Detección de Manipulaciones Copy-Move y Splicing en Audio usando Técnicas de Aprendizaje Profundo

Detection of Audio Copy-Move and Splicing Forgery using Deep Learning Techniques

TRABAJO FIN DE GRADO
CURSO 2023–2024

José Antonio Ruiz Heredia (Ingeniería Informática)
Néstor Antonio Marín Gómez (Ingeniería de Computadores)

Directores
Luis Javier García Villalba
Daniel Povedano Álvarez

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Madrid, Mayo de 2024

Agradecimientos

Primero de todo, agradecer tanto a nuestros directores, Daniel Povedano Álvarez y Luis Javier García Villalba, y a Ana Lucila Sandoval Orozco, por toda la ayuda que se nos han proporcionado durante el proceso introductorio al tema que se iba a tratar y a lo largo de la realización de este trabajo. Además, queríamos destacar la supervisión y recomendaciones que se nos brindaron durante el curso, lo que nos permitió continuar de manera eficiente el proyecto sin parones ni estancamientos.

A su vez, también queríamos hacer una apreciación a la disposición de herramientas necesarias para el estudio, como un ordenador capaz de realizar el proceso de aprendizaje debido a que nuestros equipos personales no cumplían con las especificaciones requeridas.

Segundo, a nuestras familias y amigos que han estado presentes durante el desarrollo del mismo y que nos han apoyado durante esta etapa. La curiosidad y entusiasmo que algunos conocidos nuestros han presentado sobre la temática que estábamos tratando ha mejorado nuestra motivación y ambición de obtención del conocimiento requerido por el proyecto

Tercero, cabe destacar que este proceso no sería posible sin la existencia de artículos previos donde se habían realizado aproximaciones que, aunque utilizaran otros métodos o buscaran objetivos diferentes, constituyeron el pilar sobre el que basar nuestro planteamiento para el desarrollo del modelo y de su consecuente experimentación.

De la misma forma, agradecer a los numerosos creadores del contenido que revisamos para introducirnos en el marco teórico del trabajo puesto que, de la misma forma que los artículos, sirvieron de fundamento necesario para poder comenzar a realizar nuestro estudio.

Por último, resaltar el trabajo de todos los desarrolladores de las herramientas que han sido mencionadas y con las que realizamos el grueso de este trabajo.

Índice General

Índice de Figuras

Índice de Tablas

Índice de Algoritmos

Lista de Acrónimos

ADA	*Audio Data Augmentation*
ADL	*Audio Deep Learning*
AI	Artificial Intelligence
ANN	*Artificial Neural Network*
ANS	Aprendizaje No Supervisado
AR	Aprendizaje Por Refuerzo
AS	Aprendizaje Supervisado
ASR	*Automatic Speech Recognition*
ASS	Aprendizaje Semi Supervisado
CCP	Coeficiente de Correlación de *Pearson*
CE	Comisión Europea
CQCC	*Constant Q Cepstral Coefficients*
CSV	*Comma Separated Values*
DL	*Deep Learning*
EC	*European Comission*
FCMA	*Fully Convolutional Masked Autoencoder*

FFT	*Fast Fourier Transform*
FN	Falsos Negativos
FP	Falsos Positivos
GASS	Grupo de Análisis, Seguridad y Sistemas
GPU	*Graphics Processing Units*
Grad-CAM	*Gradient-weighted Class Activation Mapping*
IA	Inteligencia Artificial
LFCC	*Linear Frequency Cepstral Coefficients*
MAE	*Masked Autoencoders*
MFCC	*Mel Frequency Cepstral Coefficients*
MIT	*Massachusetts Institute of Technology*
ML	*Machine Learning*
RANSAC	*Random Sample Consensus*
RGB	*Red-Green-Blue*
RNA	Red Neuronal Artificial
RNC	Red Neuronal Convolucional
SIFT	*Scale-Invariant Feature Transform*
STFT	*Short-time Fourier Transform*
SW	*Sliding Window*

TFG	Trabajo Fin de Grado
TIMIT	*Texas Instruments / Massachusetts Institute of Technology*
TPU	*Tensor Processing Units*
UCM	Universidad Complutense de Madrid
VAD	*Voice Activity Detection*

Abstract

In recent years, the technological improvement and the growing use of social media has taken part in the proliferation of numerous counterfeit multimedia content in our environement. Moreover, due to the sophistication of the methodologies employed, it has progressively become more difficult to detect the irregularities in these files.

This work proposes a solution for the detection of these manipulations, specifically focusing on audio files. To achieve this, we follow the approach of deep learning algorithms, presented as the best solution among the existing artificial intelligence methods.

For this study, we performed a conversion of the original and modified samples into images called spectrograms. This conversion is carried out in order to perform a preprocessing phase of the samples and, then, satisfy the correct feature extraction and learning process by the model. The architecture of the model, represented as a neural network, is capable of identifying these labeled samples in a training and validation process.

These models are characterized by their robustness and scalability, which allows the inclusion of various categories of forgeries in the study. Concretely, this work presents a classification into three different categories.

The experiments illustrates that this type of analysis has achieved excellent results with machine learning algorithms and, therefore, represents an efficient approach.

Keywords: Audio, Deep Learning, Feature Extraction, Forgeries, Labeled Samples, Neural Network, Preprocessing, Spectrogram.

Resumen

En la actualidad, con el desarrollo tecnológico de los últimos años y el creciente uso de la redes sociales, es cada vez más frecuente la existencia de falsificaciones de archivos multimedia que se comparten y circulan en nuestro entorno. Además, debido a la sofisticación de las metodologías empleadas, cada vez es más difícil detectar las irregularidades en estos archivos.

En este trabajo se propone una solución para la detección de estas manipulaciones, en concreto centrándonos en archivos de audio. Para ello seguimos el enfoque de los algoritmos de aprendizaje profundo, presentados como la mejor solución de entre los métodos de inteligencia artificial existentes.

Para este estudio se realiza una conversión de las muestras originales y modificadas en imágenes denominadas espectrogramas. Esta conversión se desarrolla con el objetivo de realizar un preprocesamiento para la correcta extracción de características y aprendizaje del modelo. La arquitectura del mismo, representada como una red neuronal, es capaz de identificar estas muestras etiquetadas en un proceso de entrenamiento y validación.

Estos modelos se caracterizan por su robustez y escalabilidad, lo que permite la inclusión de varias categorías de alteración de muestras al estudio. Concretamente, este trabajo presenta una clasificación en tres categorías distintas.

Los experimentos demuestran que este tipo de análisis tiene resultados excelentes con algoritmos de aprendizaje automático y que, por lo tanto, suponen un enfoque eficiente.

Palabras clave: Aprendizaje Profundo, Audio, Espectrograma, Extracción de Características, Manipulaciones, Muestras Etiquetadas, Preprocesamiento, Red neuronal.

Capítulo 1

Introducción

1.1. Motivación

La ciencia forense del audio comenzó con la invención del fonógrafo por *Thomas Edison* en el año 1877. Esta tecnología permitió grabar y reproducir sonido a través de un cilindro de cera, marcando un hito en la comunicación. Posteriormente, a lo largo del siglo XX, los avances tecnológicos condujeron a una mejora a la hora de realizar grabaciones, consiguiendo que estas fueran cada vez mas nítidas y precisas. Todo ello supuso un enorme adelanto en el uso del audio como herramienta de investigación.

Una de las primeras aplicaciones de la ciencia forense de audio fue durante la Primera Guerra Mundial. Los avances tecnológicos en esa época permitieron a los soldados y comandantes capturar sonidos de la batalla y transmisiones de noticias en discos fonográficos. Estas grabaciones fueron de vital importancia en la investigación sobre las responsabilidades sobre el inicio de la guerra, ya que se le acusaba a Alemania de empezar el conflicto bélico. Entonces se formó un grupo de trabajo especializado para analizar miles de grabaciones de conversaciones grabadas entre diplomáticos y políticos. Al evaluar esas grabaciones, se pudo reunir evidencias para apoyar o refutar esas acusaciones.

Esta ciencia también ha sido de gran utilidad en investigaciones criminales. Un caso histórico tuvo lugar en 1932, cuando la policía alemana, mediante un meticuloso análisis de sonido, logró identificar una única voz presente en dos grabaciones de la escena del crimen. Esta evidencia condujo a la incriminación del responsable.

Estos ejemplos demuestran la importancia de la ciencia forense del audio en la validación de testimonios y el fortalecimiento de argumentos legales. A medida que la tecnología continúa evolucionando a un ritmo vertiginoso, las técnicas y aplicaciones de esta disciplina también se expanden a un ritmo acelerado. Sin embargo, esta evolución no está exenta de desafíos, ya que exige una adaptación constante por parte de los profesionales y las instituciones involucradas.

A lo largo de los años, la ciencia forense ha experimentado tanto éxitos como fracasos, ya que en las grabaciones de gran calidad existen elementos que pueden dificultar el estudio preciso de las características originales del audio. Estos desafíos se intensifican aún más en grabaciones de menor calidad o aquellas que han sido alteradas o manipuladas.

El siglo XXI ha sido testigo del ascenso meteórico de la ciencia forense del audio como una herramienta fundamental para la búsqueda de la verdad y la justicia. A

medida que la tecnología avanza y los dispositivos de grabación se vuelven cada vez más omnipresentes, las posibilidades de capturar y analizar evidencia de audio se multiplican exponencialmente. Las fuerzas del orden, los investigadores y los profesionales del derecho han encontrado en la ciencia forense del audio un gran aliado. Su capacidad para identificar voces, analizar contenido acústico y extraer información de grabaciones de audio de diversa calidad la convierte en una herramienta indispensable en una amplia gama de casos.

La existencia de los teléfonos inteligentes, grabadoras digitales y otros dispositivos de captura de sonido ha generado un aumento exponencial de las evidencias acústicas disponibles. Esta abundancia de datos, si bien ofrece un sinfín de posibilidades para la investigación, también presenta desafíos en cuanto a su manejo, análisis y autenticación.

Sin embargo, el aumento de las evidencias acústicas y la delgada línea entre la verdad y la privacidad exigen un manejo ético y responsable de esta disciplina para garantizar el respeto a los derechos individuales. El futuro de la ciencia forense del audio es prometedor, con un enorme potencial para seguir evolucionando y aportando a la justicia en un mundo cada vez más digitalizado.

Por otro lado, cuando nació la ciencia forense de audio, el análisis de las grabaciones de sonido recaía sobre la capacidad auditiva y la experiencia de los expertos humanos. En los dos ejemplos anteriores que se han citado, sobre la Primera Guerra Mundial, y el juicio del año 1932, estos expertos se encargaron de identificar manualmente características como la voz, el ruido ambiental, etc. Sin embargo este enfoque manual no estaba exento de limitaciones, ya que existían diversos riesgos que podían afectar a la precisión de esos análisis, como la subjetividad de la percepción auditiva, la fatiga auditiva, etc.

Por otro lado, la IA en sus principios no era útil para la extracción e identificación de esas características. Pero en el año 1958, la aparición del *Perceptron* [Sim20] marcó un hito en el mundo de la IA, ya que sentó las bases para el desarrollo de sistemas computacionales capaces de procesar y analizar información de manera similar al humano. Desde entonces, la IA, con la ayuda de los avances tecnológicos, comenzó a crecer a una velocidad vertiginosa, y con ello sus aplicaciones en el mundo de la ciencia forense de audio.

Esto implicó que ahora se pudieran usar las RNA en la detección, tanto de las características ya existentes en esos audios, como de nuevas características difícilmente perceptibles por seres humanos, ya que estas redes están inspiradas en el funcionamiento del cerebro humano, y pueden procesar grandes cantidades de datos y aprender a identificar patrones complejos con mayor precisión que los expertos humanos.

Por lo que, con la ayuda de la IA, la tarea del analisis y extracción de caracteristicas propias de un audio, para su posterior decision de si ese audio es original, o bien ha sido manipulado con algun proposito delictivo. Para ello solamente sería necesario transformar ese audio a imagen, y utilizar esa imagen como entrada de ejemplo para que la inteligencia artificial pudiese entrenar y aprender a detectar cuando un audio es original o no.

1.2. Contexto

El presente Trabajo Fin de Grado (TFG) se en marca dentro de un proyecto de investigación titulado *Child protection centred strategies to fight against sexual abuse and exploitation – ALUNA*, aprobado por la Comisión Europea (CE) en la convocatoria ISF-2021-TF1-AG-CYBER en virtud del acuerdo de subvención número 101084929 y en

el que participa como coordinador del proyecto el Grupo de Análisis, Seguridad y Sistemas (GASS) de la Universidad Complutense de Madrid (UCM) (Grupo 910623 del catálogo de grupos de investigación reconocidos por la UCM, https://gass.ucm.es).

1.3. Objeto de la Investigación

A la hora de escuchar un sonido, las ondas sonoras son captadas por el pabellón auricular y conducidas a través del canal auditivo hasta el tímpano. Este fino tejido vibra al ritmo de las ondas, transmitiendo la información acústica al oído medio, y ahí la energía mecánica del sonido es transformada en señales eléctricas, que viajan a través del nervio auditivo hasta llegar al cerebro, donde es procesada toda la información acústica perteneciente a ese sonido. Esto implica la dificultad de diferenciación entre la originalidad o falsificación de un audio, ya que puede localizarse en un punto imperceptible al oído humano.

El siguiente trabajo va a enfocarse en intentar conseguir un modelo que extraiga las características de un audio de la mejor manera, para después poder pasárselo a un modelo de IA, con el objetivo de entrenarla y que aprenda lo mejor posible a detectar si un audio es original o modificado.

1.4. Plan de Trabajo

El desarrollo de este trabajo se ha realizado en tres fases:

1. **Investigación**:

   Los primeros pasos de la realización de este trabajo, los cuales tuvieron lugar los primeros 4 meses, consistieron en la toma de contacto sobre el trabajo a realizar.

   Primeramente se tuvo una reunión inicial en la que nos reunimos los profesores del departamento y los alumnos, en la que se explicó el punto de partida, los objetivos del trabajo, los conocimientos necesarios para el desarrollo de este, y las primeras directrices a tener en cuenta antes de realizar el estudio. Posteriormente, se llegó a un consenso y se acordaron unas reuniones periódicas para revisar el estado del desarrollo del trabajo. También se nos facilitó un espacio en *Google Drive* sobre el que deberíamos realizar nuestro trabajo, incluyendo numerosos recursos por si necesitábamos consultar material de apoyo para afianzar algunos conceptos necesarios para continuar con nuestro desarrollo.

   Aunque se tenía acceso al material de apoyo, fue necesario buscar más recursos para consultar, por lo que se revisaron los trabajos existentes relacionados con nuestra investigación, y se resumieron varios de esos trabajos,revisando el conjunto de datos utilizado, los métodos propuestos, las técnicas de experimentación, y los resultados y conclusiones de cada uno.

   Finalmente, una vez obtenidas varias conclusiones acerca de qué se quería y cómo se quería conseguir, el equipo comenzó a dar prioridad al desarrollo.

2. **Desarrollo**:

   Una vez adquiridos los conocimientos necesarios, y teniendo claro las metodologías existentes, sin dejar de lado la investigación, se comenzó a desarrollar el trabajo

propuesto. Este proceso duró en torno a los tres meses. Comenzamos, primeramente, eligiendo un conjunto de datos de audios, y desarrollando un código que lo alterase aplicándole las técnicas de modificación que se van a estudiar a lo largo del trabajo.

Posteriormente se empezó con la experimentación sobre esos datos para tener claro el formato y los parámetros más eficientes para representarlos, transformarlos, y modificarlos con el objetivo de proporcionárselos al modelo de *Deep Learning* (DL).

Durante esta fase se investigaron conceptos avanzados del lenguaje de programación de *Python* y de librerías como *Matplotlib*, *Librosa*, *TensorFlow*, *PyTorch*, etc.

3. **Experimentación**:

   Finalmente, cuando ya se había completado el análisis sobre el preprocesado de datos, se comenzó con la experimentación del modelo elegido con las muestras preprocesadas, con el objetivo de conseguir unos resultados superiores a los trabajos consultados.

   Durante este periodo de experimentación, se realizaron los ajustes necesarios sobre los parámetros que le proporcionábamos al modelo propuesto para conseguir las mejores métricas posibles.

1.5. Estructura del Trabajo

El resto del trabajo está organizado en 9 capítulos con la estructura que se comenta a continuación:

El Capítulo 1 hace una introducción del trabajo, indicando la motivación a realizarlo, el contexto, el objeto de la investigación y el plan de trabajo realizado.

El Capítulo 2 contiene una explicación del marco teórico y los conceptos necesarios para la comprensión del trabajo. Este capítulo se divide en dos partes: por un lado las técnicas y procesos involucrados en el aprendizaje del modelo y, por otro lado, la representación y tratamiento de audio para poder manejar los datos del estudio.

El Capítulo 3 comienza dando una introducción sobre el estudio de las modificaciones en audio. A continuación, se explica que es una manipulación de audio, los objetivos de esta, y los tipos de modificaciones según la parte que se modifica. Después, se profundiza en el tipo de modificación en la que se basa el trabajo, y se hace una recopilación de los trabajos existentes mas consultados en la etapa de investigación. De estos trabajos se extrae de manera resumida el conjunto de datos utilizado, el método propuesto, la experimentación realizada, y los resultados o conclusiones. Finalmente, se desarrollan las limitaciones encontradas en esos trabajos que han sido consultados, con el objetivo de mejorarlas en nuestro trabajo propuesto.

El Capítulo 4 muestra el proceso completo realizado, primeramente explicando la metodología aplicada para la creación de un conjunto de datos propio sobre el que trabajar, explicados en el algoritmo 1. Después se explica el proceso de preprocesamiento de audio junto con imágenes que ilustran la diferencia de resultados según los valores de los parámetros. También se explican las técnicas de ADA utilizadas, en la figura **??**

El Capítulo 5 describe los experimentos realizados para evaluar la efectividad de los algoritmos propuestos en el capítulo 4 y presenta los resultados obtenidos.

El Capítulo 6 muestra las principales conclusiones de este trabajo, las líneas futuras de investigación y las publicaciones derivadas del presente trabajo.

El Capítulo 7 recoge las aportaciones personales en concreto de cada alumno de una manera mas puntual, teniendo en cuenta que el trabajo ha sido correctamente repartido entre los alumnos que han realizado el trabajo.

Los Capítulos 8 y 9 son las traducciones al inglés del Capítulo 1 y del Capítulo 6.

Capítulo 2

Contexto de la Investigación

En este Capítulo 2 se condensa el trasfondo teórico y los conceptos necesarios para tratar el objeto del estudio. Dividimos el enfoque en dos campos principales a ser contextualizados, por un lado las técnicas y procesos involucrados en el aprendizaje del modelo y, por otro lado, la representación y tratamiento de audio para poder manejar los datos del estudio.

