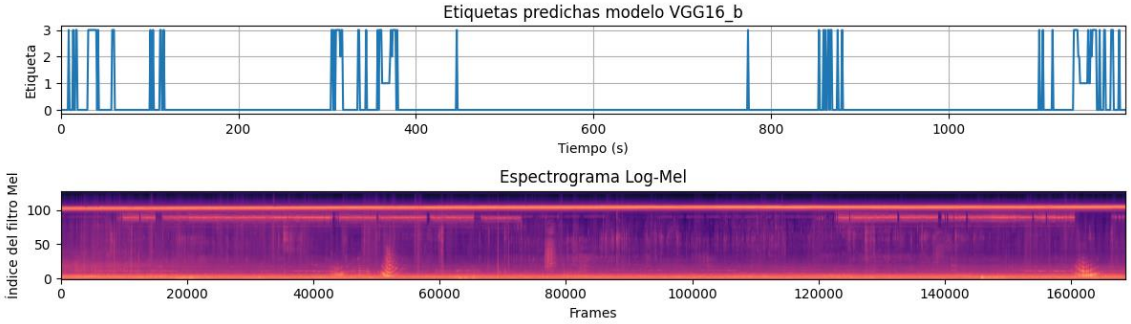

Figura 51. VGG16b, tercera prueba con grabación de aviones y sonido de fondo.

La VGG16b tampoco parece acercarse al objetivo.

- **Validación ResNet50**

Por último probamos con la red ResNet50. En este caso, al igual que con AlexNet, no parece ser capaz de distinguir ninguna etiqueta con el dataset de muestras variadas.

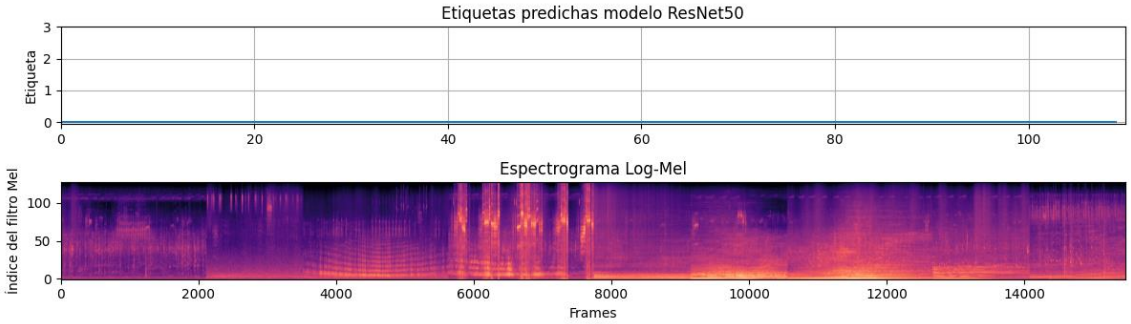

Figura 52. ResNet50, prueba con dataset variado.

Al probar con una de las grabaciones se ve que aparecen etiquetas pero ninguna parece ser acertada, tal y como se ve en la figura 53.

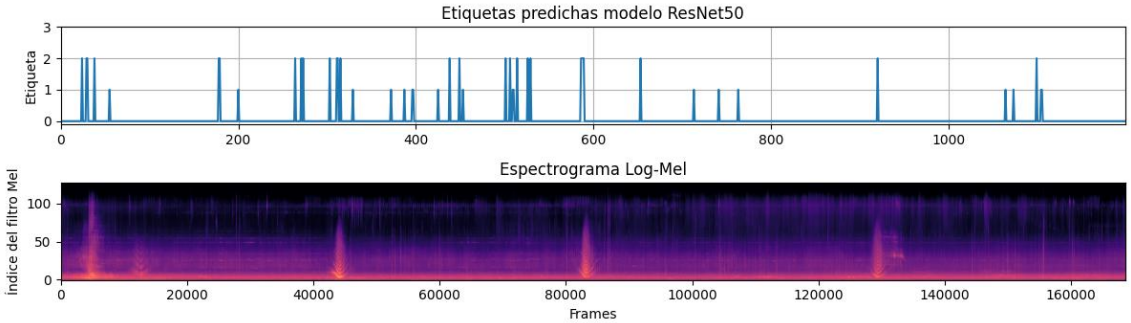

Figura 53. ResNet50, segunda prueba con grabación de aviones y paisaje.

Por lo que tampoco es válido este modelo.

3.3.5 Conclusiones de la primera etapa de evaluación.

Tal y como se ha podido comprobar, los modelos basados en la arquitectura VGG han sido capaces de detectar el sonido de avión y de clasificar el resto de sonidos como paisaje acústico, siendo la VGG11b y la VGG13c las que parecían tener menor número de errores, es un avance pero al resultar incapaz de detectar voces o motores en la primera muestra resulta insuficiente para el objetivo de este proyecto.

Tras las múltiples pruebas realizadas se llega a la conclusión que el problema de la incapacidad de detectar las muestras de motor o voces está en el dataset de entrenamiento, es por ello que se plantea una revisión completa de estas etiquetas y realizar una segunda vuelta de entrenamiento y evaluación.