2.1. Inteligencia Artificial

La IA plantea una dificultad a la hora de definirse ya que implica comprender tanto el significado de lo **artificial** como de lo **inteligente** [Fer95]. En cuanto a que sea artificial, se refiere a un sistema de creación humana con una finalidad, y que, además, posee características no presentes en entidades naturales. Por otro lado, la definición de inteligencia se presenta como la capacidad de aprender, razonar y responder eficazmente ante nuevas situaciones.

Por ello, para brindar una definición más detallada, la IA es la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones tal y como lo haría un ser humano.

De esta forma, es capaz de ofrecernos sugerencias y predicciones en una multitud de áreas como la salud, el bienestar, la educación, el trabajo o las relaciones interpersonales.

2.1.1. Aprendizaje Automático

El **Aprendizaje Automático** o *Machine Learning* (ML), por sus siglas en inglés, es uno de los enfoques principales de la IA. Constituye el concepto de una máquina que es capaz de aprender sin estar programada con una serie de reglas estrictas para producir el resultado [JM15].

En este enfoque se utilizan diferentes algoritmos para aprender de los patrones de datos y utilizar el conocimiento adquirido en la toma de decisiones.

2.1.2. Tipos de Aprendizaje

Entre los principales enfoque de aprendizaje dentro del marco del ML encontramos los siguientes.

- **Aprendizaje Supervisado**: En el Aprendizaje Supervisado (AS) se usan datos que ya han sido etiquetados u organizados previamente para indicar cómo tendría que ser categorizada la nueva información. Con este método, se requiere la intervención humana para proporcionar retroalimentación en el aprendizaje.
- **Aprendizaje no Supervisado**: Por el contrario, en los algoritmos de Aprendizaje No Supervisado (ANS) no se usa ningún dato etiquetado u organizado previamente para indicar cómo tendría que ser categorizada la nueva información, sino que tienen que encontrar la manera de clasificarlas ellos mismos. Por tanto, este método no requiere de una intervención humana.
- **Aprendizaje Semi Supervisado**: En el caso del Aprendizaje Semi Supervisado (ASS) se utilizan algoritmos de ANS para extraer características relevantes y representaciones útiles de los datos no etiquetados, y posteriormente se utilizan estos conocimientos para mejorar la calidad del modelo de AS.
- **Aprendizaje por Refuerzo**: Alternativamente, en el Aprendizaje Por Refuerzo (AR), los algoritmos aprenden de la experiencia por medio de respuestas del sistema en base a la acción realizada. En otras palabras, requieren de un refuerzo positivo cada vez que aciertan y una penalización cada vez que fallan.

2.1.3. Aprendizaje Profundo

Una de las aplicaciones más poderosas y de mayor crecimiento de la IA es el **Aprendizaje Profundo** o DL, por sus siglas en inglés. Se trata de un subcampo del ML que se utiliza para resolver problemas muy complejos y que normalmente implican grandes cantidades de datos [Agg18].

De esta forma, podemos enmarcar los distintos tipos de IA según la Figura 2.1.

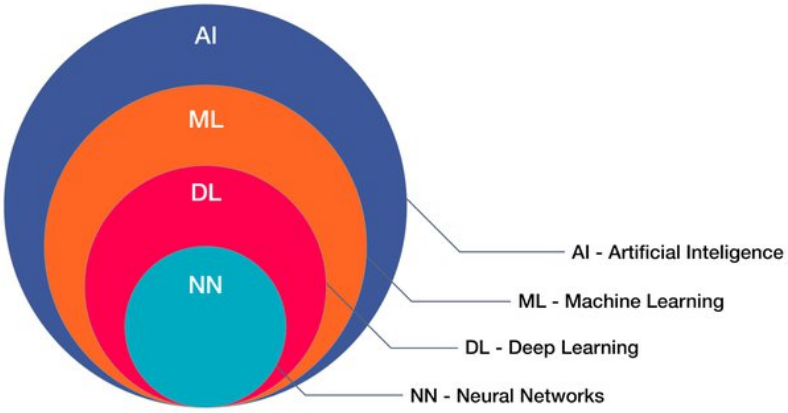

Figura 2.1: Clasificación de tipos de IA.

El DL se desarrolla mediante la implementación de una RNA, la cual consiste en un modelo computacional inspirado en la estructura y funcionamiento del cerebro humano.

Se compone de una red de nodos interconectados, llamados neuronas artificiales o **perceptrones**, que procesan información de entrada y generan una salida. Mediante esta arquitectura buscamos replicar ciertos procesos propios del cerebro humano que se diferencian del funcionamiento de un computador convencional.

- **Procesamiento en paralelo**: Intervienen miles de millones de neuronas para procesar una gran cantidad de información simultáneamente.
- **Memoria distribuida**: La información se distribuye por la sinapsis de la red actuando a modo de **caja negra**.
- **Adaptabilidad**: Aprenden de la experiencia y generalizan conceptos a partir de casos particulares.
- **No linealidad**: De modo que sus respuestas no se corresponden a una simple relación proporcional entre una acción y su efecto.

Cada neurona artificial o **perceptrón** [Sim20] recibe datos a través de sus enlaces de entrada, procesándolos para generar una salida. Estas conexiones entre neuronas se llaman **sinapsis**, formando un sistema interconectado en capas que procesa la información de entrada hasta obtener un resultado final. Aunque una neurona puede recibir múltiples señales de entrada simultáneamente, solo produce una señal de salida, cuyo valor, positivo o negativo, está determinado por la relevancia asignada a cada entrada, conocida como su **peso**, siendo crucial para determinar si la neurona se activa o no.

La **función de activación** en una red neuronal es un mecanismo fundamental que transforma el valor de activación de una neurona en su respectiva salida, actuando como un filtro o umbral para determinar si la neurona se activa o no. Entre las distintas funciones de activación encontramos: *Función Escalón*, *Función Sigmoidal*, *Función Rectificadora (ReLu)* o *Función Tangente Hiperbólica*.

La organización de una red neuronal implica la disposición de las neuronas en capas.

- La **capa de entrada** recibe señales y las transmite sin procesar a las capas siguientes.
- La **capa de salida** emite la respuesta final del modelo.
- Las **capas intermedias** u **ocultas** realizan el procesamiento y reconocimiento de la información.

Las RNA adoptan un enfoque de aprendizaje supervisado, donde ajustan gradualmente sus pesos durante el entrenamiento para minimizar la diferencia entre la predicción y el valor ideal. Este proceso permite que la red aprenda patrones aplicables a todos los datos, facilitando la generalización y la capacidad de hacer predicciones precisas sobre nuevos datos.

De esta forma, la representación de un perceptrón desde sus entradas hasta su correspondiente salida, se puede ejemplificar por medio de la Figura 2.2.

En el proceso de aprendizaje, la función de costes mide la diferencia entre las predicciones y el error, y es en base a este valor que, como bien se comentó anteriormente, nuestro modelo se ajusta para minimizar esta diferencia y encontrar una solución óptima.

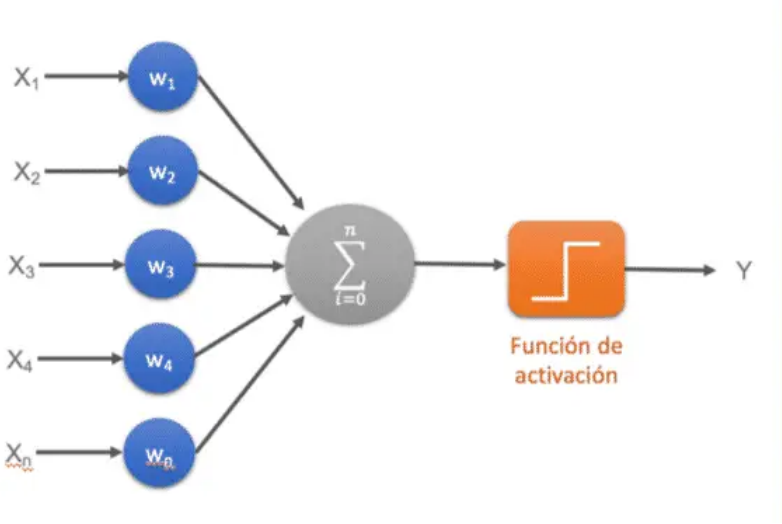

Figura 2.2: Capas de una RNA.

El **gradiente descendente** aborda este problema calculando la dirección para ajustar los pesos de manera eficiente, evitando la exploración exhaustiva de todas las combinaciones posibles. Sin embargo, puede converger a mínimos locales en lugar del global, por lo que se utiliza el **gradiente descendente estocástico**, que ajusta los pesos de manera más frecuente después de ver cada observación, evitando quedarse atrapado en mínimos locales y permitiendo una convergencia más rápida. Para equilibrar convergencia y eficiencia, se puede combinar con el gradiente descendente por lotes, dando lugar al **gradiente descendente en mini lotes** o *mini-batch gradient descent*.

Otro de los conceptos fundamentales en el entrenamiento de redes neuronales es la **propagación hacia atrás** (*Backpropagation*). Mientras que la propagación hacia adelante (*Feed Forward*) calcula la función de costes, la propagación hacia atrás corrige errores de la red simultáneamente, lo que ahorra tiempo y recursos. Este proceso ajusta los pesos considerando su contribución al error, y se repite para todas las observaciones del conjunto de datos, ya sea observación por observación o en lotes. Tras procesar todos los datos, se completa un episodio o **época** y se repite el proceso nuevamente para mejorar el ajuste de los pesos.

2.1.4. Redes Neuronales Convolucionales

Las Red Neuronal Convolucional (RNC) son un tipo especializado de redes neuronales diseñadas para procesar datos estructurados, especialmente imágenes [JJ19]. Este método está inspirado en el funcionamiento del sistema visual humano, por ello, son capaces de reconocer patrones espaciales en las imágenes, lo que las hace ideales para tareas de visión por computadora.

En el proceso de reconocimiento, las RNC buscan extraer características relevantes de las imágenes, como bordes, texturas y formas, para reconocer patrones y ajustar los pesos de la red según la tarea específica que se requiera, como clasificación o detección de objetos.

Este proceso se divide en cuatro fases fundamentales [Ude23].

- **Operación de convolución:** La primera etapa de la convolución se define como una operación matemática que combina dos funciones para describir la superposición entre ambas [Mat24]. De esta forma, un modelo de convolución trataría con la imagen que se quiere procesar junto con un **detector de rasgos** específico, multiplicando y sumando los valores correspondientes de ambos para obtener un nuevo conjunto de características conocido como **mapa de características**. En este sentido, los filtros, como el de mejora de bordes, el de difuminado y el de detección de bordes, permiten resaltar las diferentes características de la imagen.
- **Operación de agrupación máxima o *Max Pooling*:** Busca reducir la dimensionalidad de los datos de entrada, preservando al mismo tiempo las características más relevantes para el reconocimiento de patrones obtenidos en la etapa anterior. Para ello, divide la imagen en regiones pequeñas y selecciona el valor máximo de cada región resultando en una nueva imagen con una resolución reducida pero que conserva las características más importantes de la original.
- **Operación de aplanado o *Flattening*:** Este proceso consiste en convertir el mapa de características resultante de las operaciones anteriores en un vector unidimensional. Esto implica tomar las filas de la imagen y colocarlas una debajo de la otra para formar un gran vector que servirá como entrada para la RNC.
- ***Full Connection*:** En esta fase final, se incorpora una capa totalmente conectada que busca establecer relaciones entre las características extraídas de las capas anteriores y las clases de salida. Esta capa procesa los datos aplanados, transformándolos en vectores que representan las características más relevantes de la imagen. Durante el entrenamiento, los pesos de esta capa se ajustan para mejorar la precisión de las predicciones.

  El proceso completo de convolución se puede ejemplificar mediante la Figura 2.3.

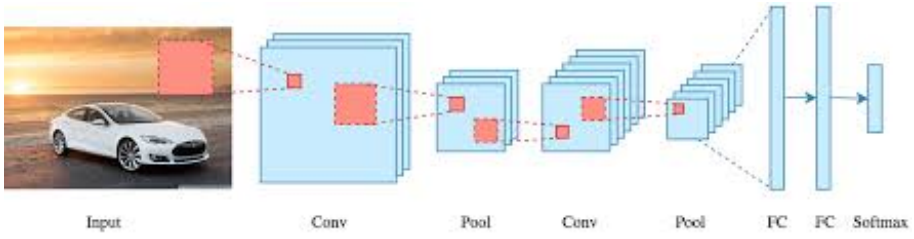

Figura 2.3: Etapas del proceso de convolución.

Las RNC han demostrado ser especialmente útiles en una variedad de aplicaciones, desde reconocimiento facial y diagnóstico médico hasta conducción autónoma y análisis de imágenes satelitales. Su capacidad para aprender automáticamente características relevantes de los datos las hace indispensables en el campo de la IA y el DL.

Además, cabe destacar que en el proceso de estudio con redes neuronales es común el concepto de ***transfer learning***, el cual consiste en reutilizar los elementos de un modelo de ML previamente entrenado en un nuevo modelo. Específicamente, se hace uso de esta técnica cuando se busca realizar un entrenamiento con un fin similar al de otra arquitectura ya existente.

Este enfoque reduce los recursos utilizados y la cantidad de datos etiquetados necesarios para entrenar a los nuevos modelos, por lo que se está convirtiendo en una práctica recurrente en el proceso de desarrollo de modelos de DL.

De esta forma se introduce el concepto de ***fine tuning*** que consiste en el entrenamiento del modelo pre entrenado, para el cual se ajustan algunas capas de la red para obtener las salidas deseadas. Es decir, se ajustan ligeramente ciertas características del modelo pre entrenado para que este encaje con los requerimientos del nuevo problema. Así, se evita definir la estructura de la red neuronal y realizar el entrenamiento desde cero.

2.2. Procesamiento de Audio con Aprendizaje Profundo

El procesamiento de **Audio con Aprendizaje Profundo** o *Audio Deep Learning* (ADL), por sus siglas en inglés, es una rama del DL que se centra en el procesamiento y análisis de datos de audio utilizando redes neuronales profundas y técnicas de IA relacionadas. Su objetivo principal es desarrollar modelos de aprendizaje automático capaces de comprender, procesar y extraer información útil a partir de señales de audio [KD21a].

2.2.1. Representación del Audio

El **sonido** es una señal que se produce cuando el aire experimenta cambios en la presión. De esta forma, podemos medir estos cambios en la presión y representarlos digitalmente. Un sonido puede ser suave o fuerte, definido como su **amplitud**, y también agudo o grave, denominado como **frecuencia**.

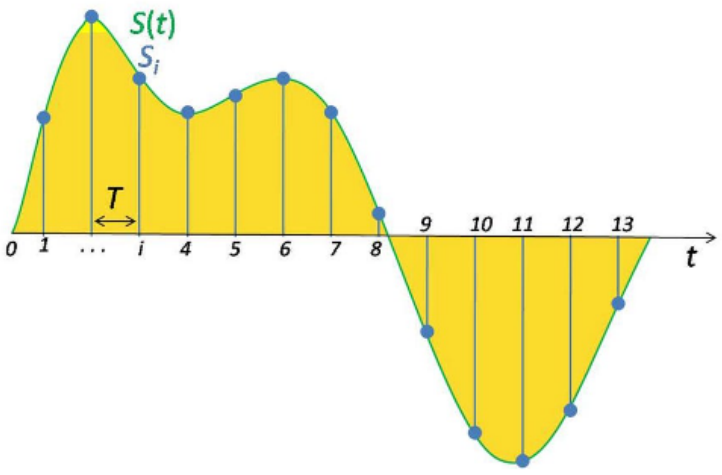

Figura 2.4: Muestra de audio.

Para que una computadora pueda trabajar con sonido, debe ser representada de forma numérica para introducirse en nuestro modelo como una entrada manejable y finita, en contraposición a la forma continua de la onda. Esto se hace tomando mediciones o **muestras** de la amplitud del sonido. La cantidad de muestras que tomamos por segundo se llama **tasa de muestreo** o *sample rate*. Una tasa de muestreo común es de alrededor de 44,100 muestras por segundo. Un ejemplo de esta transformación y representación de audi se muestra en la Figura 2.4.

2.2.2. Espectrograma

En los últimos años, el DL ha simplificado el proceso de tratamiento de audio. Por ello, en lugar de trabajar con datos de audio en bruto, se realiza una conversión en imágenes para posteriormente procesarlas en una RNC [KD21b].

Así, definimos el **espectro** como el conjunto de frecuencias que se combinan para producir una señal, de forma que representa todas las frecuencias presentes en la señal junto con la intensidad o amplitud de cada frecuencia.

Por otro lado, el **espectrograma** de una señal representa su espectro a lo largo del tiempo, es decir, una instantánea de los diferentes momentos en el tiempo de la señal unidos en un solo gráfico, como se muestra en la Figura 2.5.

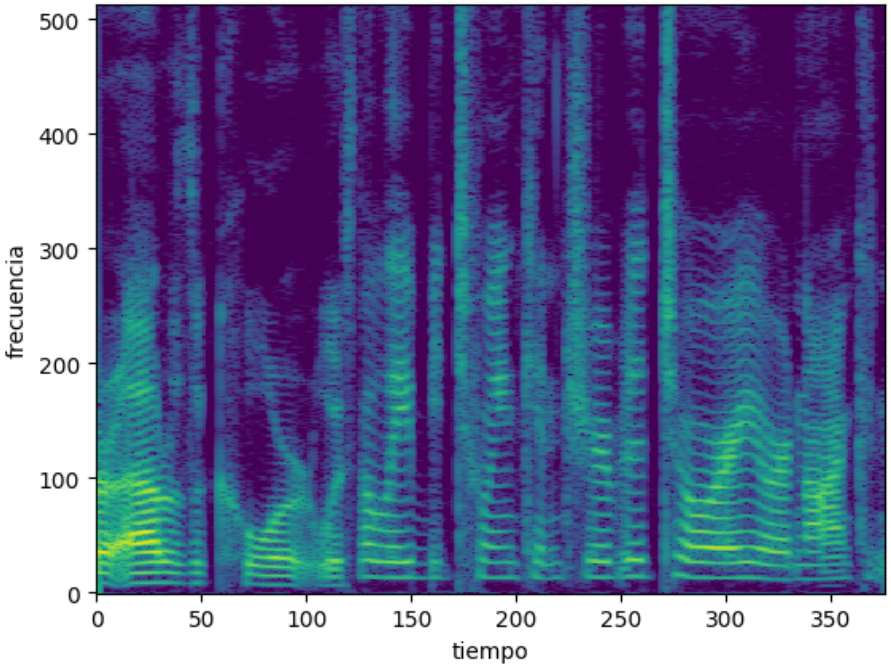

Figura 2.5: Espectrograma de una muestra de audio.

Una de las herramientas más importantes en el procesamiento de audio es la transformada de Fourier, ya que se encarga de descomponer cualquier señal en sus frecuencias constituyentes. Esta conversión se realiza mediante el algoritmo de la **Transformada Rápida de Fourier** o *Fast Fourier Transform* (FFT), por sus siglas en inglés.

Asimismo, para representar el espectro de estas señales a medida que varían en el tiempo, es decir, para generar un espectrograma utilizamos la **Transformada de Fourier de Tiempo Corto** o también denominado en inglés como *Short-time Fourier Transform* (STFT). Esta transformada se encarga de calcular varios espectros mediante FFT en segmentos solapados de la señal.

Para realizar esta conversión, necesitamos entender los siguientes parámetros.

- La variable *n_fft* se refiere al número de puntos de la transformada de Fourier a realizar en cada segmento de la señal durante el cálculo del espectrograma.

- Por otro lado, *hop_length* se refiere al número de muestras entre el inicio de un segmento de FFT y el inicio del siguiente segmento adyacente. En otras palabras, determina cuánto se solapan los segmentos en el tiempo.

2.2.3. Espectrograma de Mel

La percepción auditiva humana reconoce, por ejemplo, el intervalo entre 100 Hz y 200 Hz más distante en comparación con el intervalo entre 1000 Hz y 1100 Hz pese a que el incremento haya sido equivalente. Ya que en el primer caso observamos que se duplica el valor anterior, mientras que en el segundo caso no se aprecia una diferencia tan significativa con respecto al valor previo.

De esta forma, la **escala de Mel** es una herramienta fundamental en el procesamiento de señales de audio, diseñada para reflejar la capacidad humana para reconocer las frecuencias sonoras. Ya que, a diferencia de una escala lineal donde las frecuencias aumentan en intervalos uniformes, la escala de Mel opera de manera logarítmica siguiendo la naturaleza auditiva humana.

Para crear un **espectrograma de Mel**, se utiliza STFT de la misma manera que se comentó anteriormente, dividiendo el audio en segmentos cortos para obtener una secuencia de espectros de frecuencia. Además, cada espectro se pasa a través de un conjunto de filtros, llamados filtros Mel, para transformar las frecuencias a la escala Mel. De esta forma obtenemos una representación como la Figura 2.6.

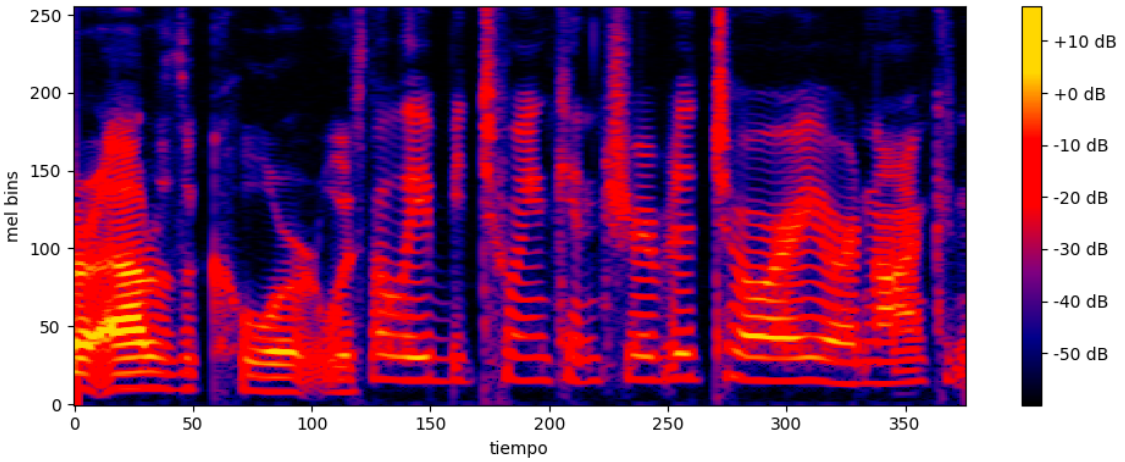

Figura 2.6: Espectrograma de mel.

Para realizar esta conversión debemos configurar el valor de *n_mels* que representa el número de bandas Mel a generar. Las **bandas Mel** definen un conjunto de rangos de frecuencia que dividen el espectro en componentes significativos, utilizando un conjunto de filtros cuya forma y espaciado se eligen para imitar la manera en que el oído humano responde a diferentes frecuencias.

Al igual que con un espectrograma regular, la intensidad de los componentes de frecuencia Mel se expresan en decibelios. Esto se conoce comúnmente como un **espectrograma log-Mel**, porque la conversión a decibelios implica una operación logarítmica.

2.2.4. Coecientes Cepstrales en las Frecuencias de Mel

Otra de las representaciones más populares y relevantes en el sector del procesamiento de audio consiste en los **Coeficientes Cepstrales de las Frecuencias de Mel** o MFCC, por sus siglas en inglés.

Esta representación alternativa a las mencionadas anteriormente, surge con el objetivo de representar de forma precisa el habla basándose en la percepción auditiva humana. Para ello, los MFCC muestran las características locales de la señal de voz obtenidas a partir de un proceso de filtrado conocido como *Mel-Cepstral*.

Tras este proceso, se recuperan los coeficientes como un conjunto de valores numéricos que resumen la información básica que constituye una señal de voz.

2.2.5. Coeficientes Cepstrales en las Frecuencias Lineales

De forma similar, LFCC y MFCC son en gran medida representaciones idénticas en cuanto al esquema de extracción de coeficientes. La única diferencia se encuentra en el banco de filtros aplicados, ya que los coeficientes del banco de filtros de los LFCC cubren por igual todas las gamas de frecuencias del habla y las consideran de igual importancia.

Por ello, los LFCC obtienen en la mayoría de ensayos mejoras significativas respecto a los coeficientes con Mel en cuanto a muestras con diversidad de hablantes por género, acento o edad.

Capítulo 3

Estado del Arte

Cada vez es más fácil grabar muestras de audio o vídeo con los aparatos modernos ya que, gracias al avance tecnológico, tenemos una amplia gama de posibilidades a nuestro alcance. Todo esto se traduce en que cada vez tenemos acceso a un mayor número de herramientas que pueden causar una distorsión de la realidad de forma que no podemos diferenciar lo que es real de lo que no. Esto aplica, por ejemplo, al hecho de modificar un vídeo o un audio con el objetivo de alterarlo o falsificarlo y pasar como un ejemplar original.

Desde las primeras décadas del siglo pasado, ha ido aumentando el interés en la detección de estas modificaciones, inicialmente con técnicas basadas en características de audio tradicionales, y más recientemente utilizando técnicas de aprendizaje automático debido a la popularización de estas.

Entre los trabajos que ya existen respecto a esta temática, la mayoría se centra en la detección de las dos formas más comunes de falsificación de audio. Por un lado, la primera conocida como **empalme** o *splicing*, y por otro, la segunda conocida como **copia-pega** o *copy-move*.

3.1. Manipulaciones de Audio

Las manipulaciones del audio es un proceso que implica modificar las características de una señal de audio, o bien para mejorarla o bien para alterarla con fines específicos. Esta práctica se utiliza en diversos ámbitos, como la producción musical, la edición de audio, la forense acústica, e incluso con fines maliciosos.

Estas manipulaciones tienen diversos objetivos, según el contexto en las que se apliquen. Los objetivos mas comunes son:

- Mejora de la calidad: Eliminando el ruido de fondo, reduciendo las distorsiones, etc.
- Creación de efectos sonoros: Generando ecos, distorsiones, reverberaciones, etc.
- Manipulación con fines maliciosos: Insertando mensajes subliminales, alterando pruebas de audio, creando contenido falso, etc.

Entre las manipulaciones mas comunes encontramos dos tipos:

- Basadas en el Contenedor: Se centran en modificar el formato o la estructura del archivo del audio sin alterar su contenido.
- Basadas en el Contenido: Modifican el contenido de la señal del audio.

Más concretamente, el trabajo a realizar consistirá en la detección de manipulaciones de la segunda categoría, con el objetivo de la detección eficaz de la presencia de alguna modificación en el contenido de un audio.

3.1.1. Métodos Basados en el Contenido

En este apartado se profundiza en la manipulación del audio basada en el contenido. Esta manipulación de audio se enfoca en la alteración de las características intrínsecas del audio en sí, como su forma de onda, el espectro de frecuencias, u otras. Esto puede resultar en que un audio sea modificado de tal manera que su sonido o su espectrograma resultante sea diferente al original, consiguiendo así, la ocultación de las características originales.

Las dos técnicas mas conocidas de modificación de audio basadas en el contenido, modificando la forma de onda, y en las que se basa nuestro trabajo son:

- Empalme o *Splicing*
- Copia-pega o *Copy-move*

3.1.2. Trabajos Previos de Detección de *Splicing*

Esta técnica, que se podría traducir al español como empalme, consiste en la unión de dos o más segmentos de audio de diferentes fuentes con el objetivo de crear uno nuevo diferente al original. Existen tres tipos principales de *splicing* son:

- *Splicing* de inserción: Consiste en insertar un segmento de audio dentro de otro archivo de audio, con el objetivo de crear nuevos sonidos, efectos especiales o corregir errores en la grabación original.
- *Splicing* de eliminación: Consiste en eliminar un segmento de audio de un archivo, con el objetivo de eliminar partes no deseadas de ese audio.
- *Splicing* de sustitución: Este tipo de *splicing* consiste en reemplazar un segmento de audio de un archivo por otro segmento de audio, logrando así un cambio en el orden de las partes del audio.

Con el objetivo de que sea más facil de entender, la siguiente Tabla 3.1 explica de una manera gráfica los tres tipos de *splicing*.

Tabla 3.1: Tipos de *splicing*.

Tipo de modificación	Transcripción
Original	**Me encanta jugar al fútbol.**
Inserción	Me encanta jugar al fútbol *con él*.
Sustitución	Me encanta jugar al *tenis*.
Eliminación	Me encanta jugar ~~al fútbol~~.

Como primera referencia al estudio de la detección de *splicing* en audio, se ha tomado la publicación [PZL12], del año 2012 que utiliza el conjunto de datos de *Texas Instruments / Massachusetts Institute of Technology* (TIMIT) [MIoTMTI93]. El método propuesto en el articulo se basa en la estimación del nivel del ruido local para detectar puntos de *splicing* en el audio. Para ello, primero generan el espectrograma de Mel usando la STFT, después para cada región dentro de ese espectrograma de Mel, calcula una medida estadística llamada *Kurtosis*, la cual indica la distribución de los valores dentro de una distribución de probabilidad, y posteriormente se comparan la *kurtosis* calculada para cada region y se compara con la *kurtosis* esperada para esa región. Si la diferencia entre ambas supera un valor determinado, esto indica potenciales puntos donde se ha realizado el *splicing*. Con sus experimentaciones consiguen mejorar los trabajos existentes en cuanto a detección de *splicing*.

Otro trabajo que se ha tenido en cuenta es el artículo publicado en el año 2017, [ZCWM17], que utiliza el conjunto de datos de TIMIT y otro creado por ellos, y propone como técnica de detección la extracción de las características del entorno en el que se ha grabado el audio. Para extraer esas características utilizan la STFT. Una vez que se han extraído las características, se comparan las firmas ambientales de cada segmento del audio para que, en el caso de la aparición de una diferencia notable entre dos segmentos, se marque ese punto como posible lugar de *splicing*. Los autores del artículo evaluaron su método utilizando dos conjuntos de datos: el conjunto de datos TIMIT y otro conjunto de datos creado por ellos mismos. Los resultados obtenidos mostraron que el método propuesto era capaz de detectar *splicing* con una precisión y una tasa de detección superiores a las de los métodos existentes.

Posteriormente se consultó el trabajo [JPR19], publicado en el año 2019, que propone una RNC como método de detección de la falsificación en el audio. El conjunto de datos utilizado es el *Free Spoken Digit Database*, y es manipulado insertando segmentos de entre 1 y 3 segundos en diferentes partes de los audios originales. Experimentan aplicando diversas varias modificaciones al audio original como por ejemplo superposición, compresión, adición de ruido o eliminación de silencios. Este es el primer trabajo que utiliza una RNC para detectar si se ha realizado *splicing*. Utilizan una RNC para poder usar las características del espectrograma como entrada para el aprendizaje de la red, y tras la experimentación llegan a la conclusión de que la mejora que ello produce es que cuando el audio es comprimido, no se ve afectada la eficacia de la detección del *splicing*.

Por otro lado, se ha tenido en cuenta el trabajo [ZW22], publicado en el año 2022, en el que usan el conjunto de datos de *LibriSpeech*, que contiene alrededor de 1000 horas de conversaciones de audio libros en inglés, y manipulan ese conjunto de datos aplicándole

inserción, substitución y eliminación de partes de unos audios en otros. Como modelo de detección de falsificaciones proponen una *ResNet* [HZRS16] de 18 capas. Esta red en concreto se caracteriza por su capacidad para evitar el problema del desvanecimiento del gradiente, lo que le permite aprender a partir de conjuntos de datos más grandes y complejos. Esta *ResNet* se compone de varias capas convolucionales residuales, que extraen características del espectrograma y las combinan y procesan para obtener una predicción de si el segmento de audio contiene *splicing* o no. La experimentación, como se ha comentado previamente, consiste en la alteración del conjunto de datos de *LibriSpeech* utilizando los tres tipos de *splicing* explicados al principio de la subsección 3.1.2, ya que no existen muchos conjuntos de datos para *splicing*. La Tabla 3.2 ilustra los resultados obtenidos.

Tabla 3.2: Métricas de validación

Chunk size	Precisión (%)	Sensibilidad (%)	*F1-score* (%)
04	40,9	31,0	35,2
08	74,6	38,2	50,6
16	81,3	46,0	58,8
32	84,6	**64,2**	73,0
64	**0,904**	61,7	**73,4**
32 MFCC	73,4	59,8	65,9
32 LFCC	82,3	56,6	67,1

3.1.3. Trabajos Previos de Detección de *Copy-Move*

La técnica de *copy-move*, traducido como copia-pega en español, básicamente funciona de manera que una parte de un archivo de audio se copia y se pega en otra parte del mismo archivo con la intención de manipularlo y cambiar su estado original. Un ejemplo de copia-pega en audio sería el copiar una palabra de un discurso, y pegarla en otro lado del discurso consiguiendo así repetir la palabra que se ha dicho en diferentes puntos de ese discurso. La siguiente Tabla 3.3 ilustra como funciona el copia-pega o *copy-move* en audio.

Tabla 3.3: *Copy-move* en audio

Tipo de modificación	Resultado
Original	**No siempre hay gente dispuesta a ayudar a los demás.**
Copy-Move	No siempre hay *gente* dispuesta a ayudar a *gente*.

Como primer trabajo para observar las técnicas de detección de copia-pega en audio, se leyó el trabajo [IABA17], publicado en el año 2017. Dicho trabajo utiliza como conjunto de datos la base de datos [AMA+22], que contiene audios de hablantes árabes, africanos e

indios, para después aplicarle cambios para conseguir un conjunto de datos modificado para detectar los audios con copia-pega. Como parte importante en la detección de la falsificación introducen el término *Voice Activity Detection* (VAD) para detectar los momentos en los que se está hablando en el audio, y tras detectar esos puntos en los que se produce habla. Posteriormente extraen las características de esos fragmentos usando MFCC, y para cada fragmento de voz, se compara con sus vecinos para generar un código basado en las diferencias. Finalmente se crea un histograma según esos códigos, y se comparan los fragmentos sospechosos con los auténticos, y donde haya una diferencia significativa, indica que puede existir una posible falsificación. Los autores destacaron que el uso de códigos e histogramas permitió a su método ser robusto ante las variaciones en las características acústicas del habla.

Posteriormente se revisó el trabajo [UTU23], publicado en Octubre de 2022, en el que utilizan dos conjuntos de datos para trabajar, el de TIMIT, y el de *Arabic Speech Corpus* [Hal16], proponen un método que emplea imágenes de espectrogramas Mel y puntos clave *Scale-Invariant Feature Transform* (SIFT) para detectar falsificaciones. Lo que hace ese algoritmo SIFT es extraer de los canales *Red-Green-Blue* (RGB) las características distintivas que ayudan a identificar regiones similares en las imágenes. Después se descartan coincidencias aleatorias entre puntos clave mediante *Random Sample Consensus* (RANSAC) y vectores de desplazamiento, se analizan los bloques alrededor de los puntos clave coincidentes para detectar posibles alteraciones, y para identificar regiones duplicadas se buscan áreas con características similares en los espectrogramas Mel mediante etiquetado de componentes conectados. Como experimentación, modifican los conjuntos de datos aplicándoles diversas técnicas de ADA como por ejemplo añadiéndole ruido o realizando una compresión del audio, y se comparan con métodos ya existentes, logrando mejores resultados que estos. La siguiente Tabla 3.4 ilustra la comparación entre los resultados del trabajo y los de los demás estudios sobre el conjunto de datos utilizado en el trabajo [Hal16]:

Tabla 3.4: Tipos de ADA aplicados y sus resultados.

Modificación	Tasa de Acierto en Detección de Copy-move (%)
Añadiendo Ruido de 20 dB	80
Añadiendo Ruido de 30 dB	86
Comprimiendo a 64 kbps	86

El trabajo *Robust audio copy-move forgery detection on short forged slices using sliding window* publicado en Mayo del año 2023 nos sirvió de gran ayuda, ya que los autores generan dos conjuntos de datos modificados, uno basado en *LibriSpeech* para audios en inglés, y otro en chino. Como método de detección de copia-pega combinan el uso de la ventana deslizante *Sliding Window* (SW) y los *Constant Q Cepstral Coefficients* (CQCC) llamándolo *SW-CQCC*. Primeramente se hace uso del VAD para detectar fragmentos de habla humana, y usando la SW van dividiendo las partes del audio para extraer las características CQCC para cada ventana, y se compara cada par de segmentos midiendo la similitud entre ambos usando el Coeficiente de Correlación de *Pearson* (CCP), así se puede localizar dos ventanas similares, lo que podría significar una posible falsificación de copia-pega. Como experimentación, ellos modifican los dos conjuntos de datos anteriormente mencionados consiguiendo 500 audios originales y 500 modificados,

de 5 segundos cada uno, y esto en ambos conjuntos de datos, y al aplicar su método de detección logran un buen resultado comparado con los ya existentes y sacando como conclusión que los métodos existentes detectan de forma decente la existencia de copia-pega en audio, pero no son tan eficaces cuando esa modificación existe en un fragmento muy pequeño, que es en lo que se enfoca este trabajo.

3.1.4. Limitaciones Encontradas

En este apartado se discuten las limitaciones comunes que se encuentran en los trabajos de detección de falsificación por splicing y copy-move en audio. Estas limitaciones son importantes para considerar al desarrollar y evaluar nuevos métodos de detección de falsificación.

Estos trabajos existentes, se enfocan en detectar o bien si solamente hay *copy-move* o bien si solamente hay *splicing*, pero no detectan si existen ambas modificaciones en un mismo conjunto de datos. Estos trabajos consiguen unas métricas realmente buenas, pero solamente para cuando se ha hecho una de las dos modificaciones exclusivamente, por lo que el trabajo propuesto se centra en la detección de la existencia de ambas modificaciones en fragmentos de audio, para así mejorar la eficacia de la detección de falsificación en audios.