3.3.6 Revisión del etiquetado del dataset.

Debido a los resultados obtenidos hasta ahora y la delicadeza que puede llegar a requerir este proceso se opta por hacer la revisión completamente manual. Para ello me centro directamente en las muestras de motor y de voces. El proceso seguido se basa tanto en la visualización del espectrograma de la señal como en escuchar el fragmento seleccionado, observando las etiquetas puestas inicialmente y detectar posibles anomalías en ese etiquetado inicial. Durante este se aprecia la presencia de un posible inconveniente a la hora de entrenar las redes neuronales que ya se mencionó al inicio, la disparidad de sonidos. Visualizando diferentes fragmentos podemos apreciar este efecto:

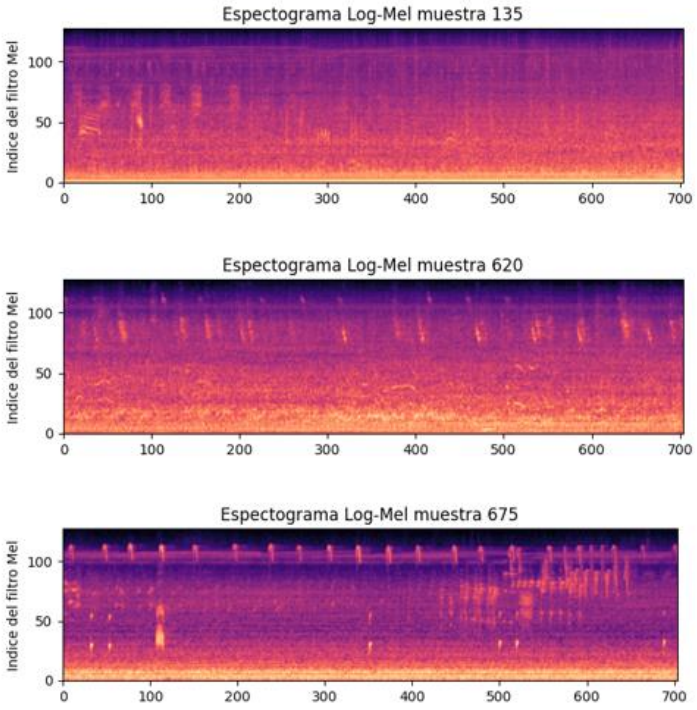

Figura 54. Espectrograma Mel de 3 muestras diferentes (Motor, Voces, Paisaje)

En la figura 54 se puede apreciar 3 espectrogramas diferentes que, al escuchar se puede apreciar las diferencias, el primero tiene una presencia clara de sonido de motor, en el segundo se escuchan claramente personas hablando, y en el tercero solo hay un ruido de fondo que parece de origen natural, pero al visualizar se atisba uno de los principales puntos en la complejidad que acompaña este trabajo.

Tras la revisión del etiquetado se ajusta para minimizar el error, ya que algunas muestras estaban mal etiquetadas, además se incorporan algunas muestras adicionales de paisaje acústico y de motor para aumentar la diversidad de muestras de estas dos etiquetas, quedando como resultado la siguiente distribución:

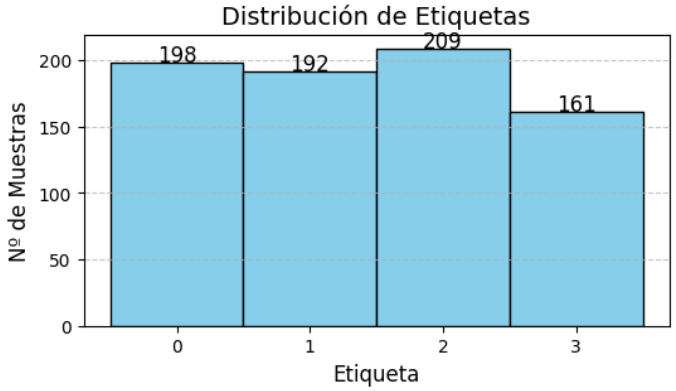

Figura 55. Distribución de etiquetas tras ajuste.

3.3.7 Evaluación de la segunda etapa de entrenamiento.

Tras la revisión y ampliación del dataset se repiten la etapa de entrenamiento y evaluación de modelos utilizando los mismos datos de pruebas para ver si se ha conseguido una mejora en los resultados.

- **Validación VGG11 con dropout 0.5**

Esta ocasión podemos observar en la figura 56 que el etiquetado ha mejorado considerablemente, detectando las señales de voces y ruido de motor.

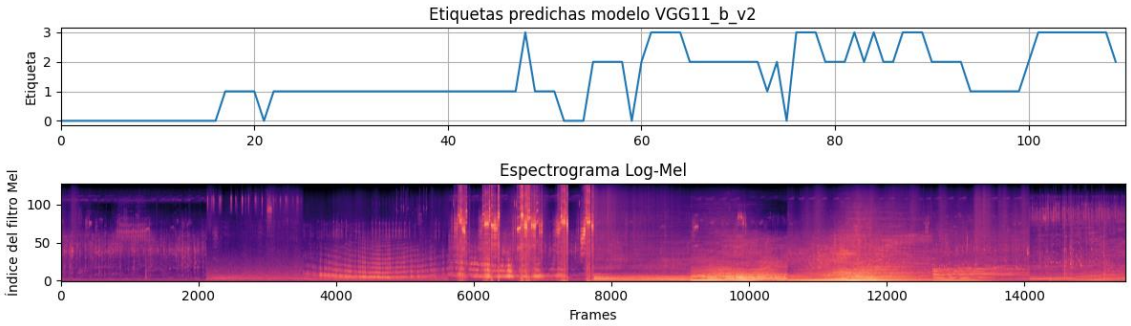

Figura 56. VGG11b_v2, prueba con dataset variado.