Tras revisar los trabajos existentes, se pudo apreciar que no existían numerosos conjuntos de datos de dominio público en el que existiesen tanto audios originales, como esos mismos con la aplicación de alguna de estas dos técnicas de falsificación, por lo que para la realización de nuestro trabajo, fue necesaria la creación de nuestro propio conjunto de datos, proceso el cual se explicará a continuación en el Capítulo 4.1.

Tras la revisión de los trabajos existentes, se llegó a la conclusión de que pocos de ellos utilizaban técnicas de ADA para mejorar los modelos de detección de falsificación en audio, y al nosotros utilizar un conjunto de datos propio, se ha decidido utilizar algunas de estas técnicas con el objetivo de conseguir un mejor entrenamiento, evitando así que nuestro modelo memorice a la hora de entrenar. La aplicación de estas técnicas se explicará en el Apartado 4.3.

Dadas las limitaciones encontradas en el Capítulo 3, nuestro objetivo es conseguir un conjunto de datos propio para trabajar con él y poder entrenarlo para conseguir detectar si a un audio se le ha aplicado alguna de las técnicas de modificación del contenido explicadas previamente, intentando alcanzar los resultados obtenidos en los trabajos que se han revisado. En el siguiente capítulo, se procederá a explicar el trabajo realizado.

Capítulo 4

Detección de Manipulaciones de Audio

A través del Capítulo 4, se expone el proceso completo que se ha realizado desde la obtención de nuestro conjunto de datos, posterior estudio y manipulación de algunas muestras de ejemplo para comprender las modificaciones y características relevantes a estudiar y, por último, la arquitectura y modificaciones propuestas para el modelo a entrenar. En resumen, en este Capítulo se expone todo el grueso del trabajo realizado con el objetivo de entrenar un modelo para reconocer las manipulaciones de *Copy-move* y *Splicing* en muestras de audio.

4.1. Conjunto de Datos

Para realizar el entrenamiento de nuestro modelo requeríamos de un *dataset* que incluyese tanto audios no modificados como muestras a las que se les hayan aplicado *copy move* y *splicing*. Debido a la ausencia de conjuntos de datos con sendas modificaciones decidimos, finalmente, desarrollar un código propio para realizar esta edición sobre un conjunto con audios originales.

Concretamente utilizamos el conjunto TIMIT [MIoTMTI93], un reconocido corpus de audios desarrollado por el *Massachusetts Institute of Technology* (MIT) y frecuentemente utilizado para **Reconocimiento Automático del Habla** o *Automatic Speech Recognition* (ASR), por sus siglas en inglés. Este dataset cuenta con más de 1600 audios en inglés de entre 3 y 7 segundos de duración, y con una heterogeneidad de hablantes de diferentes etnias, géneros y edades.

La variedad y diversidad de los audios nos aseguraba tener un conjunto que permitiese una mejor generalización de nuestro modelo, evitándonos de esta forma un posible sobre aprendizaje en base a la repetición de audios con un mismo tipo de hablante. Cabe destacar, aún así, que este corpus únicamente incluye audios en inglés ya que, pese a que se ha observado como en trabajos relacionados se incluyen muestras en otros idiomas, esta práctica no es preponderante en la mayoría de artículos ni se ha observado una mejora significativa en las comparaciones del rendimiento con trabajos donde solo se utilizan audios en inglés.

Respecto al código para generar audios modificados, nos basamos en la explicación del

Artículo [ZZY22], el cual obtuvo unos resultados consistentes utilizando este método en un conjunto de audios en inglés y chino, superando el 97 % de *accuracy* en las pruebas realizadas.

De esta forma realizamos un método de modificación siguiendo las instrucciones de este trabajo que, por abreviar, denominamos *procedimiento por concatenación* y resumimos en el Algoritmo 1.

Algoritmo 1: Generación de muestras de audio por concatenación

1: Segmentar *clips* de audio originales en *clips* de 1 y 2 segundos
2: Seleccionar aleatoriamente un clip de 1 segundo y un clip de 2 segundos
3: Concatenar los *clips* de las siguientes formas:
Audio 1: Colocar primero el *clip* de 1 segundo y luego el de 2 segundos.
Audio 2: Colocar primero el *clip* de 2 segundos y luego el de 1 segundo.
Audio 3: Dividir el *clip* de 2 segundos en dos segmentos de 1 segundo y colocar el clip de 1 segundo entre ellos.
4: Seleccionar aleatoriamente dos *clips* de 1 segundo
5: Realizar las mismas concatenaciones explicadas en el paso anterior, dividiendo en el último caso uno de los audios en dos segmentos de 0.5 segundos e incluyendo el otro clip en medio

Mediante este proceso, se generan 20.160 *clips* de audio manipulados de 3s y 2s, de forma que se obtienen 10.080 audios para cada tipo de modificación. Cabe destacar, entonces, que se observa un desequilibrio en la cantidad resultante de audios modificados en comparación con los originales, representado en la Figura 4.1.

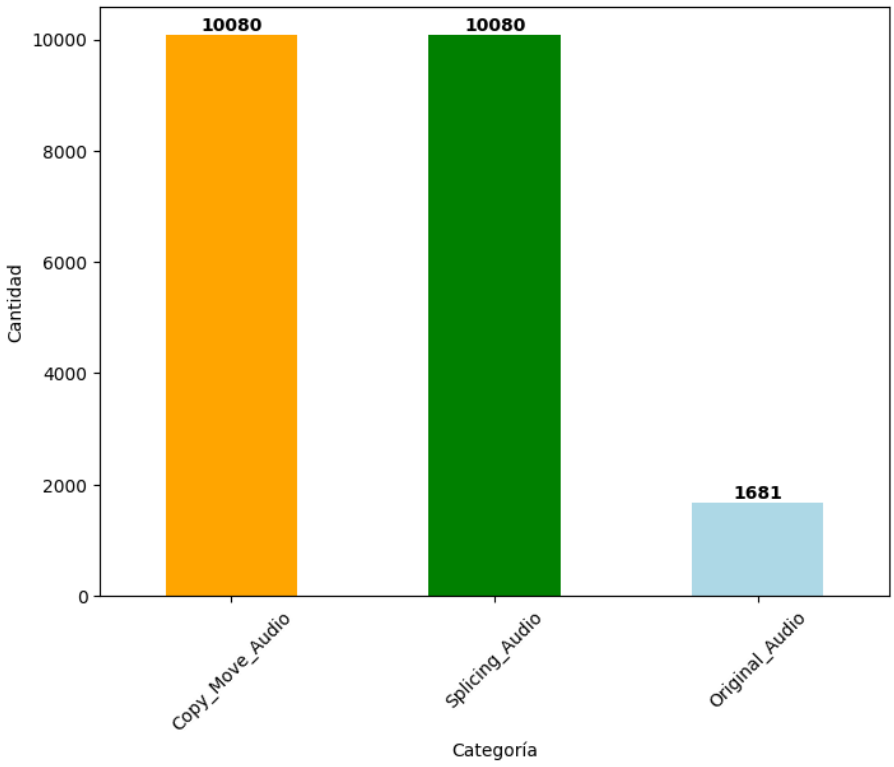

Figura 4.1: Comparativa de los audios por categoría.

Este desequilibrio facilita un sesgo hacia la clasificación de audios modificados, ya que los pesos pueden no estar ajustados correctamente, lo que puede resultar en un excesivo número de falsos positivos.

Con el objetivo de aliviar este problema, aplicamos una de las solución más populares que involucra ajustar el modelo, es decir, sin modificar los datos realizamos una ponderación de los pesos de la red en base al desequilibrio. Para ello se define cada peso de manera inversamente proporcional a la frecuencia relativa de la clase, y luego se normaliza para que el sumatorio de pesos sea igual al número de clases.

Otra de las principales características que se pueden observar en este conjunto es la preponderancia de audios de duraciones de 2 y 3 segundos debido al proceso comentado en el Algoritmo 1, realizándose concatenaciones con fragmentos de longitudes exactas. Existen otras duraciones pero todas se encuentran clasificadas dentro del conjunto de audios originales, lo que podría traducirse en otro problema de clasificación en el que enfocar un posible análisis tras la evaluación de los resultados. Esta distribución de duraciones de las muestras de nuestro conjunto puede observarse en la Figura 4.2.

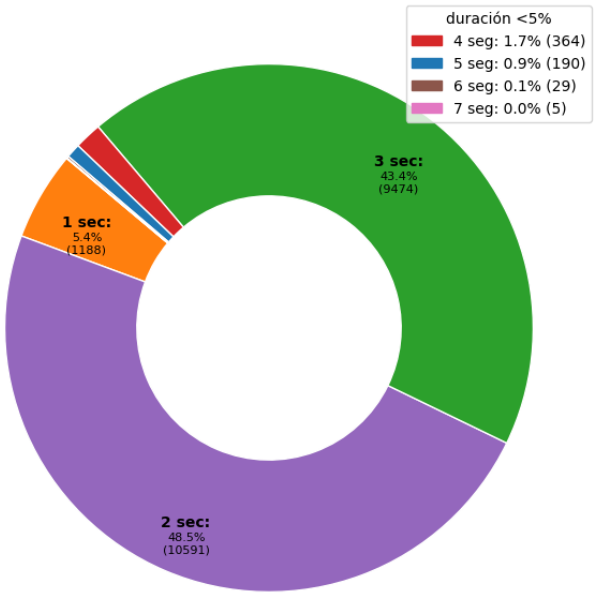

Figura 4.2: Comparativa de la duración de los audios.

4.2. Preprocesamiento de Audio

La siguiente etapa consistió en estudiar los *clips* de audio y sus correspondientes espectrogramas. Concretamente buscábamos ajustar los hiper parámetros que afectaban directamente a la conversión en imagen con el fin de mejorar el rendimiento del modelo, enfatizando las secciones más relevantes del espectrograma y optimizando lo máximo posible el tamaño de las imágenes generadas sin perder información.

Primero de todo, en la Figura 4.3 observamos 3 audios, siendo el primero una muestra original y los otros dos sus respectivas modificaciones *splicing* y *copy-move*. Se aprecian diferencias con respecto a su equivalente audio original por lo que, con la correcta extracción de características, nuestro modelo podrá identificar las alteraciones aplicadas.

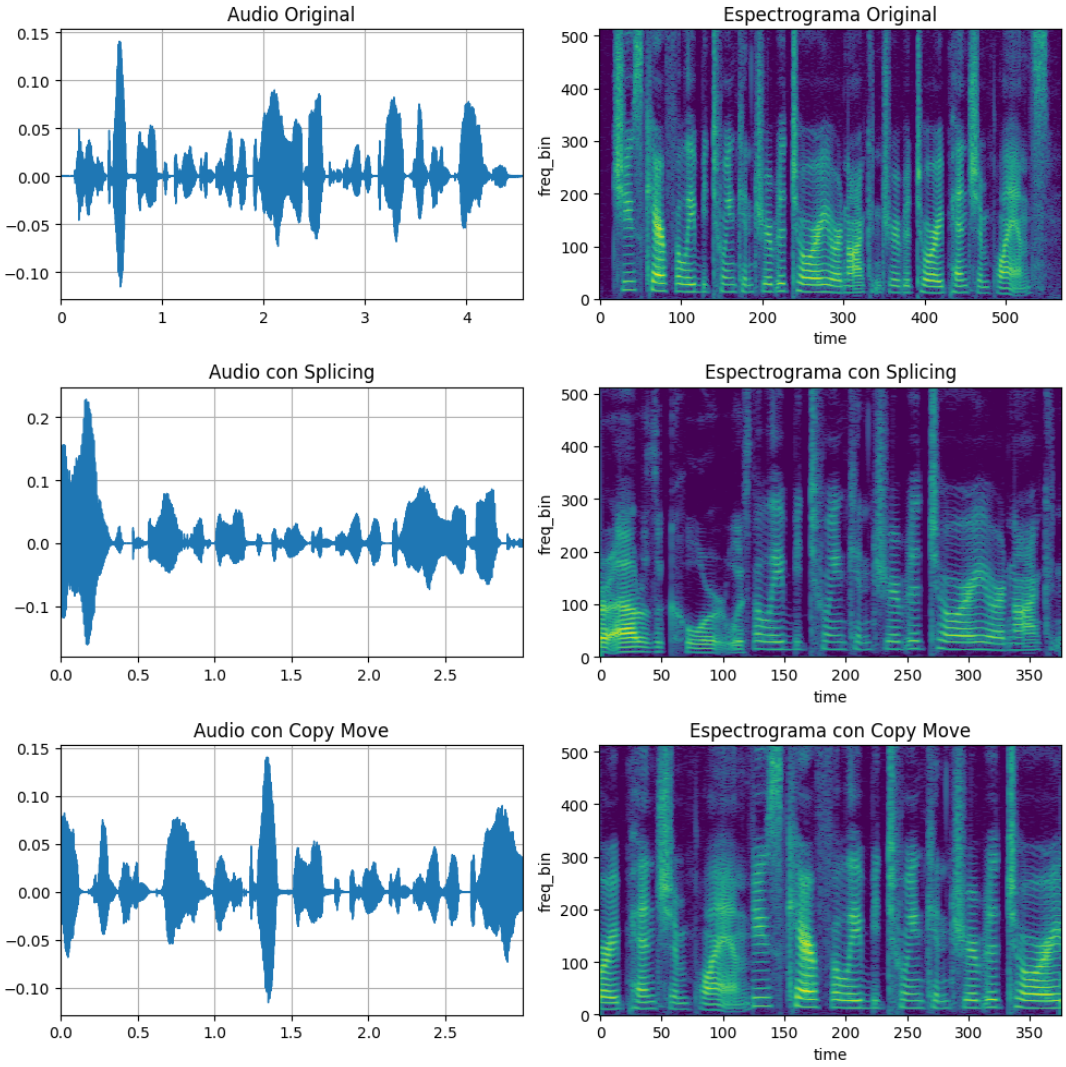

Figura 4.3: Comparativa entre audio original, audio con *splicing* y audio con *copy-move*.

Cabe recordar que debido al método utilizado y comentado previamente en el Algoritmo 1, se observa como el audio original tiene una duración de 5 segundos mientras que los modificados 2 segundos, por lo que se trata de una sección del audio original.

4.2.1. Ajuste de Tamaño de Espectrogramas Mediante Redimensionamiento

Para estandarizar la duración de los *clips* de audio, se llevaron a cabo pruebas de ajuste de tamaño de los espectrogramas mediante la técnica de *resize*. Esta metodología consiste en modificar las dimensiones de los espectrogramas para que se ajusten a una duración común, lo que permite el correcto procesamiento y entrenamiento del modelo.

4.2.2. Ajuste de Parámetros *Hop_length* y *N_fft*

Los parámetros de *hop_length* y *n_fft* son cruciales en el proceso de generación de espectrogramas mediante la FFT. Para ello, realizamos ajustes en estos parámetros y observamos comparativamente los resultados para optimizar y mejorar la calidad de la representación resultante. Cabe destacar que, debido a que ambos parámetros son complementarios, realizamos numerosos experimentos modificando ambos valores simultáneamente.

En la Figura 4.4, mostramos concretamente el estudio de la alteración que se produce con los cambios del valor de *n_fft*.

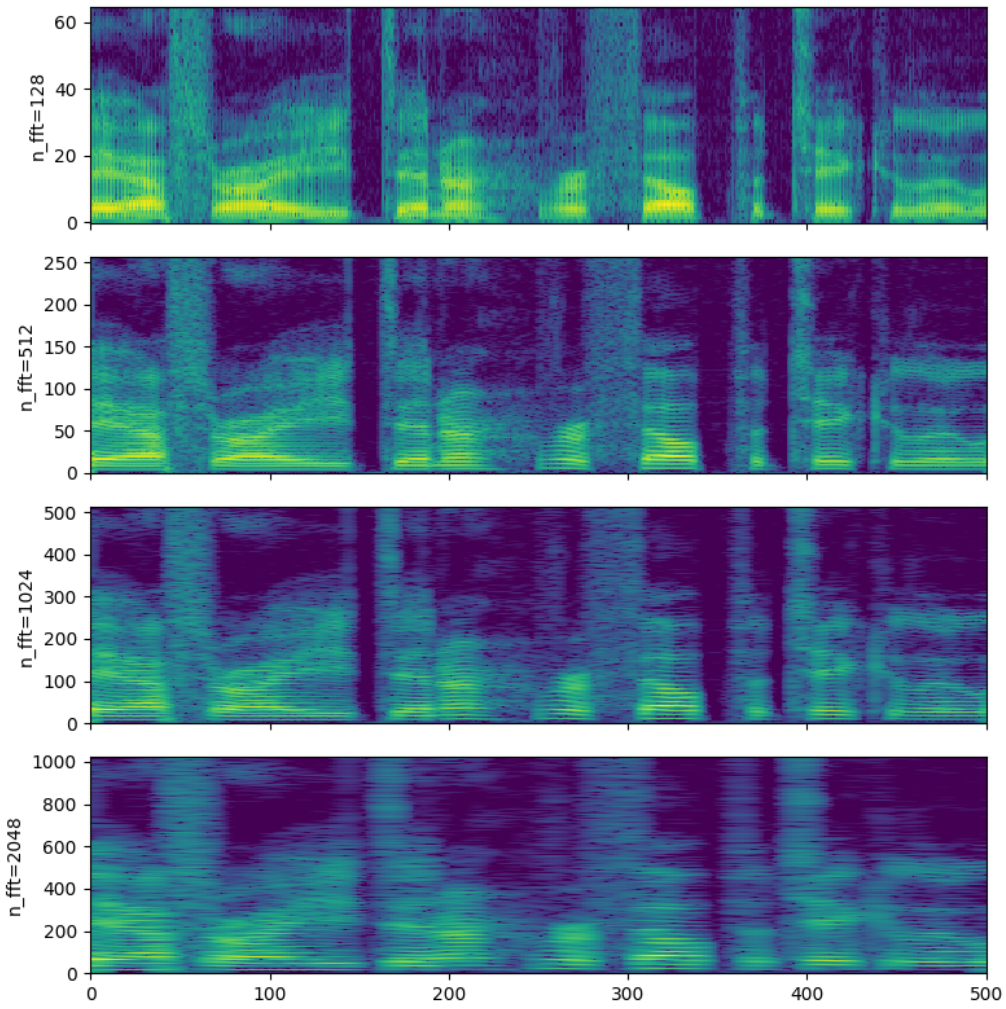

Figura 4.4: Comparativa de espectrogramas con diferentes valores de *n_fft*.

4.2.3. Observaciones de los Niveles de Tono de los Audios

Por último, realizamos observaciones de los niveles de *Pitch* de los audios para evaluar su relevancia como característica. Esto implicaba analizar la variación de tono en los *clips* de audio y determinar si esta característica aporta información útil para el proceso de clasificación.