En la segunda prueba parece que el resultado empeora, parece detectar los aviones, pero al resto de paisaje acústico le asigna las etiquetas de voces o motor, lo cual está muy alejado de la realidad. Este resultado parece repetirse para el resto de las redes neuronales con esta grabación. Es posible que algunas muestras de voces se asemejen a estas muestras de paisaje.

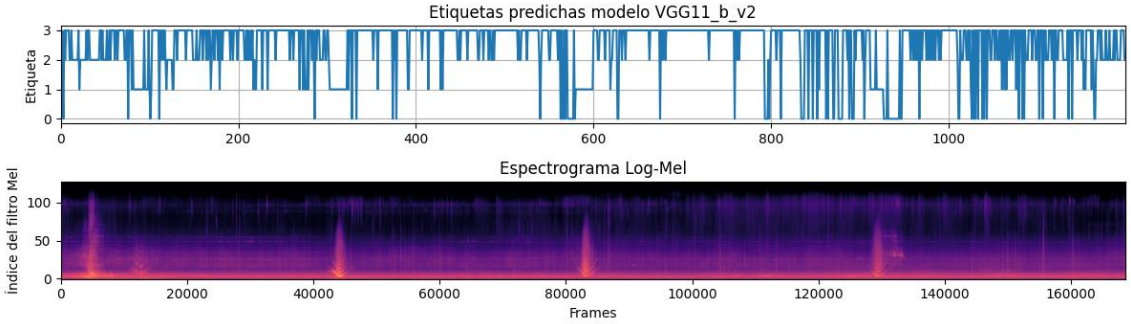

Figura 57. VGG11b_v2, segunda prueba con grabación de aviones y paisaje.

En la tercera prueba parece detectar adecuadamente las muestras de paisaje acústico y las muestras de avión, a excepción de alguna muestra puntual.

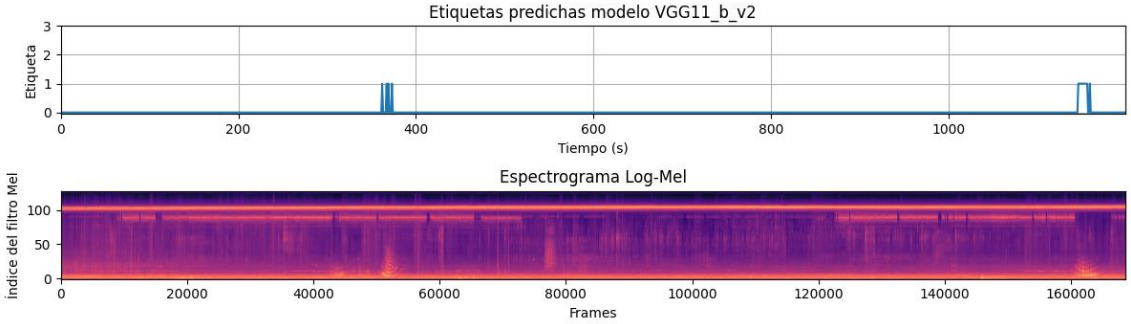

Figura 58. VGG11b_v2, tercera prueba con grabación de aviones y sonido de fondo.

- **Validación VGG13c con dropout 0.5**

En la primera muestra de nuevo se aprecia una mejora respecto a la misma red con el entrenamiento anterior, identificando las diferentes etiquetas y generando pocos falsos positivos, aunque estos siguen presentes.

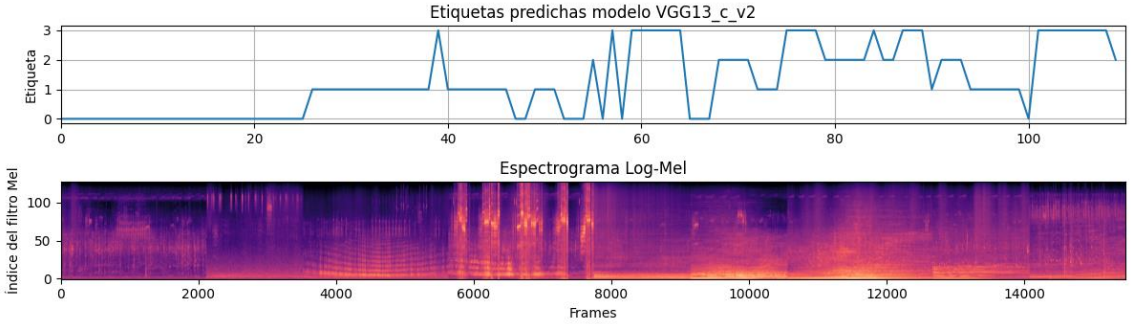

Figura 59. VGG13c_v2, prueba con dataset variado.

En esta segunda predicción se aprecia de nuevo el defecto a la hora de etiquetar el paisaje acústico como una mezcla entre sonidos de voces y motor, cuando en esta grabación no hay presencia de estas muestras.

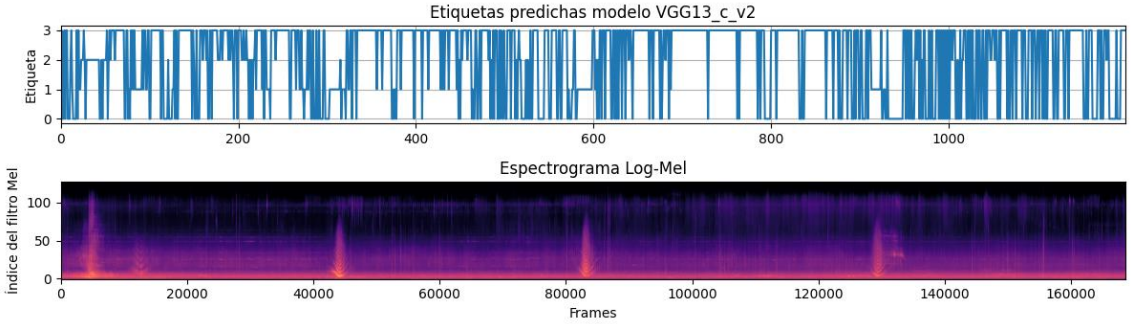

Figura 60. VGG13c_v2, segunda prueba con grabación de aviones y paisaje.

En la última prueba se detectan pocas muestras de los aviones, empeorando en esta parte respecto al entrenamiento anterior de este modelo.

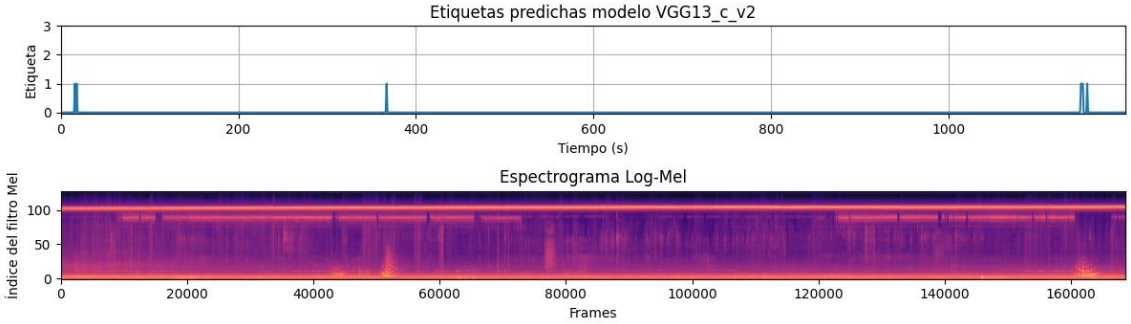

Figura 61. VGG13c_v2, tercera prueba con grabación de aviones y sonido de fondo.

Aunque presenta mejoras el modelo sigue generando falsos positivos.

- **Validación AlexNetb_v2**

Recordando los resultados obtenidos previamente con este modelo fueron prácticamente nulos en el entrenamiento anterior. Al volver a entrenar con la ampliación de muestras en el dataset se consigue una gran mejora en la clasificación de datos, pudiendo detectar con cierta precisión gran parte de las muestras.

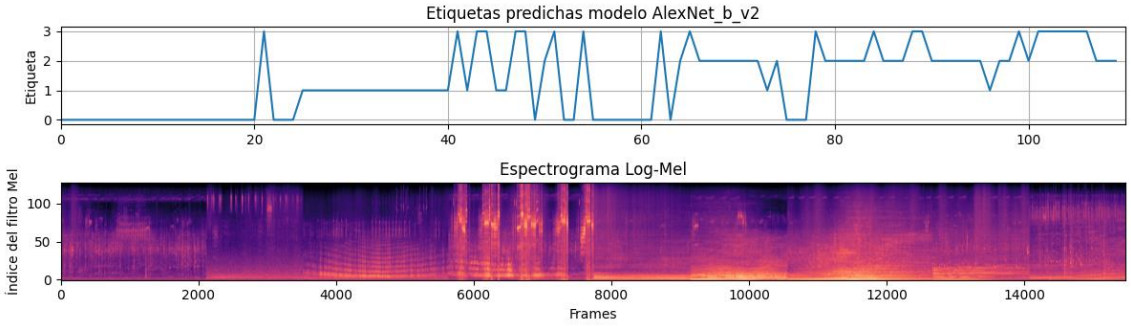

Figura 62. AlexNetb_v2, prueba con dataset variado.

En la segunda prueba parece que el primer avión no lo detecta adecuadamente y el resto de los aviones sí, aunque de nuevo el paisaje acústico no es correctamente etiquetado.

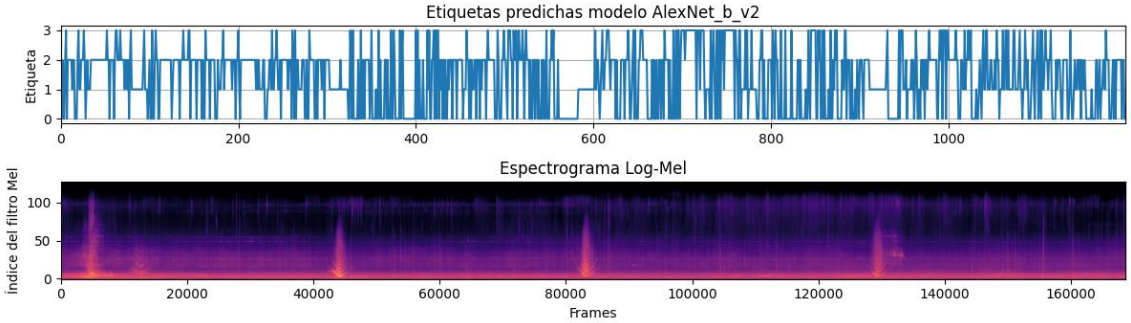

Figura 63. AlexNetb_v2, segunda prueba con grabación de aviones y paisaje.

En la tercera prueba de predicción parece que solo es capaz de detectar un avión, reconociendo sin problemas el resto de los sonidos como paisaje acústico.