4.2.4. Pruebas de *Downsampling* de Audio

Además, llevamos a cabo pruebas de *downsampling* para reducir la frecuencia de los audios a un estándar óptimo de 16.000 Hz. Con esta reducción buscamos evaluar posibles pérdidas de información relevante para el entrenamiento del modelo y, así, determinar si la reducción de la frecuencia afecta significativamente la calidad de la representación de los datos de audio.

Esta reducción de frecuencia se puede apreciar visualmente en la Figura 4.5.

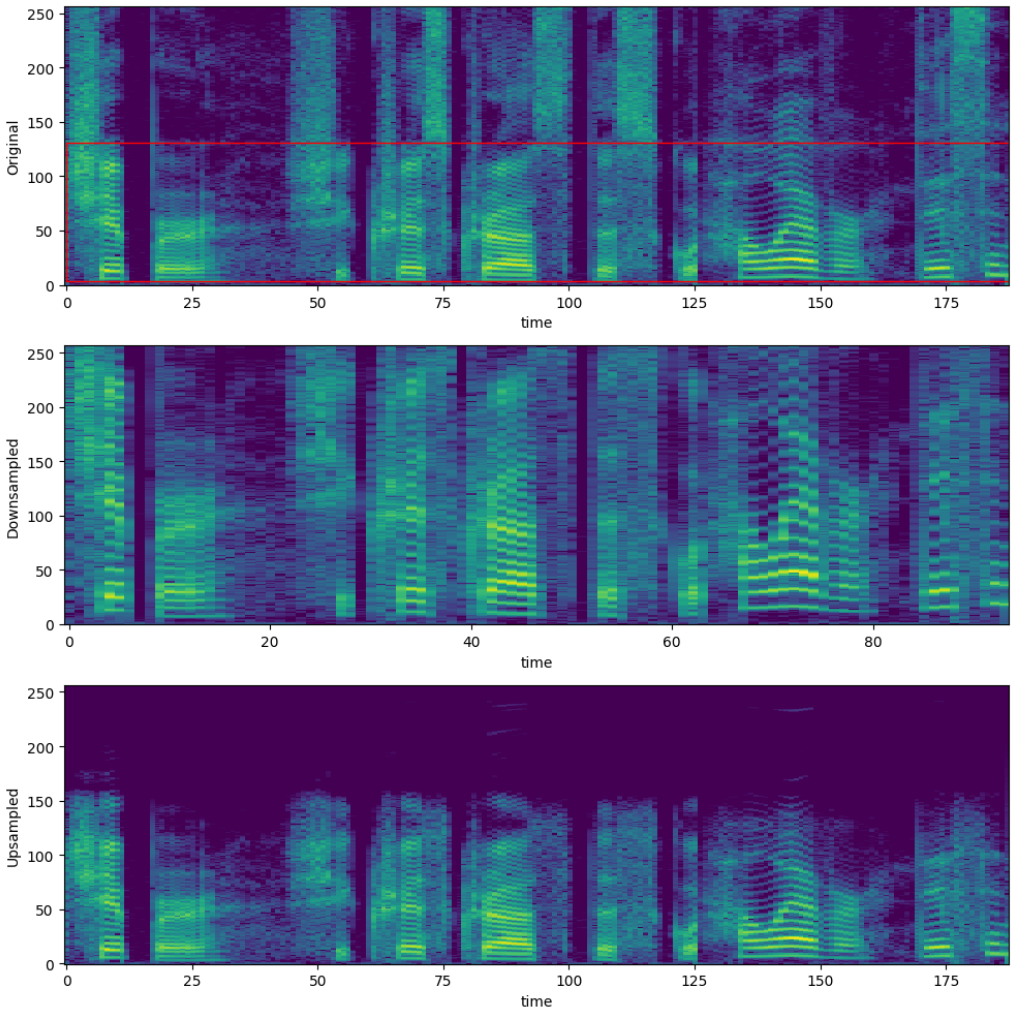

Figura 4.5: Comparativa del espectrograma aplicándole *downsampling*.

4.2.5. Conversiones a Diferentes Formatos de Representación de Audio

En esta fase, realizamos conversiones a los diferentes formatos de representación de audio propuestos, es decir, conversiones a *Espectrograma Log Mel*, MFCC y LFCC.

Así, podemos determinar cuál de estos formatos es más adecuado para la arquitectura del modelo y proporciona la mejor representación con la menor perdida de información de los datos originales. Para ello, ajustamos los parámetros que afectan directamente a la conversión de cada uno de los formatos con el fin de obtener el valor más eficiente.

Por un lado, realizamos varias pruebas modificando el *n_mfcc* en el proceso de generación de MFCC, dándonos como resultado imágenes como la de la Figura 4.6.

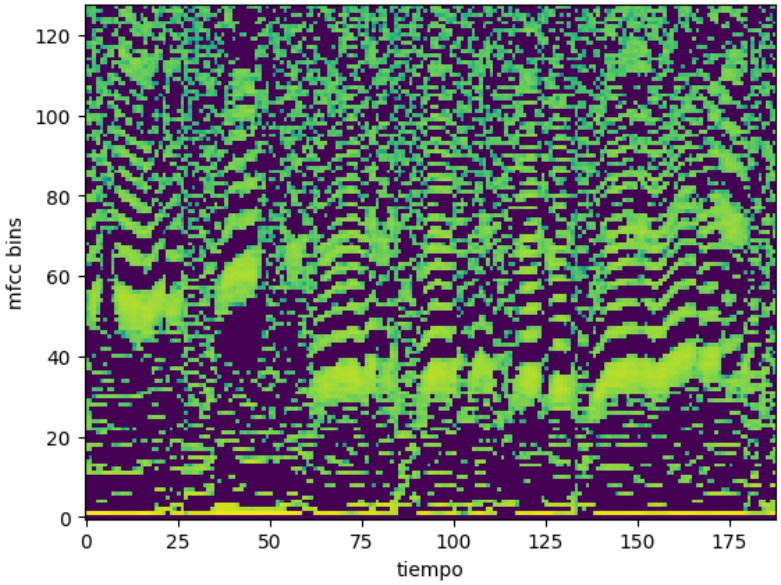

Figura 4.6: Representación de un espectrograma en formato MFCC.

Por otro lado, en la Figura 4.7 se observa una muestra en formato LFCC. Para este tipo de representación también se realizó un estudio para distintos valores de *n_lfcc*.

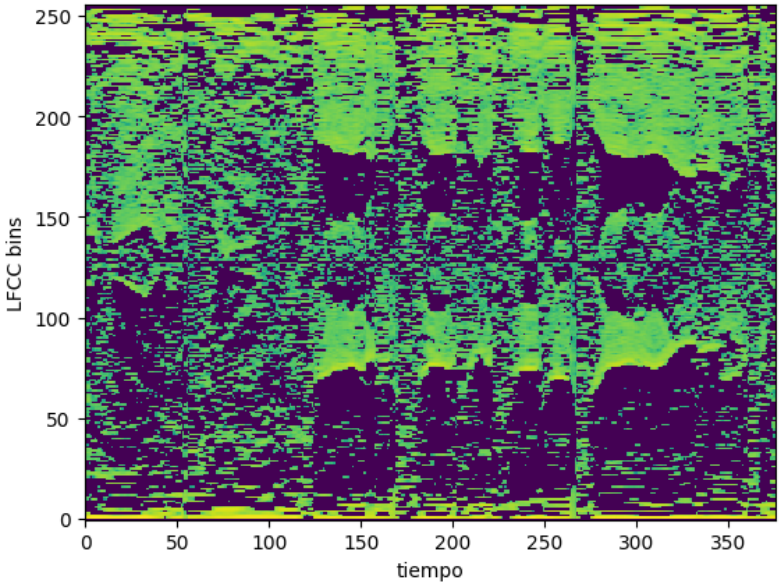

Figura 4.7: Representación de un espectrograma en formato LFCC.

4.3. Aumento de Datos

El aumento de datos, en inglés *data augmentation*, es un proceso de gran relevancia para la etapa de entrenamiento de un modelo, ya que nos permite suministrar datos nuevos y diferentes en cada época del entrenamiento a partir de los datos originales. Así, realizamos ciertas alteraciones a algunos audios con el objetivo de proveer al modelo con el mayor número de ejemplos posibles y evitar una memorización de las muestras.

Este aumento consiste en la aplicación de transformaciones o modificaciones a los datos originales existentes. Debido a que nuestros datos consisten en muestras de audio, vamos a llamarle **Aumento de Datos de Audio** o ADA, por sus siglas en inglés. Las principales técnicas de ADA que han sido consideradas para aplicar son las que modifican el eje Y, el cual corresponde a la frecuencia en los espectrogramas, ya que no nos interesaba modificar la duración de los segmentos, sino el aspecto. Concretamente, las técnicas utilizadas han sido:

- **Ruido Gaussiano**: Añade ruido gaussiano al fragmento de audio, lo que puede simular condiciones de ruido de fondo a la grabación.
- **Ganancia**: Multiplica el audio por un factor de ganancia, lo que puede aumentar o disminuir su amplitud, simulando variaciones en el volumen.
- **Desplazamiento de Tono**: Cambia el tono del audio, desplazando sus frecuencias hacia arriba o hacia abajo, lo que puede simular variaciones en la altura tonal.
- **Invertir**: Invierte el audio en su eje X, dejándolo al revés, lo que puede ser útil para crear variaciones en la dirección temporal del sonido.

Todas estas modificaciones pertenecen a la librería audio_augmentations [Spi21], una librería de *Python* comúnmente utilizada para este proceso en el entrenamiento de modelos de DL.

Asimismo, después de realizar el estudio, se ha observado una mejora en el realismo de los audios y variedad de las muestras, ya que estas características nos permiten simular audios grabados en diferentes situaciones cotidianas incluyéndoles, por ejemplo, un ruido de fondo que podría darse al grabar en entornos públicos o una saturación que podría producirse al grabar muy próximo al micrófono.

Podemos observar en la Figura 4.8 el impacto de todas estas alteraciones aplicadas sobre los audios realizando una comparativa con sus equivalentes originales en formato de espectrograma.

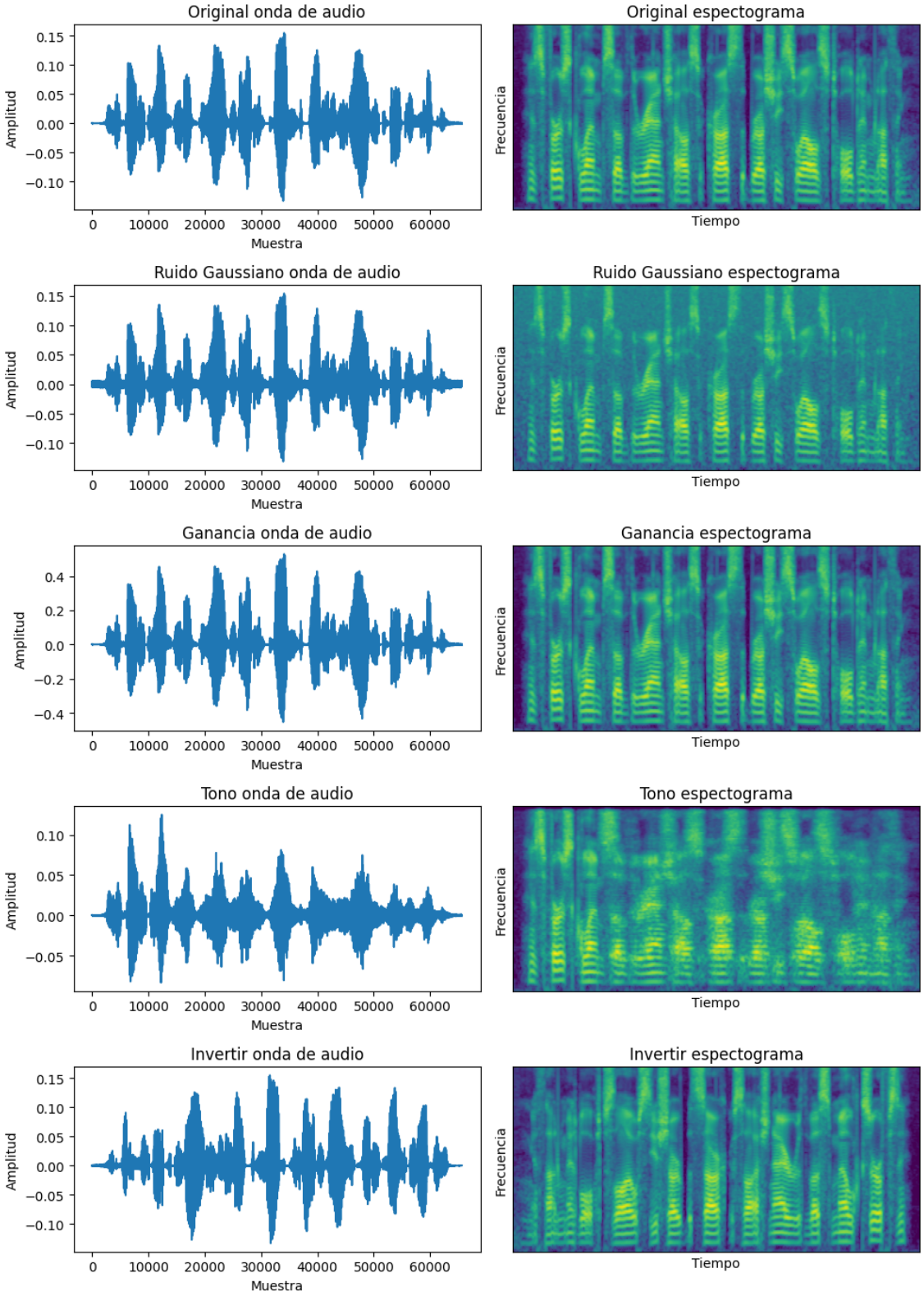

Figura 4.8: Comparativa de técnicas de ADA.

También, cabe destacar que se estudiaron otras técnicas de ADA como el caso de las máscaras de frecuencia y de tiempo. Para ello elegimos un tamaño aleatorio, aunque no muy elevado, y añadimos una máscara de ese tamaño tanto en el eje X como en el eje Y, de forma que se eliminase la información de la zona donde se aplicase la máscara.

Finalmente decidimos no utilizar esta técnica ya que eliminaba información relevante y podría ocultar la parte del audio modificada. En la Figura 4.9 se muestra el resultado del proceso de adición de ambas máscaras visualizado en un espectrograma de mel.

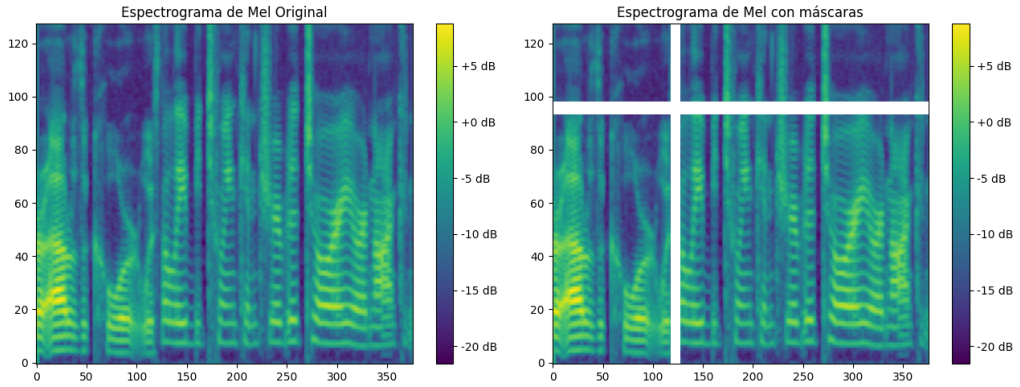

Figura 4.9: Comparativa de espectrogramas de mel con máscaras de frecuencia y tiempo.

4.4. Arquitectura del Modelo

Como bien se expuso al explicar los términos de *transfer learning* y *fine tunning* en el Capítulo 2, para este trabajo hemos utilizado un modelo pre entrenado con el objetivo de ajustar sus pesos sin partir desde cero. Para ello requeríamos de una arquitectura con la capacidad de reconocer imágenes para ajustarlo para clasificar espectrogramas.

Finalmente, nuestra elección fue **ConvNextv2_femto.fcmae_ft_in1k**, un modelo de clasificación de imágenes pre entrenado con un autoencoder convolucional o *Fully Convolutional Masked Autoencoder* (FCMA) al que se le realizó ajuste fino con **ImageNet-1k** [DDS+09].

ImageNet-1k es una base de datos de imágenes organizada de acuerdo a la jerárquica *WordNet* [Uni], es decir en base a un conjunto de datos léxico de vocabulario en inglés. En este conjunto, cada nodo esta representado por miles de imágenes etiquetadas según el conjunto léxico.

Este modelo cuenta con 5,2 Millones de parámetros y requiere de imágenes estandarizadas a un tamaño de 224 x 224 píxeles.

Los *ConvNets* modernos, representados por *ConvNeXt*, han demostrado un gran rendimiento en diversos escenarios. Aunque estos modelos se diseñaron originalmente para el aprendizaje supervisado con etiquetas de *ImageNet*, también pueden beneficiarse de técnicas de aprendizaje autosupervisado como los *Masked Autoencoders* (MAE).

Sin embargo, debido a que la combinación de estos dos enfoques da lugar a un rendimiento significativamente peor, es entonces cuando se desarrollaron los modelos denominados FCMA.

Este conjunto de técnicas de aprendizaje auto supervisado y de mejora de la

arquitectura da como resultado una nueva generación de modelos conocidos como *ConvNeXt V2*, que mejoran significativamente el rendimiento de los *ConvNets* puros en varias pruebas de reconocimiento, como la clasificación de *ImageNet*.

4.5. Propuesta

Tras haber presentado el modelo pre entrenado con el cual realizamos los entrenamientos, cabe especificar el proceso que se debe realizar para ajustar la arquitectura con respecto al nuevo objetivo.

Primero de todo, definimos los hiper parámetros previamente estudiados y que han obtenido unos mejores resultados. En concreto elegimos un *sample_rate* de 16.000, un *n_fft* de 1024, *hop_length* de 512, *n_mels* de 224 y una estandarización de las imágenes a 224x224.

Seguidamente, cargamos nuestro conjunto de datos en un dataframe, al cual, posteriormente, se le realiza una partición en dos dataframes *test_df* y *train_df* representando un 20 % y un 80 % respectivamente del total de las muestras.

Una vez realizada la división, incluimos un *Custom Image Dataset* propio que se encargará de aplicar las modificaciones necesarias y guardar los datos en dos conjuntos que se cargarán en sus respectivos *Data Loaders* encargados de proporcionar los datos segmentados en *batches* a nuestro modelo.

El proceso realizado por este método se encuentra esquematizado en el Algoritmo 2.

Algoritmo 2: Funcionamiento del *CustomImageDataset*

Entrada: *Dataframes* con rutas de audios y etiquetas del conjunto de datos.
Salida: Tensor del espectrograma de mel y etiqueta.