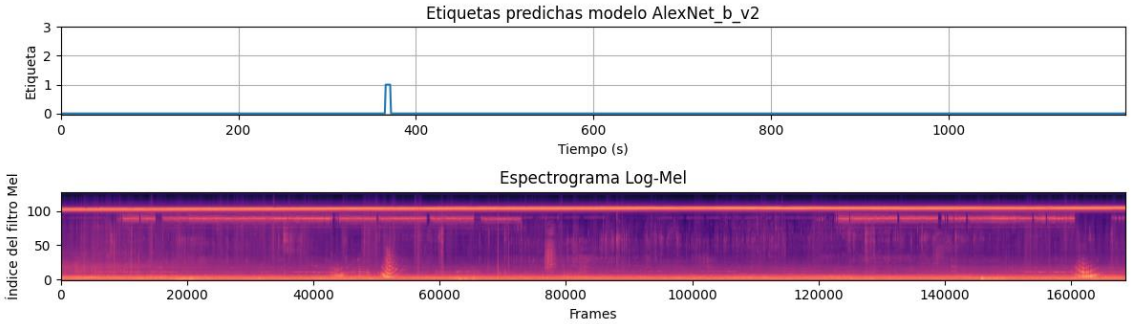

Figura 64. AlexNetb_v2, tercera prueba con grabación de aviones y sonido de fondo.

Aunque presenta grandes mejoras frente a la primera etapa de entrenamiento, sigue sin resolver el problema planteado.

- **Validación VGG16b_v2**

Al igual que con el resto de los modelos, se aprecian mejoras en las predicciones con el primer dataset. En este caso la VGG16b genera muchos falsos positivos con la etiqueta de voces, tal y como se ve en torno al segundo 40 de la figura 65.

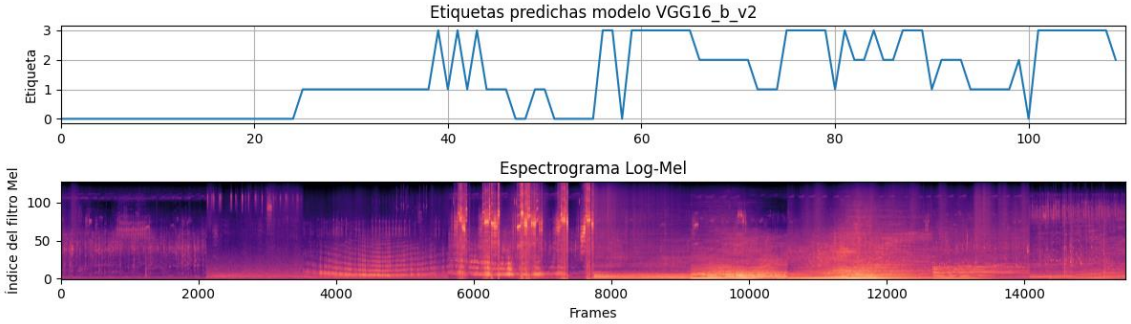

Figura 65. VGG16b_v2, prueba con dataset variado.

En la segunda prueba se repiten los resultados hasta ahora vistos con esta grabación, mucho falso positivo en voces, adicionalmente este modelo parece ser incapaz de detectar algunas muestras de avión.

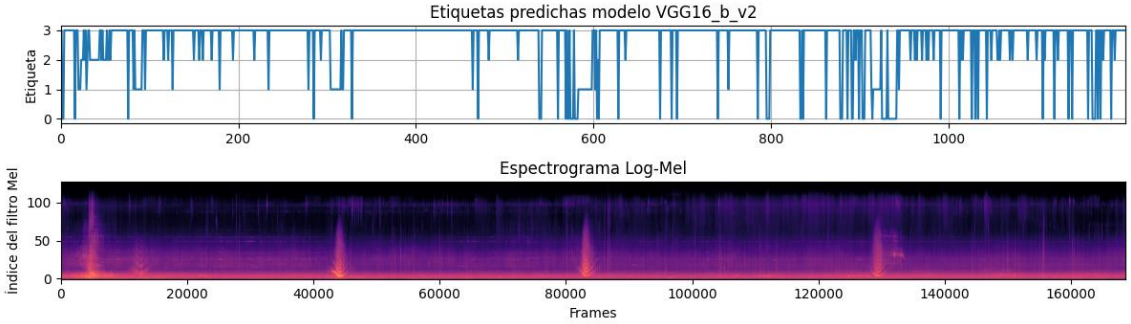

Figura 66. VGG16b_v2, segunda prueba con grabación de aviones y paisaje.

En la última prueba vemos que las muestras de avión son reconocidas, aunque no parece detectarlas completamente, se ve algún falso positivo y e resto etiquetado correctamente como paisaje acústico.

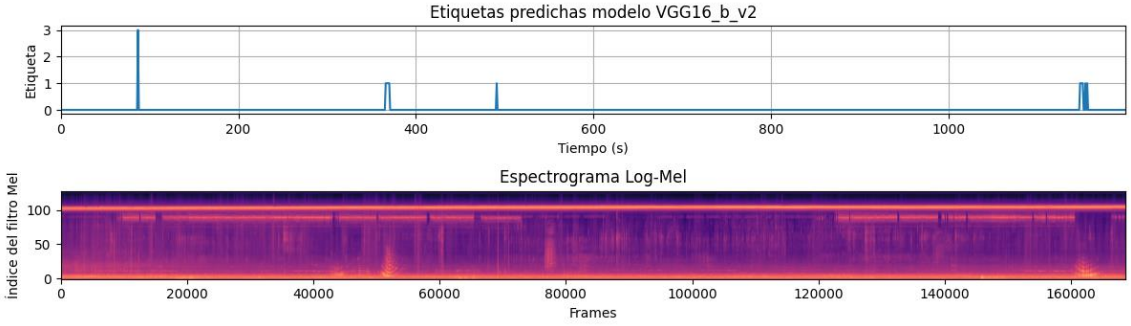

Figura 67. VGG16b_v2, tercera prueba con grabación de aviones y sonido de fondo.