1: Obtener la ruta de audio y la etiqueta del *DataFrames*.
2: Extraer el espectrograma de Mel del audio ubicado en la ruta proporcionada.
3: Convertir el *array NumPy* del espectrograma en un tensor *PyTorch*.
4: Redimensionar los espectrogramas según sea necesario.
5: Normalizar los datos del espectrograma para tener una escala común.
6: Aplicar los aumentos de datos comentados.

Previo a realizar el entrenamiento, definimos algunos términos necesarios a la hora de desarrollar este proceso de forma adecuada y eficiente.

- **Criterio de pérdida**: Calcula la diferencia entre la salida predicha y la etiqueta real, por lo que se busca minimizar esta diferencia durante el entrenamiento.
- **Optimizador**: Ajusta los pesos del modelo para minimizar la función de pérdida.
- **Early Stopping**: Detiene el entrenamiento si no se producen mejoras significativas en la métrica de evaluación durante un número de épocas consecutivas.
- **Scheduler**: Ajusta la tasa de aprendizaje durante el entrenamiento, mejorando la convergencia y evitando sobreajuste.

Así, el proceso general de entrenamiento del modelo se resume en el Algoritmo 3.

Algoritmo 3: Proceso de entrenamiento del modelo

```
Inicializar la mejor precisión y F1-score a 0,0
Inicializar paciencia para detención temprana
Para cada época en el rango de épocas hacer:
    Entrenar el modelo con los datos de entrenamiento
    Calcular la pérdida y métricas de rendimiento en entrenamiento
    Evaluar el modelo con los datos de validación
    Calcular la pérdida y métricas de rendimiento en validación
    Actualizar la mejor precisión y F1-score si es necesario
    Guardar el modelo si hay mejora en las métricas de validación
    Verificar si se activa la detención temprana o early_stopping
```

Todo este proceso se reduce a las etapas de procesamiento de datos, conversión de las imágenes y entrenamiento del modelo para su posterior ajuste y validación. Este conjunto de fases suponen el principal flujo de trabajo del aprendizaje del modelo, por lo que para una mejor representación y esquematización lo podemos visualizar mediante la Figura 4.10.

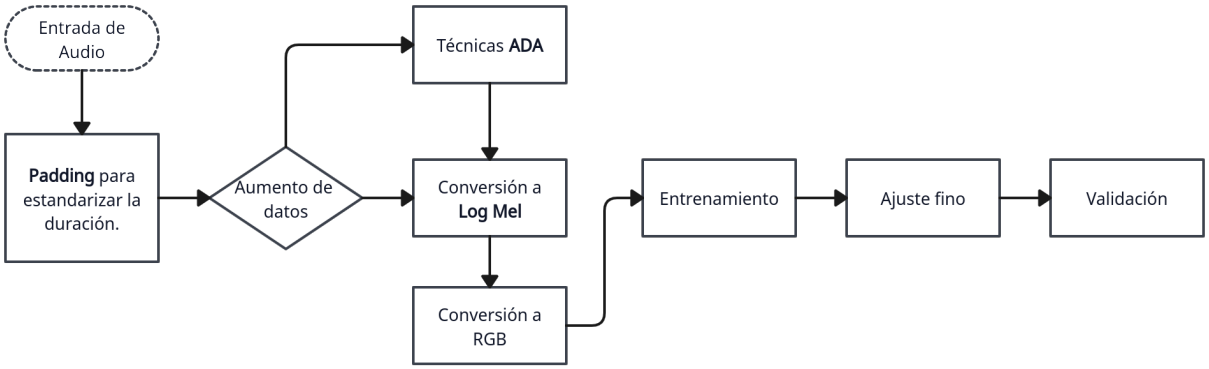

Figura 4.10: Flujo de trabajo del modelo.

Capítulo 5

Experimentos y Resultados

En este Capítulo desarrollamos y exponemos los diferentes resultados obtenidos por el modelo propuesto en el Capítulo 4 para cada uno de los ajustes y aumentos de datos aplicados sobre nuestro corpus de audios. Para ello, dividimos los experimentos realizados en dos grupos principales en base a la técnica utilizada para la estandarización del tamaño de los espectrogramas.

Previamente, cabe especificar que hemos utilizado una serie de librerías de *Python* que nos han facilitado la implementación, análisis y estudio de este tema.

Por un lado, hemos requerido de *PyTorch* [PyT22], la cual consiste en una biblioteca de aprendizaje automático que nos ha proporcionado las herramientas para la construcción y entrenamiento del modelo.

Por otro lado, hemos utilizado *NumPy* [Num22] para manipular y transformar matrices de datos, es decir para la manipulación de nuestras muestras de audio.

En el caso de *LibROSA* [M+22], se trata de una biblioteca de Python para el análisis y manipulación de audio, por lo que nos ha proporcionado herramientas para cargar archivos de audio, extraer características útiles o realizar las transformaciones comentadas anteriormente.

Además, *Matplotlib* [Mat22] ha resultado fundamental en la creación de gráficos para visualizar ondas de audio, espectrogramas, características del conjunto de datos, entre otros.

Por último, *Pandas* [Pan22] ha sido imprescindible para el análisis de datos, concretamente para tareas como la carga y procesamiento de nuestro conjunto de muestras de audio.

5.1. Audios con Redimensionamiento

Ya que es necesario tener un estándar en anchura y altura para realizar el entrenamiento por lotes, aplicamos un redimensionamiento o *resize* de forma que interpolamos los pixeles para conseguir un espectrograma de un tamaño común en el eje del tiempo. De esta forma, se añaden nuevas secciones de píxeles a nuestros espectrogramas pudiendo causar deformaciones o pérdidas de información relevante para la clasificación.

Cabe especificar que se eligió un estándar de 8 segundos en el eje del tiempo ya que,

como mucho, el audio de mayor duración de entre las muestras totales es de menos de 8 segundos, por lo que constituye la cota superior de la duración de nuestro conjunto.

El primer experimento aplicando la técnica de *resize*, concretamente realizando un redimensionamiento de la imagen a 224x224, no hace uso de ninguna de las técnicas ADA explicadas anteriormente, y se representa usando el espectrograma de mel. Con este planteamiento se consiguieron un 87,61 % de *f1-score* sobre el conjunto de validación, lo cuál supone una muy buena primera aproximación

Por otro lado, en el segundo experimento realizado con *resize* se realizó una conversión a MFCC para representar el espectrograma de audio, usando nuevamente una transformación a una imagen 224x224 y sin aplicar ninguna modificación ADA. Se obtuvieron, mediante este ajuste, una caida del *f1-score* a 82,09 % además de un gran aumento en la función de perdida o *loss* de hasta un 66,41 %. Esta función, también denominada función objetivo, nos presenta lo lejos que está en un momento dado la salida de nuestro modelo y el resultado esperado.

Como se puede observar en la Figura 5.1, el modelo con MFCC presenta un peor rendimiento de ambas métricas comparado con el resto de experimentos.

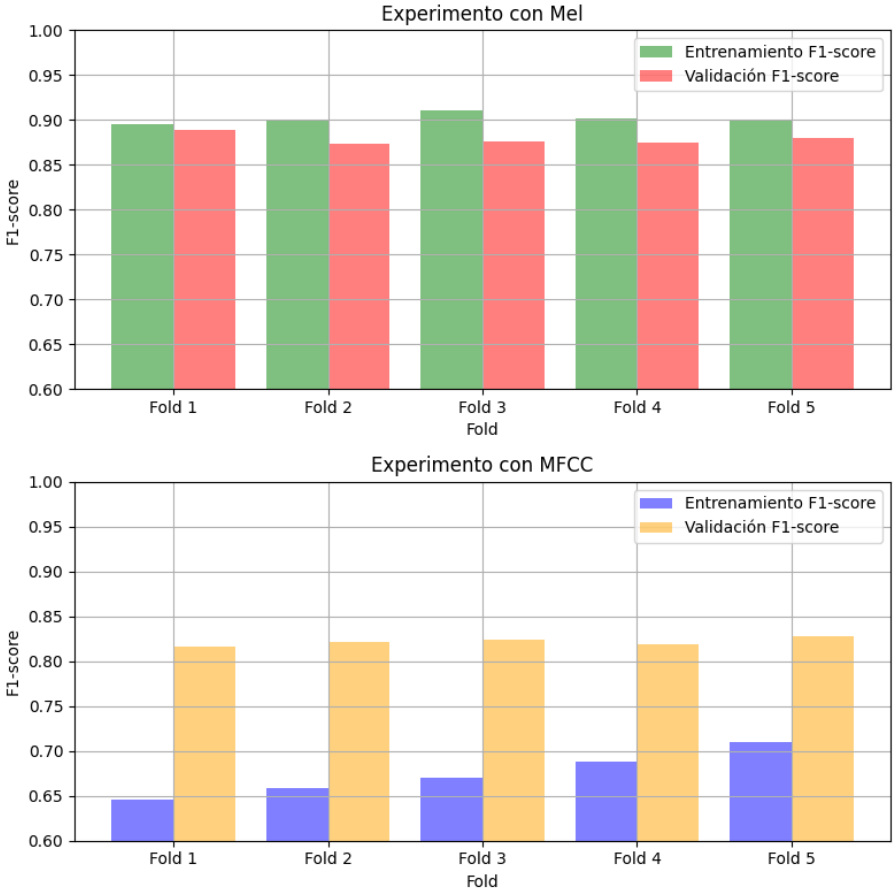

Figura 5.1: Comparativa del *f1-score* para los experimentos con redimensionamiento.

De esta forma, se concluye que una representación en formato Mel presenta una mayor capacidad de clasificación y análisis de las características del espectrograma, además de un mejor proceso de entrenamiento.

5.2. Audios Rellenados con Silencio

Posteriormente, decidimos cambiar el método de redimensionamiento de las imágenes. Para ello, se realizó la misma estandarización de la duración de los audios a un común de 8 segundos pero, en este caso, rellenando el espectrograma con un silencio hasta alcanzar la duración requerida.

En el primer experimento realizado con esta técnica utilizamos un espectrograma de mel para representar la frecuencia del audio, sin aplicar ningún ADA para observar y comparar la eficiencia de esta nueva técnica en comparación con el *resize* utilizado anteriormente.

De esta forma, mediante este nuevo enfoque, se obtuvo un *f1-score* de 89,13 % sobre 1 en la validación, no obstante, cabe destacar que se muestra un 99,66 % para el mismo valor en el proceso de entrenamiento. Esto puede indicarnos que se esta produciendo un sobre aprendizaje durante el entrenamiento, de forma que nuestro modelo puede estar memorizando el conjunto de datos y, consecuentemente, esta práctica resulte en un mal rendimiento del modelo ante nuevos datos.

El segundo experimento, ya habiendo observado una mejora con respecto a la técnica de *resize*, utilizamos nuevamente el espectrograma de mel para la representación del audio, una imagen de 224x224 completada con silencios, pero en este caso aplicándose aleatoriamente dos técnicas de ADA pertenecientes a la biblioteca *audio_augmentations* de *PyTorch*. De entre las técnicas utilizadas cabe destacar la modificación de adicción Ruido o *Noise* con una probabilidad del 50 %, cuya funcionalidad consiste en añadirle al audio un ruido de fondo; o la técnica de cambio de Tono o *PitchShift* con una probabilidad del 30 %, permitiendo modificar el tono del audio sin alterar el tempo original. Podemos observar en la Figura 5.2, el efecto que tiene la adición de ruido a un audio cuando lo visualizamos en su conversión a espectrograma,

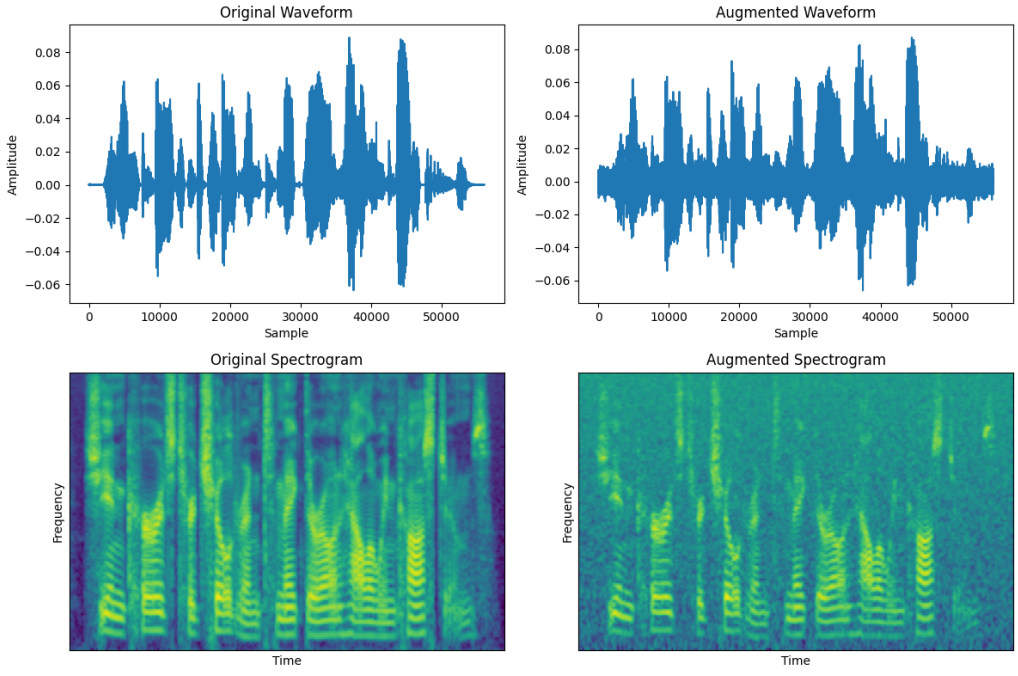

Figura 5.2: Comparativa de un audio original y su modificación con ruido gaussiano.

Al contrario que en el experimento sin ADA, en este modelo se pueden observar una serie de particularidades que nos indican una mejora más relevante de lo que pueda parecer en primera instancia.

Por un lado, en la Figura 5.3 se muestra una escasa mejora del *f1-score* que alcanza el 89,41 %. No obstante, en el proceso de entrenamiento se refleja una mayor calidad de la arquitectura al presentar una gran proximidad entre las métricas de validación y entrenamiento. Además, cabe destacar que este modelo ha ocupado un mayor número de épocas y una menor métrica de pérdida respecto al resto de experimentos, por lo que podemos deducir un mejora en cuanto a su capacidad de clasificación de una mayor variedad de muestras, una menor presencia de sesgos y un mejor análisis de las características relevantes del audio.

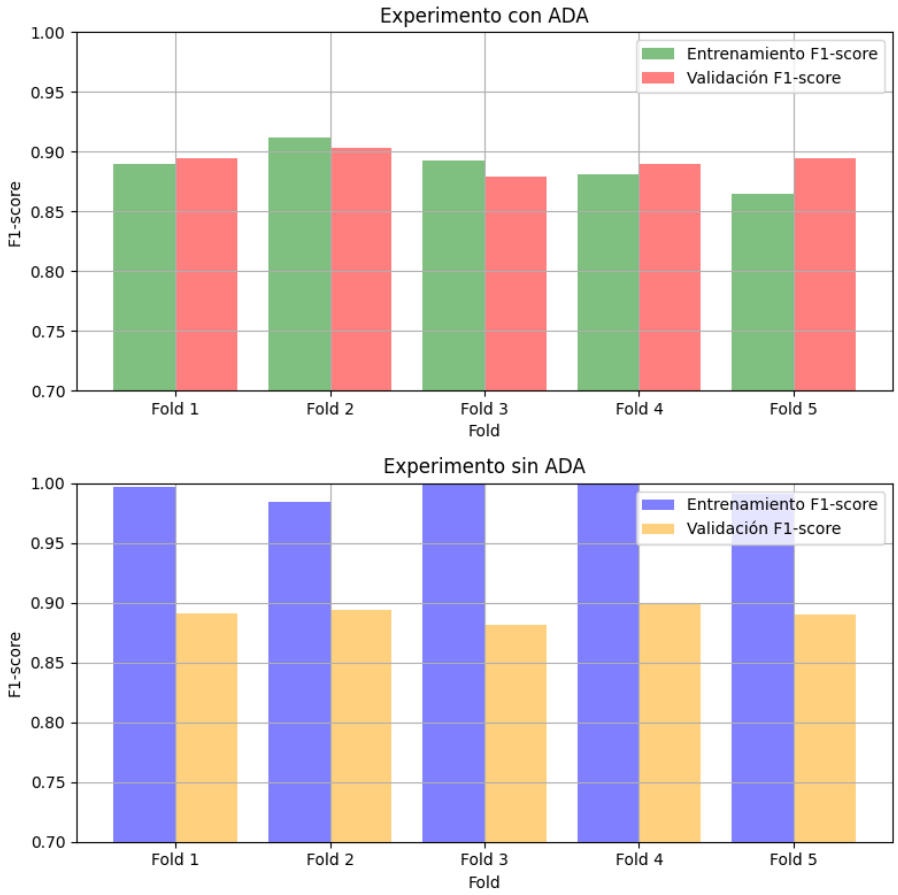

Figura 5.3: Comparativa del *f1-score* para los experimentos sin redimensionamiento.

Para el modelo al que se le aplicó ADA, se puede observar una proximidad constante al 90 % de *f1-score* mostrando, en todo momento, unas métricas muy similares entre entrenamiento y validación. Por el contrario, en el otro caso, no solo vemos unas métricas muy dispares entre ambos conjuntos sino que se observa un estancamiento de la validación pese a un supuesto gran rendimiento para el entrenamiento por lo que deducimos que este modelo esta realizando una memorización del conjunto de datos. Así, podemos destacar, como bien comentamos antes, una mayor versatilidad y robustez del modelo con ADA pese a que no se muestre tanta diferencia en el *f1-score* final en comparación con el otro modelo.

5.3. Comparativa de las Métricas

En las siguientes Tablas 5.1 y 5.2 se pueden observar de forma resumida las especificaciones de cada uno de los experimentos realizados.

Tabla 5.1: Especificaciones de los experimentos.

Nº Experimento	Redimensionado	ADA	Espectrograma
1 (BASE)	NO	NO	MEL
2	NO	NO	MEL
3	NO	SI	MEL
4	SI	NO	MEL
5	SI	NO	MFCC

De la misma forma, podemos agrupar el conjunto de métricas de la validación obtenidas para cada uno de los experimentos para facilitar el análisis comparativo de los resultados. Cabe especificar que se han incluido las métricas más relevantes y anteriormente comentadas como son el *f1-score*, la precisión o *acurracy* y la función de pérdida o *loss*, ya que consideramos que son las más relevantes para mostrar la efectividad del modelo.

Tabla 5.2: Métricas de los experimentos.

Nº Experimento	*F1-score* (%)	Precisión (%)	Pérdida (%)
1 (BASE)	77,42	76,67	66,25
2	89,13	86,81	55,58
3	89,41	86,69	52,51
4	87,61	85,54	54,61
5	82,09	77,56	66,41

Además, podemos visualizar de una forma más gráfica y concisa la optimalidad de los modelos basándonos en el *f1-score* de la validación a lo largo de las épocas del entrenamiento. En la siguiente Figura 5.4 se muestra un computo de la mediana, cota superior e inferior del *f1-score* para cada experimento para así analizar el conjunto de todos los resultados obtenidos para dicha métrica con el modelo en específico.

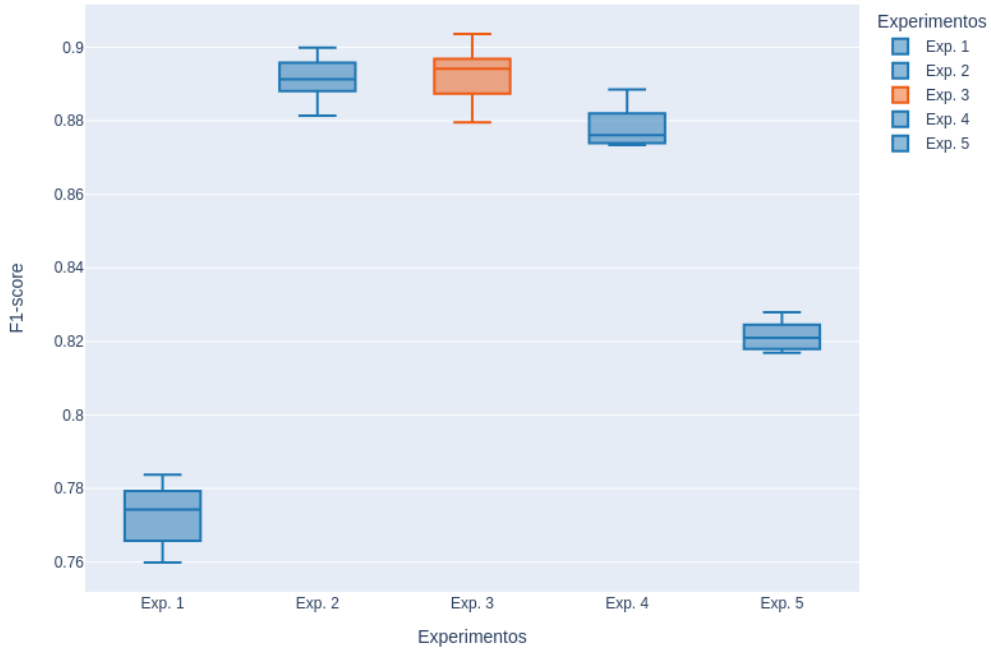

Figura 5.4: Resultados del *f1-score* de los experimentos.

Tras el análisis de los resultados, se destaca el experimento número 3, debido no solo a que obtuvo unas métricas ligeramente mejores, sino que además se considera más apto para audios externos al conjunto con el que ha sido entrenado.

Al haberse realizado un aumento de datos y, observándose el proceso de entrenamiento, se puede concluir que no se ha producido un sobre aprendizaje que pueda inducir en sesgos o mala praxis en la toma de decisiones.

Esta conclusión puede también observarse en el número de épocas que duró el entrenamiento de los *folds* de cada experimento hasta realizar una parada temprana, representado en la Tabla 5.3. Esta parada se produce por no observarse una mejora en las métricas de validación del entrenamiento, por lo que recupera el mejor modelo y no continua entrenando para evitar sobre aprendizaje.

Tabla 5.3: Duracion del entrenamiento por *folds*.

Nº Experimento	*Fold 1*	*Fold 2*	*Fold 3*	*Fold 4*	*Fold 5*
1 (BASE)	33	39	37	46	49
2	23	15	36	30	17
3	44	50	33	34	28
4	44	33	41	30	37
5	25	45	41	20	22

O de forma similar, en la Figura 5.5 realizamos una media de cada *fold* para ver una aproximación del proceso de entrenamiento en cada experimento.

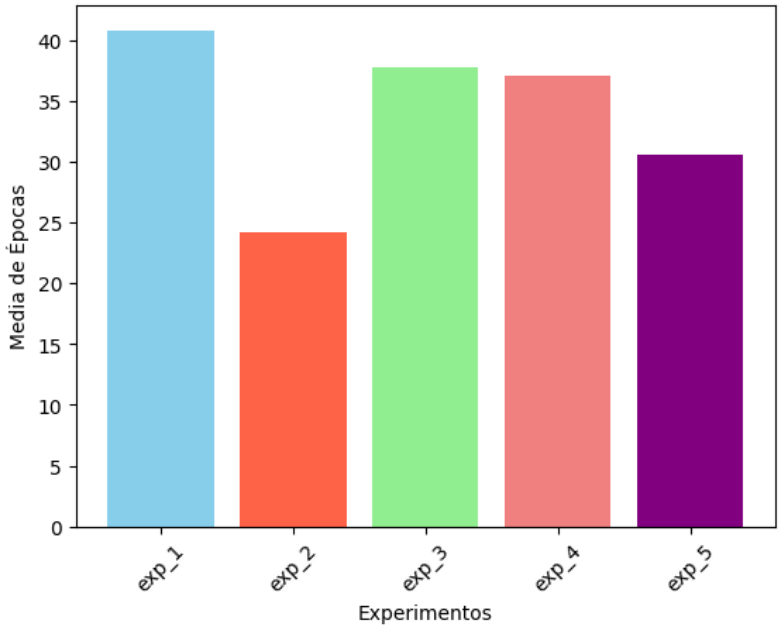

Figura 5.5: Comparativa de épocas.

5.4. Resultados Finales

Tras haber realizado todos los experimentos comentados, se observa un mejor rendimiento del modelo realizado sin *resize* y con técnicas ADA. Para ello podemos observar su matriz de confusión en la Figura 5.6.

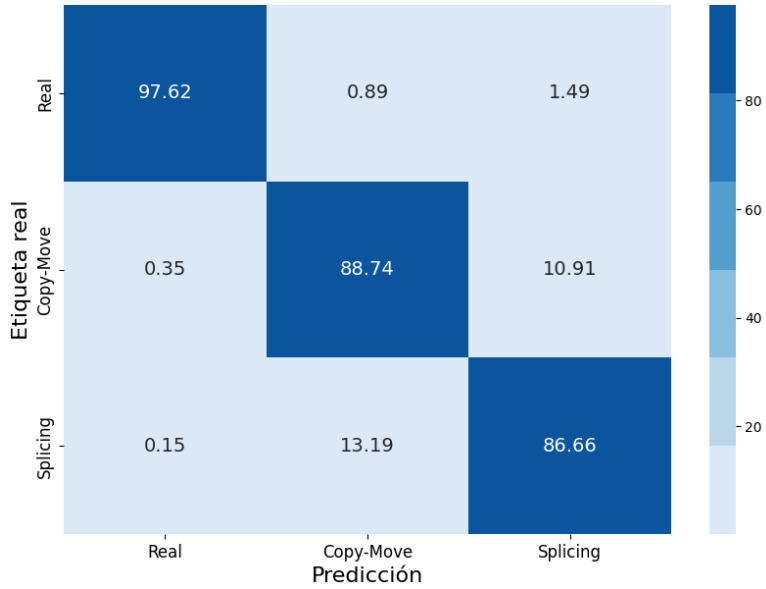

Figura 5.6: Matriz de confusión del mejor modelo.

En esta matriz se muestran las métricas del modelo seleccionado completando el entrenamiento sin realizar particiones sobre el conjunto de datos. Se observa un porcentaje de al menos el 86 % en la diagonal de la matriz de confusión, donde se agrupan aquellas clasificaciones realizadas correctamente.

Además, tras este nuevo entrenamiento exhaustivo sobre todas las muestras, podemos analizar las métricas obtenidas para el modelo final segmentadas por categoría y esquematizadas en la siguiente Tabla 5.4.

Tabla 5.4: Métricas del mejor modelo.

Etiqueta	Precisión (%)	Sensibilidad (%)	*F1-Score* (%)	Muestras
Original	97,0	97,6	97,3	336
Copy-move	86,9	88,7	87,8	2016
Splicing	88,6	86,7	87,6	2017

Capítulo 6

Conclusiones y Trabajo Futuro

En esta sección se presentan las conclusiones a las que se ha llegado tras todos los procesos explicados anteriormente, en concreto, la experimentación con nuestro modelo desarrollada en el Capítulo 5 y la comparación de los resultados obtenidos frente a los trabajos ya existentes documentados en el Capítulo 3.

6.1. Conclusiones

Analizados los resultados obtenidos por el mejor modelo, podemos concluir que existe un mejor rendimiento para la clasificación de clips reales, ya que se consigue un alto porcentaje de acierto en la detección de audio no manipulado. Todo esto a pesar de que en el conjunto de datos utilizado para el entrenamiento, la cantidad de archivos originales es considerablemente menor que la cantidad de archivos modificados.

También, se puede sacar en claro que nuestro modelo detecta ambas modificaciones, cosa que los anteriores trabajos no consiguen, aunque en la matriz de confusión de la Figura 5.6 se puede observar que existe un mayor número de Falsos Positivos (FP) respecto a Falsos Negativos (FN) generado por la clasificación de la manipulación en concreto. Esto se debe a que se encuentran más errores a la hora de diferenciar si es *copy-move* o *splicing*, por lo que existe un leve problema de clasificación entre ambas modificaciones.

Pese a este pequeño margen de error a la hora de clasificar entre ambas manipulaciones, nuestro modelo presenta unas métricas muy elevadas, especialmente de *f1-score*, y un gran rendimiento en comparación con los trabajos referenciados y estudiados. Además, debido a que se generó un conjunto de datos propio, cabe destacar que se incluye una gran variedad de datos modificados pudiendo realizar un aprendizaje exhaustivo teniendo en cuenta una multitud de distintas apariciones de estas modificaciones.

Asimismo, podríamos intuir que en el caso de simplificarse el estudio a realizar solamente una clasificación entre audio real y audio modificado o utilizando únicamente un tipo de alteración, como es el caso de la mayoría de artículos expuestos en el Capítulo 3, se observaría un mayor rendimiento del modelo respecto al resultado actual.

No obstante, este estudio nos permite comprender la capacidad que tienen este tipo de modelos en el proceso de clasificación, suponiendo un enfoque más eficiente que la mayoría de otras formas de abordarse. Además, cabe enfatizar la escalabilidad que presenta el modelo, la cual permite un crecimiento tanto en una mejora de rendimiento como en la

posibilidad de aumentar las categorías a clasificar.

Por ello, concluimos que el estudio, experimentación y modelo resultante han conseguido alcanzar las expectativas del trabajo e incluso sobresalir en cuanto a personalización del estudio al utilizar un conjunto de datos propio.

6.2. Trabajo Futuro

Con respecto a los trabajos futuros del estudio, pueden señalarse los siguientes:

- Utilizar arquitecturas más avanzadas con mayor número de parámetros con el objetivo de poder llegar a conseguir que el modelo aprenda patrones más complejos y pueda ser más flexible con el objetivo de reducir el sesgo al mínimo.
- Aplicar un mayor número de técnicas de ADA con el fin de tener más datos con los que entrenar el modelo. De esta manera se podría disminuir la probabilidad de que el modelo memorice patrones en los datos, y aprenda características mas generales y robustas.
- Entrenar el modelo con mayores recursos hardware. Esto podría mejorar el rendimiento del entrenamiento, reduciendo el tiempo de este, aumentando la capacidad de procesamiento, mejorando la escalabilidad y aumentando la precisión y eficiencia. Como ejemplo de mejora de recursos hardware sería utilizar una *Graphics Processing Units* (GPU) potente o una *Tensor Processing Units* (TPU), que ofrece rendimientos superiores a las GPU en tareas de DL.
- Realizar un análisis exhaustivo de la extracción de características que realiza nuestro modelo y que utiliza para la clasificación. Este proceso sería posible gracias a la implementación de Mapas de Activación por medio del método *Gradient-weighted Class Activation Mapping* (Grad-CAM), de forma que se modificase el conjunto de muestras, ajustasen los hiperparámetros o aplicasen nuevas técnicas de ADA para redirigir el estudio y evitar la activación del modelo en base a características irrelevantes.
- Ampliar el número de categorías de manipulaciones de audio. Ya que, gracias a la escalabilidad del modelo y suponiendo que se hubieran realizado los trabajos futuros mencionados, se podría aumentar la cantidad de alteraciones detectadas por el modelo, incrementando su potencia y su uso como herramienta práctica por empresas y particulares.
- Localizar la posición de la modificación aplicada sobre el audio. De esta forma, no solo se detecta y clasifica la muestra sino que se habilita mostrar la ubicación y duración de la alteración encontrada.

Capítulo 7

Contribuciones personales

En este Capítulo 7 se expondrán los desafíos, recursos y aportaciones individuales que ha requerido y aportado cada uno de los integrantes de este proyecto durante su realización.

7.1. José Antonio Ruiz Heredia

Para el desarrollo de este TFG a lo largo del presente curso, puedo organizar el planteamiento y resolución de la temática en tres principales desafíos que han surgido durante el progreso del mismo.

7.1.1. Marco Teórico

Por un lado, debía realizar un estudio previo para contextualizar y ampliar mi conocimiento sobre el tema que se iba a tratar. En primera instancia, cabe destacar que mi primer contacto con el campo de la IA fue al cursar las asignaturas de **IA1** e **IA2**. No obstante, este trabajo requería de una ampliación sobre el funcionamiento e implementación de técnicas de DL.

Primero de todo opté por el Curso [Staha], el cual aportaba una extensa variedad de vídeos sobre diferentes conceptos de ML. No obstante, también necesitaba una profundización teórica sobre las RNA, para lo que elegí el Curso [Ude23] en el que se aborda más extensamente las distintas técnicas de DL, y cuyo conocimiento adquirido se expone resumidamente en el Capítulo 2. Cabe destacar otras fuentes que revisé, como los vídeos del MIT impartidos en 2023 y compartidos en la plataforma de *Youtube* [oTM23], o artículos web como [JJ19], [Sim20] o [JM15], entre otros.

Seguidamente, también debía afrontar la temática del tratamiento de audios, ya que serían los datos que estaría analizando nuestro modelo y cuyas características debía conocer y saber modificar para poder realizar un manejo adecuado de las muestras.

De esta forma, me ayude mayoritariamente de los dos artículos de la serie *Audio Deep Learning Made Simple* parte 1 [KD21a] y parte 2 [KD21b], donde se introducen conceptos sobre audio enfocados principalmente en su posterior uso en modelos de DL.

7.1.2. Conjunto de Datos

Por otro lado, la siguiente gran problemática fue el propio conjunto de muestras para el reconocimiento. Para ello, me encargué de desarrollar un primer código para generar un conjunto de audios modificados realizando cortes y mezclas aleatorias sobre el corpus de clips del TIMIT. Sin embargo, debido a su practicidad, buenos resultados y eficiencia, acabé implementando el código de concatenación expuesto en el Algoritmo 1.

Una vez obtenidos nuestros datos sobre los que trabajar, seguimos una serie de pasos para realizar un estudio exhaustivo de los audios. Concretamente me encargué del código para generar el archivo *Comma Separated Values* (CSV) necesario para el entrenamiento del modelo, que se encargaba de recopilar las direcciones de las muestras, etiquetas y otras métricas como la duración de cada audio. Además, utilicé posteriormente esta información para generar gráficas que mostrasen las características del conjunto de datos, las cuales están expuestas en la Figura 4.1 y la Figura 4.2 al principio del Capítulo 4.

A su vez, también realicé algunos de los apartados del estudio del preprocesamiento de audio, que se encuentran agrupados en un *Jupyter Notebook*, con el objetivo de obtener hiper parámetros como *n_fft* o *hop_length* para la generación de espectrogramas y *n_mels*, *n_mfcc* o *n_lfcc* para los otros formatos de representación. De esta forma, tras este estudio, pudimos elegir e introducir los espectrogramas adecuados para el correcto aprendizaje del modelo. Igualmente, este estudio me permitió incluir más figuras a lo largo del Capítulo 4 para ejemplificar el trabajo que se había realizado.

Entre los principales recursos utilizados para este preprocesamiento se encuentran el Artículo [Dev23], algunos tutoriales de la página web de *Tensorflow* específicos sobre audio como el Artículo [Ten22] o explicaciones sobre manejo de datos de audio con *Python* como en el Artículo [Hug22].

7.1.3. Aprendizaje del Modelo

Por último, una vez realizados los estudios y análisis de los datos, requeríamos del desarrollo y entrenamiento del modelo. De esta forma, tras haber revisado algunos artículos similares, nos decidimos por el modelo pre entrenado comentado al final del Capítulo 4.

Concretamente realicé algunos cambios en la función de carga de datos, añadí la función de redimensionamiento, completé la normalización de las muestras e incluí los hiper parámetros finales obtenidos en el preprocesamiento. Para el correcto entendimiento del funcionamiento de la carga de datos, reflejé este proceso en el Algoritmo 2. Asimismo, también diseñé el Algoritmo 3 para ilustrar esquemáticamente el aprendizaje del modelo y, a su vez, el diagrama de flujo que muestra el proceso de trabajo completo, el cual se presenta en la Figura 4.10.

Posteriormente, debido a las especificaciones de *hardware* que requería el aprendizaje del modelo, el entrenamiento se realizó en el equipo del profesor Daniel Povedano Álvarez. Una vez realizados los distintos experimentos, me encargué de graficar y agrupar en tablas las métricas comparativas presentes en el Capítulo 5.

Además, se realizó un entrenamiento más profundo y con mayor cantidad de muestras para el modelo elegido de los experimentos, por lo que, una vez finalizado, desarrollé el apartado final del Capítulo 5 incluyendo la Matriz de Confusión 5.6 y el resumen de las métricas obtenidas por este en la Tabla 5.4.

Por último, tras completar el grueso del trabajo, ambos miembros realizamos unas conclusiones sobre el proceso y resultados obtenidos por el modelo, además del planteamiento de tareas futuras para la ampliación del estudio. Todo ello se encuentra expuesto en el Capítulo 6.

7.2. Néstor Antonio Marín Gómez

Procedo a explicar mi contribución personal a este TFG.

7.2.1. Marco Teórico

Como alumno de último curso del Grado de Ingeniería de Computadores, conocía los conceptos de la IA, debido a que había cursado la asignatura de *Sistemas Inteligentes*, pero con ese conocimiento no era necesario, por lo que antes de empezar con el trabajo de investigación del trabajo ya existente, tuve que revisar diversos cursos para refrescar y ampliar mis conocimientos sobre la IA, el ML, y el DL. El primer curso que revisé fue el curso anteriormente mencionado [Ude23], de la plataforma de cursos en línea *Udemy*, que se centra en la explicación de las RNA. También fueron muy útiles los vídeos del canal de *YouTube* de Pepe Cantoral [CC] , en el cual hay numerosos vídeos sobre DL, y .