- **Validación ResNet50_v2**

Con este último modelo también se observan grandes avances, en el entrenamiento anterior fue incapaz de etiquetar las diferentes muestras, indicando todo como paisaje acústico, gracias a los cambios se aprecia mayor variabilidad en las etiquetas. El avión parece detectarlo correctamente y las muestras de motor y voces parece asignar etiquetas con cierta precisión.

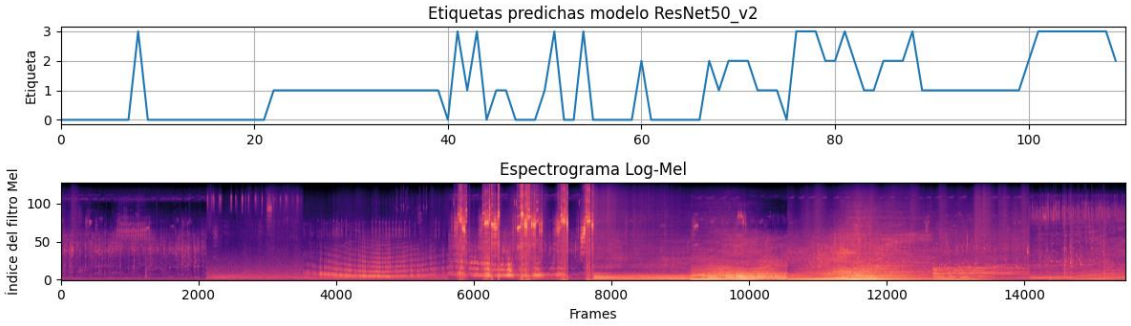

Figura 68. ResNet50_v2, prueba con dataset variado.

Con la segunda prueba se observa que, aunque genera mucha etiqueta de voces, parece ser capaz de etiquetar correctamente el paisaje acústico y de detectar los aviones. Sigue habiendo bastante falso positivo.

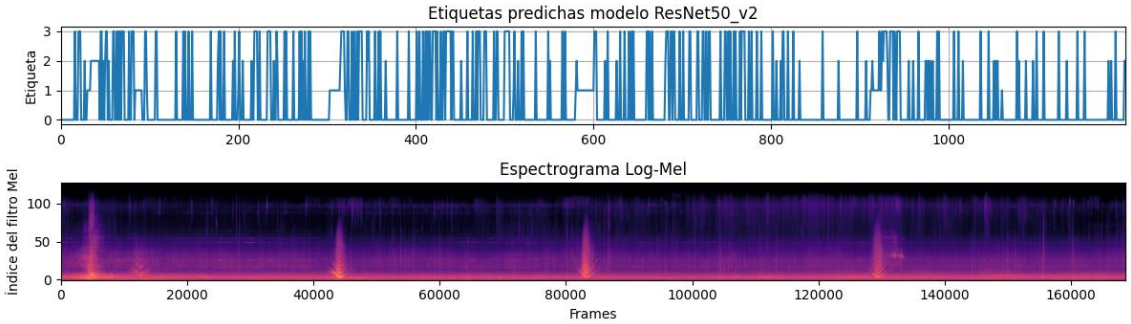

Figura 69. ResNet50_v2, segunda prueba con grabación de aviones y paisaje.

En la última prueba, figura 70, se observa que se detecta adecuadamente los aviones aunque sigue habiendo mucho falso positivo, ya que el resto debería estar con la etiqueta “0”.

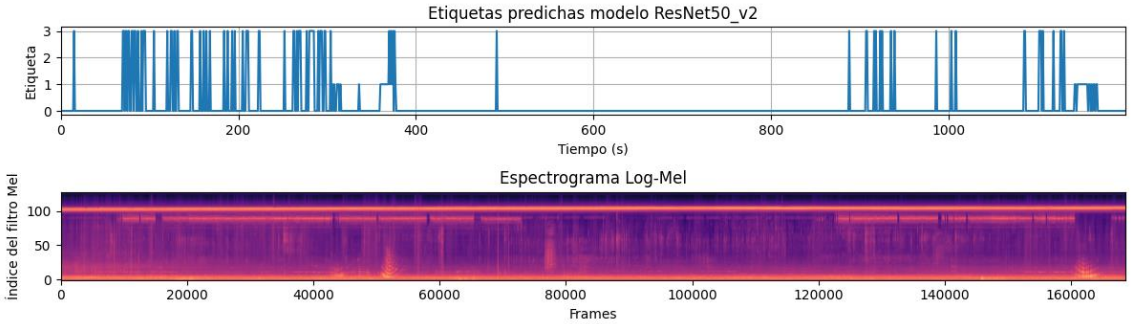

Figura 70. ResNet50_v2, tercera prueba con grabación de aviones y sonido de fondo.

Aunque presenta grandes avances se observan muchos falsos positivos.

3.4 Resultados

Tras entrenar y validar los modelos se aprecia una mejora en las predicciones gracias a la adición de nuevas muestras al dataset de entrenamiento. Con la primera muestra más variada vemos una mejora significativa especialmente en la modelo VGG11b respecto al primer entrenamiento. Esto se puede ver reflejado en la matriz de confusión de la figura 71.

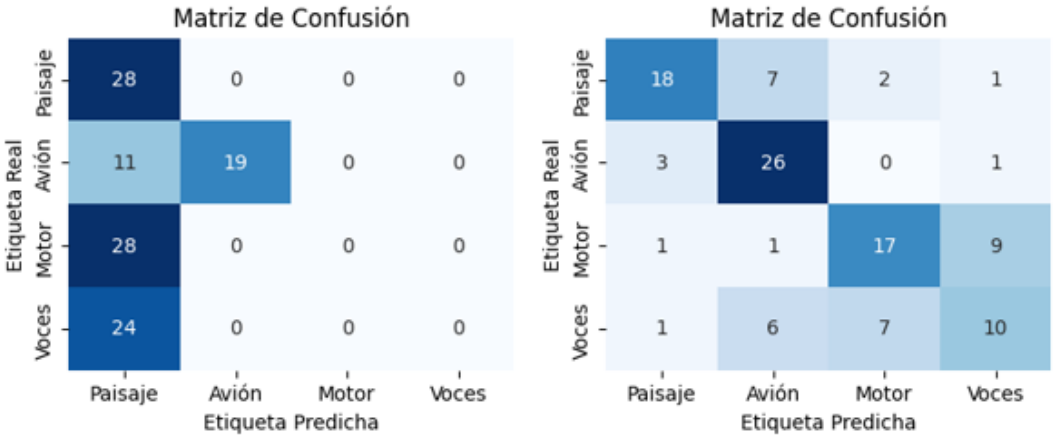

Figura 71. Predicción inicial (izquierda) y predicción final (derecha)

La ampliación del datasets de entrenamiento ha mejorado la precisión de los modelos permitiendo calcular una densidad de la presencia de ruido con origen humano, para cada una de las grabaciones comentadas en los pasos anteriores.

- **Resultados primera grabación**

Recordando el contenido, la grabación se compone un dataset variado, con muestras diversas de paisaje acústico, avión, motor y voces. En esta grabación se ha medido manualmente una densidad de ruido humano del 75%. Observando los resultados de las diferentes CNNs en la tabla 13 vemos que la VGG11 es capaz de obtener un 79% de densidad de ruido con una precisión del 65%.

Grabación variada	**VGG11**	VGG13	AlexNet	VGG16	ResNet50
Precisión de predicción	**65%**	60%	59%	58%	54%
Densidad de ruido humano	**79%**	66%	65%	69%	64%

Tabla 13. Resultados de procesar el dataset variado

- **Resultados segunda grabación**

En esta grabación tenemos una densidad de ruido de origen humano del 8%, con su contenido repartido en varias muestras de avión sobre un fondo de paisaje acústico muy suabe. Tal y como se ve en la tabla 14 que el modelo ResNet50 es el que tiene un mejor rendimiento, con una precisión general del 77% detectando un 27% de densidad de ruido humano.

Grabación 2	VGG11	VGG13	AlexNet	VGG16	**ResNet50**
Precisión de predicción	15%	31%	36%	12%	**77%**
Densidad de ruido humano	92%	75%	71%	93%	**27%**

Tabla 14. Resultados de procesar la segunda grabación

- **Resultados tercera grabación**

Esta última grabación es una muestra de una hora formada por paisaje acústico con un ruido de fondo constante de lo que parecen ser grillos, a lo largo de la grabación se tienen 2 muestras de avión, teniendo esto último una densidad de 3% en la grabación completa. De los 5 modelos el más preciso ha sido la VGG11, detectando un 1% de densidad de ruido humano. El resto de los modelos presentan falsos positivos y una menor precisión.

Grabación 3	**VGG11**	VGG13	AlexNet	VGG16	ResNet50
Precisión de predicción	**98%**	97%	97%	97%	91%
Densidad de ruido humano	**1%**	1%	1%	2%	11%

Tabla 15. Resultados de procesar la tercera grabación

Capítulo 4. Conclusiones y líneas futuras

4.1 Conclusiones

Este proyecto se ha centrado en el análisis de grabaciones obtenidas de una serie de nodos instalados a lo largo del Parque Natural de l'Albufera, en Valencia, con objetivo de desarrollar un motor capaz de extraer y detectar los sonidos de origen humano para facilitar el estudio del impacto de este en la fauna local. Para ello se han realizado pruebas con modelos no supervisados y redes neuronales convolucionales para validar su potencial.

En una primera etapa se ha centrado en los métodos no supervisados. Durante este proceso se ha visto que el primer factor relevante es la elección de características adecuadas, una correcta elección de las características aumenta la información reduciendo la cantidad de datos a utilizar. También se ha demostrado como las técnicas de reducción de dimensionalidad son muy efectivas para la visualización y análisis inicial de los datos, en concreto PCA, al ser un modelo fácil de utilizar y que mantiene gran cantidad de información reduciendo la dimensión de los datos. También se ha comprobado cómo, en conjunto al PCA, el modelo K-Means es capaz de distinguir matices entre los datos y realizar una agrupación muy cercana a la realidad, aunque debido a la naturaleza de los datos el número de clases necesario era ligeramente superior al deseado ya que la presencia de ciertos sonidos puede alterar el resultado de la predicción, como por ejemplo en muestras de avión que contienen ruido elevado de pájaros genera una etiqueta diferente tanto a la asociada al ruido de avión como a la de pájaro.