A mitad del desarrollo del trabajo, empecé las prácticas de mi grado en una empresa llamada *Making Science*, y pude realizar tareas de entrenamiento de modelos de IA que utilizaban audio, por lo que eso también me ayudó para poder entender mas a fondo los conceptos y parámetros usados en los trabajos que usan audio. Algunos de los trabajos que proyectos utilizados fueron *DeepFilterNet* para limpieza de audio, *CoquiAI* para el tema de entrenamiento de modelos que convierten texto en audio. Y *Gradio* para la realizacion de interfaces gráficas sencillas para el uso de modelos de DL.

7.2.2. Conjunto de Datos

Cuando ya se habían vez se tenía el conjunto de datos creado, y su respectivo archivo CSV, me encargué de desarrollar el primer código que eligiese aleatoriamente un audio original existente en ese archivo CSV, buscase el audio equivalente con *splicing*, y el audio equivalente con *copy-move*, y representara los espectrogramas de los tres audios, con el objetivo de que se pudieran ver gráficamente las diferencias entre los tres. Para ello hice un archivo que usaba la biblioteca *Matplotlib* [Mat22] para la representación gráfica, y otro que utilizaba la biblioteca *Librosa* [M+22] para poder tener mas de una referencia visual.

En la parte de experimentación sobre el conjunto de datos previa al entrenamiento del modelo, me encargué de aplicar algunas de las técnicas de ADA utilizando la biblioteca *audiomentations*, la cual tiene una gran cantidad de técnicas de aumento de datos para fragmentos de audio, por lo que se utilizaron algunas de ellas como *Gain*, *GaussianNoise*, *Reverse*, representando posteriormente esos audios modificados con su espectrograma para ver la diferencia tras la aplicación de esas técnicas de ADA.

7.2.3. Aprendizaje del Modelo

Cuando ya se había desarrollado el código de preprocesamiento para posteriormente entrenar el modelo, tuvimos que entrenarlo con ayuda de nuestro profesor Daniel Povedano Álvarez, y tras los diversos experimentos que se utilizaron, explicados en la Tabla 5.1, el profesor nos facilitó las métricas obtenidas en archivos con formato *.pkl*, por lo que tuve que desarrollar el código que cogiese esos archivos y los pudiese agrupar todos en una estructura válida. Esos resultados pertenecen a la Tabla 5.2. Ese código también representaba gráficamente las métricas obtenidas en los experimentos, las cuales están reflejadas en la figura 5.4.

Además de las partes anteriormente explicadas, mi compañero y yo trabajamos conjuntamente a lo largo del desarrollo del trabajo, realizando las tareas necesarias para conseguir el resultado final obtenido, consiguiendo un modelo de detección de modificaciones de audio usando técnicas de DL.

Capítulo 8

Introduction

8.1. Motivation

Forensic audio science began with the invention of the phonograph by Thomas Edison in 1877. This technology allowed sound to be recorded and reproduced through a wax cylinder, marking a milestone in communication. Subsequently, throughout the 20th century, technological advances led to enormous improvements in recording, making them increasingly clear and precise, which opened up enormous potential for using audio as a research tool.

One of the first applications of forensic audio science was during World War I. Technological advances at that time allowed soldiers and commanders to capture battle sounds and news broadcasts on phonographic records. These recordings were of vital importance in the investigation into the responsibilities for the start of the war, as Germany was accused of starting the armed conflict. A specialized working group was then formed to analyze more than thousands of recordings of conversations recorded between diplomats and politicians. By evaluating these recordings, evidence could be gathered to support or refute these accusations.

This science has also been very useful in criminal investigations. A historical case took place in 1932, when the German police, through a meticulous sound analysis, managed to identify a single voice present in two recordings from the crime scene. This evidence led to the incrimination of the perpetrator.

These examples demonstrate the importance of forensic audio science in validating testimony and strengthening legal arguments. As technology continues to evolve at a dizzying pace, the techniques and applications of this discipline are also expanding at an accelerated rate. However, this evolution is not without its challenges, as it demands constant adaptation on the part of professionals and the institutions involved.

Over the years, forensic science has experienced both successes and difficulties, as there are elements in high-quality recordings that can make it difficult to accurately study the original characteristics of the audio. These challenges are further intensified in lower quality recordings or those that have been altered or manipulated.

The 21st century has witnessed the meteoric rise of forensic audio science as a fundamental tool for the pursuit of truth and justice. As technology advances and recording devices become increasingly ubiquitous, the possibilities for capturing and analyzing audio

evidence multiply exponentially. Law enforcement, investigators, and legal professionals have found a great ally in forensic audio science. Its ability to identify voices, analyze acoustic content, and extract information from audio recordings of varying quality makes it an indispensable tool in a wide range of cases.

The existence of smartphones, digital recorders, and other sound capture devices has generated an exponential increase in the available acoustic evidence. This abundance of data, while offering endless possibilities for research, also presents challenges in terms of its management, analysis, and authentication.

However, the increase in acoustic evidence and the fine line between truth and privacy demand ethical and responsible handling of this discipline to ensure respect for individual rights. The future of forensic audio science is promising, with enormous potential to continue evolving and contributing to justice in an increasingly digitized world.

On the other hand, when forensic audio science was born, the analysis of sound recordings fell on the auditory capacity and experience of human experts. In the two previous examples that have been cited, about World War I and the 1932 trial, these experts were responsible for manually identifying features such as voice, environmental noise, etc. However, this manual approach was not without its limitations, as there were various risks that could affect the accuracy of these analyzes, such as the subjectivity of auditory perception, auditory fatigue, etc.

On the other hand, in its early stages, Artificial Intelligence (AI) was not useful for extracting and identifying these characteristics. But in 1958, the appearance of the Perceptron [Sim20] marked a milestone in the world of AI, as it laid the foundations for the development of computational systems capable of processing and analyzing information in a similar way to humans. Since then, AI, with the help of technological advances, has begun to grow at a dizzying speed, and with it its applications in the world of forensic audio science.

This implied that now *Artificial Neural Network* (ANN) could be used in the detection of both existing features in these audios and new features that are difficult for humans to perceive, since these networks are inspired by the functioning of the human brain and can process large amounts of data and learn to identify complex patterns more accurately than human experts.

Therefore, with the help of AI, the task of analyzing and extracting the characteristics of an audio, to then decide whether that audio is original or has been manipulated for some criminal purpose, would be possible. To do this, it would only be necessary to transform that audio into an image and use that image as an input example so that AI could train and learn to detect when an audio is original or not.

8.2. Context

This Bachelor's Degree Final Project is part of a research project entitled *Child protection centered strategies to fight against sexual abuse and exploitation - ALUNA*, approved by the *European Comission* (EC) in the call ISF-2021-TF1-AG-CYBER under the grant agreement number 101084929 and in which participates as project coordinator the GASS of the UCM (Group 910623 of the catalog of research groups recognized by the UCM, https://gass.ucm.es).

8.3. Object of the Investigation

When listening to a sound, the sound waves are captured by the pinna and conducted through the ear canal to the eardrum. This thin tissue vibrates to the rhythm of the waves, transmitting the acoustic information to the middle ear, and there the mechanical energy of the sound is transformed into electrical signals, which travel through the auditory nerve until they reach the brain, where all the acoustic information pertaining to that sound is processed. This implies the difficulty of differentiating between the originality or falsification of an audio, since it can be located at a point imperceptible to the human ear.

The following work will focus on trying to get a model that extracts the characteristics of an audio in the best way, in order to be able to pass it to a model of AI, with the aim of training it to learn as well as possible to detect whether an audio is original or forgered.

8.4. Workplan

The development of this work has been carried out in three main phases:

1. **Research**:

   The first steps in the development of this work, which took place during the first 4 months, consisted of making contact about the work to be done.

   First of all, there was an initial meeting in which the teachers of the department and the students met so as to establish the starting point, the objectives of this work, the necessary knowledge for the development of the project and the first guidelines to take into account. After this, we reached a consensus and agreed to do periodic meetings to review the status of the development of the work. We were also provided with a Google Drive folder on which we should carry out our work, including numerous resources in case we needed to consult support material to consolidate some of the concepts necessary to continue with our development.

   Although we had access to the supporting material, it was necessary to look for more resources to consult, so we reviewed existing works related to our research, and summarized several of those works, reviewing the data set used, the proposed methods, the experimental techniques, and the results and conclusions of each one.

   In the end, we reached several conclusions about what they wanted and how they wanted to achieve it, the team began to prioritize the development.

2. **Development**:

   Once the necessary knowledge was acquired, and having a clear understanding of the existing methodologies, without neglecting the research, we began to develop the proposed work. This process lasted about three months. Firstly, we began choosing a set of audio data, and developing a code to modify it by applying the forgery techniques that will be developed throughout the work.

   Subsequently, we began experimenting with these data to determine the best way to represent, transform, and modify them in order to provide them to the DL model.

   During this phase advanced concepts of the *Python* programming language and libraries such as *Matplotlib*, *Librosa*, *TensorFlow*, *PyTorch*, etc. were investigated.

3. **Results**:

   Ultimately, once the analysis of the pre-processed data had been completed, we began experimenting with the chosen model with the pre-processed samples, with the aim of achieving results that were superior to those of the works consulted.

   During this experimentation period, the necessary adjustments were made to the parameters provided to model in order to achieve the best possible metrics.

8.5. Struture of the Work

The rest of the work is organized in 9 chapters with the following structure:

Chapter 1 makes an introduction of the work, indicating the motivation to carry it out, the context, the object of the research and the work plan carried out.

Chapter 2 contains an explanation of the theoretical framework and the concepts necessary for the understanding of the work. This chapter is divided into two parts: on the one hand, the techniques and processes involved in learning the model and, on the other hand, the audio representation and treatment to be able to handle the data of the study.

Chapter 3 begins with an introduction to the study of audio manipulation. Then, it is explained what an audio manipulation is, the objectives of this, and the types of forgery according to the part that is modified. Afterwards, the type of forgery on which the work is based is explained in depth, and a compilation of the most consulted existing works in the research stage is made. From these works, the data set used, the proposed method, the experimentation carried out, and the results or conclusions are summarized. Finally, the limitations found in those works that have been consulted are developed, with the aim of improving them in our proposed work.

Chapter 4 shows the complete process carried out, firstly explaining the methodology applied for the creation of an own dataset to work on, explained in the algorithm 1. Then the audio preprocessing process is explained along with images illustrating the difference in results depending on the parameter values. The ADA techniques used are also explained, in the figure **??**.

Chapter 5 describes the experiments performed to evaluate the effectiveness of the algorithms proposed in Chapter 4 and presents the results obtained.

Chapter 6 shows the main conclusions of this work, the future research lines and the publications derived from this work.

Chapter 7 collects the specific personal contributions of each student in a more punctual way, taking into account that the work has been correctly distributed among the students who have carried out the work.

Chapters 8 and 9 are the English translations of Chapter 1 and Chapter 6.

Capítulo 9

Conclusions and Future Work

This section will explain the conclusions reached after all the processes explained above, in particular, the experimentation with our model explained in Chapter 5 and the comparison of the results obtained with the existing works documented in Chapter

9.1. Conclusions

Analyzing the results obtained by the best model, we can conclude that there is a better performance for the classification of real clips, since it is achieved a high percentage of success in the detection of non-manipulated audio. This is despite the fact that in the dataset used for training, the number of original files is much smaller than the number of modified files.

It is also clear that our model detects both modifications, something that previous works do not achieve, although in the confusion matrix in Figure 5.6 it can be observed that there is a greater number of FP with respect to FN generated by the classification of the particular manipulation. This is because more errors are found when differentiating between *copy-move* and *splicing*, so there is a slight classification problem between the two forgeries.

Despite this small margin of error in the classification between the two manipulations, our model has very high metrics, especially in *f1-score*, and a high performance compared to the referenced and studied works. In addition, since we have generated our own dataset, it includes a large variety of modified data so that it can perform an exhaustive learning taking into account a variety of different occurrences of these modifications.

Likewise, we could intuitively assume that if the study were simplified to perform only a classification between real and forgered audios, or using only one type of modification, as is the case in most of the articles presented in Chapter 3, a better performance of the model would be observed with respect to the current results.

However, this study allows us to understand the capacity of this type of models in the classification process, assuming a more efficient approach than most other ways of approaching it. In addition, it is worth mentioning the scalability of the model, allowing a growth both in better performance and in the capacity to increase the categories to be classified.

Therefore, we conclude that the study, the experiments and the resulting model have

managed to meet the expectations of the work and even excel in terms of customization of the study by using a custom dataset.

9.2. Future Work

In terms of future work on the study, the following can be noted:

- Using more advanced architectures with a greater number of parameters to allow the model to learn more complex patterns and be more flexible to minimize bias.
- Applying a larger number of techniques in order to have more data to train the model. This would reduce the likelihood of the model to memorize patterns in the data and learn more general and robust features.
- Training the model with more hardware resources. This could improve the training performance by reducing the training time, increasing the processing power, improving the scalability, and increasing the accuracy and efficiency. An example of improved hardware resources would be using a powerful GPU or a TPU, which offers better performance than GPUs for DL tasks.
- Performing an exhaustive analysis of the feature extraction performed by our model and used for classification. This process would be possible thanks to the implementation of activation maps by means of the Grad-CAM method, in order to modify the set of samples, adjust the hyperparameters or apply new techniques of ADA to redirect the study and avoid the activation of the model based on irrelevant features.
- Increasing the number of categories of audio manipulation. Since, due to the scalability of the model and assuming that the mentioned future work has been carried out, the number of forgeries detected by the model could be increased, improving its power and allowing its use as a practical tool by companies and individuals.
- Locating the position of the modification applied on the audio. Therefore, not only the model detects and classifies the sample, but it is also able to show the location and duration of the forgery found.

Bibliografía

[Agg18] C.C. Aggarwal. *Neural Networks and Deep Learning*. Springer, 2018.

[AMA+22] Mansour Alsulaiman, Ghulam Muhammad, Bencherif Mohamed Abdelkader, Awais Mahmood, and Zulfiqar Ali. King Saud University Arabic Speech Database. https://hdl.handle.net/11272.1/AB2/4YVL4A, 2022.

[CC] Jose Antonio Cantoral-Ceballos. Pepe Cantoral, Ph.D. https://www.youtube.com/c/pepecantoralphd/videos.

[DDS+09] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, June 2009.

[Dev23] Devopedia. Audio Feature Extraction. https://devopedia.org/audio-feature-extraction, 2023.

[Fer95] E. A. Fernández. *Artificial Intelligence: Its Scope and Limits*. Publisher Name, Publisher Address, 1995.

[Hal16] Nawar Halabi. *Modern Standard Arabic Phonetics for Speech Synthesis*. PhD thesis, University of Southampton, 2016.

[Hug22] Huggingface. Introduction to Audio Data. https://huggingface.co/learn/audio-course/chapter1/audio_data, 2022.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[IABA17] Muhammad Imran, Zulfiqar Ali, Sheikh Tahir Bakhsh, and Sheeraz Akram. Blind Detection of Copy-Move Forgery in Digital Audio Forensics. *IEEE Access*, pages 12843–12855, 2017.

[JJ19] Medium Jiwon Jeong. The Most Intuitive and Easiest Guide for Convolutional Neural Network. https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480, January 2019.

[JM15] M.I. Jordan and T.M. Mitchell. Machine Learning: Trends, Perspectives and Prospects. *Science*, 349(6245):255–260, July 2015.

[JPR19] Shital Jadhav, Rashmika Patole, and Priti Rege. Audio Splicing Detection Using Convolutional Neural Network. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–5, 2019.

[KD21a] Medium Ketan Doshi. Audio Deep Learning Made Simple (Part 1): State-of-the-Art Techniques. https://towardsdatascience.com/audio-deep-learning-made-simple-part-1-state-of-the-art-techniques-da1d3dff2504, 2021.

[KD21b] Medium Ketan Doshi. Audio Deep Learning Made Simple (Part 2): Why Mel Spectrograms Perform Better. https://towardsdatascience.com/audio-deep-learning-made-simple-part-2-why-mel-spectrograms-perform-better-aad889a93 2021.

[M+22] Brian McFee et al. Librosa. https://librosa.org, 2022.

[Mat22] Matplotlib Development Team. Matplotlib. https://matplotlib.org, 2022.

[Mat24] MathWorks. Introducción a la convolución. https://es.mathworks.com/discovery/convolution.html, 2024.

[MIoTMTI93] SRI International (SRI) Massachusetts Institute of Technology (MIT) and Inc. (TI) Texas Instruments. TIMIT Acoustic-Phonetic Continuous Speech Corpus. https://catalog.ldc.upenn.edu/LDC93S1, 1993.

[Num22] NumPy Contributors. Numpy. https://numpy.org, 2022.

[oTM23] Massachusetts Institute of Technology (MIT). MIT Deep Learning 6.S191. https://www.youtube.com/watch?v=QDX-1M5Nj7s&list=PLTZ1bhP8GBuTCqeY19TxhHyrwFiot42_U&ab_channel=AlexanderAmini, 2023.

[Pan22] Pandas Development Team. Pandas. https://pandas.pydata.org, 2022.

[PyT22] PyTorch Contributors. Pytorch. https://pytorch.org, 2022.

[PZL12] Xunyu Pan, Xing Zhang, and Siwei Lyu. Detecting Splicing in Digital Audios Using Local Noise Level Estimation. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1841–1844, 2012.

[Sim20] Simplilearn. What is Perceptron. https://www.simplilearn.com/what-is-perceptron-tutorial, January 2020.

[Spi21] Janne Spijkervet. Spijkervet/Torchaudio-Augmentations. https://zenodo.org/record/4748582, 2021.

[Staha] Stanford University, Coursera. Programa Especializado: Aprendizaje Automático. https://coursera.org/specializations/machine-learning-introduction, sin fecha. Recuperado de Coursera.

[Ten22] Tensorflow. Preparación y Aumento de Datos de Audio. https://www.tensorflow.org/io/tutorials/audio?hl=es-419, 2022.

[Ude23] Udemy. Deep Learning de A-Z: Redes Neuronales en Python desde Cero. https://www.udemy.com/course/deep-learning-a-z, 2023. Recuperado de Udemy.

[Uni] Princeton University. WordNet. https://wordnet.princeton.edu/.

[UTU23] Beste Ustubioglu, Gul Tahaoglu, and Guzin Ulutas. Detection of Audio Copy-Move Forgery with Novel Feature Matching on Mel Spectrogram. *Expert Systems with Applications*, 213:118963, 2023.

[ZCWM17] Hong Zhao, Yifan Chen, Rui Wang, and Hafiz Malik. Audio Splicing Detection and Localization Using Environmental Signature. *Kluwer Academic Publishers*, page 13897–13927, June 2017.

[ZW22] Z. Zeng and Z. Wu. Audio Splicing Localization: Can We Accurately Locate the Splicing Tampering? In *2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 120–124. IEEE, December 2022.

[ZZY22] Z. Zhang, X. Zhao, and X. Yi. Aslnet: An Encoder-Decoder Architecture for Audio Splicing Detection and Localization. *Security and Communication Networks*, 2022.