En una segunda etapa se ha estudiado el uso de diferentes redes neuronales convolucionales para intentar reducir el número de clases y aumentar la precisión en las predicciones. En este caso se ha demostrado que el espectrograma Mel es una característica adecuada para poder realizar el estudio, por lo que la importancia recae tanto en la elección de las muestras como en su cantidad y correcto etiquetado. Dada la diversidad de sonidos que encontramos en la naturaleza, el proceso de selección de muestras para el entrenamiento se vuelve más delicado y extenso, ya que requiere disponer de la suficiente variedad correctamente etiquetada para poder entrenar las redes neuronales de forma adecuada.

Evaluación del cumplimiento de objetivos

En la última etapa de este proyecto se ha podido realizar un análisis de varias muestras con las redes neuronales resultantes. Se ha hecho una evaluación de la densidad del sonido humano en dichas grabaciones, para las cuales se ha comprobado la precisión en la predicción respecto a un análisis hecho a mano, llegando a precisiones en un dataset variado del 65%. Con esto se cumplen los objetivos propuestos en el punto 1.3 de la memoria. Aunque la precisión no es la deseada, es posible mejorarla continuando la línea de trabajo en un futuro.

Impacto y alcance

El proceso seguido ha quedado totalmente documentado en un repositorio de libre acceso en github https://github.com/Damarde/TFM_Clasificador_Sonidos_con_ML [7], por lo que la comunidad podrá utilizar las técnicas y procedimientos empleados para que sirva de guía y apoyo en investigaciones futuras.

4.2 Líneas futuras

La principal línea de desarrollo a futuro sería la ampliación y mejora del dataset de entrenamiento para las redes neuronales. Como se ha podido comprobar, las redes neuronales son capaces de distinguir sonidos con pocas muestras de entrenamiento, al ampliar el número de las muestras se ha conseguido una mejora importante en los resultados de las predicciones, aunque sigue siendo insuficiente. Como se ha demostrado en los últimos puntos, una elección más extensa y cuidada mejoraría enormemente los resultados obtenidos y le permitiría una mayor robustez al detectar sonidos de origen humano frente al resto de sonidos naturales presentes.

Bibliografía

[1] Mesaros, Annamaria & Heittola, Toni & Diment, Aleksandr & Elizalde, Benjamin & Shah, Ankit & Vincent, Emmanuel & Raj, Bhiksha & Virtanen, Tuomas. (2017). DCASE 2017 CHALLENGE SETUP: TASKS, DATASETS AND BASELINE SYSTEM.

[2] Siadati, Saman. (2018). What is unsupervised Learning. 10,13140/RG.2.2.33325.10720,

[3] Liu, Qiong & Wu, Ying. (2012). Supervised Learning. 10,1007/978-1-4419-1428-6_451.

[4] Matplotlib. "Matplotlib API reference.". https://matplotlib.org/stable/api/. [Online].

[5] Librosa Development Team. Librosa: Audio and music signal analysis in Python. https://librosa.org/doc/latest/. [Online].

[6] SciPy Developers. SciPy: Scientific Computing Tools for Python. https://docs.scipy.org/doc/scipy/. [Online].

[7] Martínez, David. "TFM_Clasificador_Sonidos_con_ML" Repositorio de GitHub https://github.com/Damarde/TFM_Clasificador_Sonidos_con_ML

[8] Jeon, Hohyub & Jung, Yongchul & Lee, Seongjoo & Jung, Yunho. (2020). Area-Efficient Short-Time Fourier Transform Processor for Time–Frequency Analysis of Non-Stationary Signals. Applied Sciences. 10, 7208. 10,3390/app10207208.

[9] Kogan, Pablo. (2004). Análisis de la Eficiencia de la Ponderación "A" para Evaluar Efectos del Ruido en el Ser Humano. 10,13140/RG.2.2.29133.69607.

[10] Matlab, documentation. Mel spectrogram. https://es.mathworks.com/help/audio/ref/melspectrogram.html [Online]

[11] Zinemanas, Pablo. (2019). Herramientas computacionales para el análisis del entorno sonoro urbano.

[12] Vines, Greg & Nemer, Elias. (2022). Comparison of Audio Spectral Features in a Convolutional Neural Network.

[13] Scheirer, E., and M. Slaney. "Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator." IEEE International Conference on Acoustics, Speech, and Signal Processing. Volume 2, 1997, pp. 1221–1224.

[14] Chambi, Pedro. (2023). Segmentación de mercado: Machine Learning en marketing en contextos de covid-19. Industrial Data. 26. 275-301. 10,15381/idata.v26i1.23623.

[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96). AAAI Press, 226–231.

[16] Contreras, Stevenson & De la Rosa, Fernando. (2016). Aplicación de Deep Learning en Robótica Móvil para Exploración y Reconocimiento de Objetos basados en Imágenes. 10,1109/ColumbianCC.2016.7750800,

[17] Kong, Qiuqiang & Xu, Yong & Plumbley, Mark. (2018). A Joint Separation-Classification Model for Sound Event Detection of Weakly Labelled Data. 10,1109/ICASSP.2018.8462448.

[18] N. P. García-de-la-Puente, F. Fuentes-Hurtado, L. Fuster, V. Naranjo, and G. Pinero, "Deep Learning Models for Gunshot Detection in the Albufera Natural Park," ITEAM, KNODIS Research Group, Dep. Sistemas Informaticos, Universidad Politécnica de Madrid, I3B, Universitat Politécnica de Valencia, 2022, doi: 10,13039/501100011033.

[19] Kumar, Ajitesh. "Different Types of CNN Architectures Explained: Examples." Analytics Yogi, 4 de diciembre de 2023. https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/#VGGNet_%E2%80%93_CNN_Architecture_with_Large_Filters. [Online]