Detección de Contenido Sensible en Audio usando Técnicas de Aprendizaje Profundo

Sensitive Content Detection in Audio using Deep Learning Techniques

TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA DEL SOFTWARE
CURSO 2023–2024

Carlos Forriol Molina

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

Agradecer a mis directores y acompañantes Daniel Povedano Álvarez, Luis Javier García Villalba y a Ana Lucila Sandoval Orozco, por toda la ayuda que me han dado para realizar este trabajo.

Segundo, a mi familia y amigos en todo momento.

Finalmente, a todos los desarrolladores de las tantas herramientas que se han utilizado en este trabajo.

Índice General

Índice de Figuras

Índice de Tablas

Índice de Algoritmos

Lista de Acrónimos

AA	Aprendizaje Automático
ANS	Aprendizaje No Supervisado
AP	Aprendizaje Profundo
AS	Aprendizaje Supervisado
AUC	*Area Under the Curve*
AUC-ROC	*Area Under the Receiver Operating Characteristic Curve*
BERT	*Bidirectional Encoder Representations From Transformers*
Bi-GRU	Redes Neuronales Bidireccionales con Puertas Recurrentes
BiLSTM	*Bidirectional Long Short-Term Memory*
CNN	*Convolutional Neural Networks*
DPFTNet	*Dual-Path Fused Transformer Network*
FN	*False Negatives*
FNN	*Feedforward Neural Network*
FP	*False Positives*
GAN	*Generative Adversarial Network*

GFCC	*Gammatone Frequency Cepstrum Coefficient*
IA	Inteligencia Artificial
KNN	*K-Nearest Neighbors*
LSTM	*Long Short-Term Memory*
MAE	*Mean Absolute Error*
MFCCs	*Mel-Frequency Cepstral Coefficients*
MLP	*Multilayer Perceptron*
MSE	*Mean Squared Error*
RCSF	*Repeated Curve Shape Spectrum*
ReLU	*Rectified Linear Unit*
RNN	*Recurrent Neural Networks*
ROC	*Receiver Operating Characteristic*
STFT	*Short-time Fourier transform*
SVM	*Support Vector Machines*
TN	*True Negatives*
TP	*True Positives*

Abstract

Currently, the creation and consumption of content on the internet have increased, facilitating the spread of illegal material such as child pornography. An important challenge in combating this issue is that videos uploaded to the internet do not always show explicit nudity, making them difficult to detect using conventional methods like image analysis. Audio, often overlooked, can contain features that other models cannot detect. This work aims to develop a technology capable of identifying and classifying this type of content through audio analysis. The main motivation is to provide more effective tools to combat the spread of these materials.

The study focuses on the implementation of technologies to detect sensitive content in audio. Deep learning-based models were developed and evaluated to analyze audio fragments from videos. Despite the positive results, the model presents limitations such as the misclassification of water sounds and dependency on audio quality. A precision of over 90 % is achieved, but there is always room for improvement. Future work suggests a detailed analysis of model failures, the incorporation of temporality in sequence analysis, and experimentation with more advanced technologies. It also proposes improving the dataset to increase the model's robustness.

Keywords: Forensic Analysis, Content Detection, Deep Learning, Audio, Content Moderation, Child Protection, Artificial Intelligence.

Resumen

Actualmente, ha aumentado la creación y el consumo de contenido en internet, lo que ha facilitado la difusión de material ilegal como la pornografía infantil. Un desafío importante en la lucha contra este problema es que los vídeos subidos a internet no siempre muestran desnudez explícita, lo que dificulta su detección mediante métodos convencionales como el análisis de imágenes. El audio, a menudo pasado por alto, puede contener características que otros modelos no pueden detectar. Este trabajo trata de desarrollar una tecnología capaz de identificar y clasificar este tipo de contenido a través del análisis de audio. La principal motivación es proporcionar herramientas más efectivas para combatir la difusión de estos materiales.

El estudio se centra en la implementación de tecnologías para detectar contenido sensible en audio. Se desarrollaron y evaluaron modelos basados en aprendizaje profundo para analizar fragmentos de audio de vídeos. A pesar de los resultados positivos, el modelo presenta limitaciones como la mala clasificación de sonidos de agua y la dependencia de la calidad del audio. Se alcanza una precisión superior al 90 %, pero siempre hay margen de mejora. El trabajo futuro sugiere un análisis detallado de los fallos del modelo, la incorporación de la temporalidad en el análisis de secuencias, y la experimentación con tecnologías más avanzadas. También se propone mejorar el conjunto de datos para aumentar la robustez del modelo.

Palabras clave: Análisis Forense, Detección de Contenido, Aprendizaje Profundo, Audio, Moderación de Contenido, Protección Infantil, Inteligencia Artificial.

Capítulo 1

Introducción

1.1. Motivación

En la actualidad, el acceso a las nuevas tecnologías ha cambiado la forma en que se crea, comparte y consume contenido en internet. Esto ha traído beneficios, pero también ha facilitado la difusión de contenido ilegal, como la pornografía infantil. Algunos informes recientes destacan un aumento de este tipo de contenido.

La Fiscalía General del Estado ha expresado su preocupación sobre cómo se ha incrementado producción de pornografía infantil, lo que implica la urgencia de mejorar las herramientas tecnológicas para combatir este tipo de delitos [not23]. Además, el año 2023 ha registrado una cifra récord en la detección de abuso sexual infantil en España [elp].

Uno de los desafíos más complicados es que los vídeos de este tipo de contenido no siempre muestran desnudez explícita, lo que dificulta su detección mediante métodos basados en análisis de imágenes. El audio, que muchas veces se pasa por alto, puede contener pistas sobre el contenido del vídeo.

Este trabajo trata de desarrollar una tecnología que pueda clasificar contenido sensible a través del análisis de audio. La motivación es proporcionar herramientas más efectivas para combatir la difusión de estos materiales, garantizando un entorno digital más seguro para los menores.

1.2. Contexto

El presente Trabajo Fin de Grado se enmarca dentro de un proyecto de investigación titulado *Child protection centred strategies to fight against sexual abuse and exploitation– ALUNA*, aprobado por la Comisión Europea 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 GASS de la Universidad Complutense de Madrid (*Grupo de Análisis, Seguridad y Sistemas*, `https://gass.ucm.es`, grupo 910623 del catálogo de grupos de investigación reconocidos por la UCM)

1.3. Objeto de la Investigación

Este estudio se centra en el desarrollo y la implementación de tecnologías para la detección y clasificación de contenido sensible en pistas de audio. Normalmente, la detección de este tipo de contenido se centra en detectarlo en imágenes, pero el audio puede proporcionar otro enfoque, especialmente en casos donde el contenido visual puede ser difícil de detectar.

La investigación explora la aplicación de técnicas de AP para clasificar fragmentos de audio extraídos de vídeos, buscando patrones o características que puedan detectar este contenido.

El desarrollo de estos modelos no solo busca reducir la cantidad de contenido dañino en línea, sino también mejorar la automatización de detección de este tipo de contenidos.

1.4. Plan de Trabajo

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

1. **Investigación**: La primera parte del trabajo fue una fase de introducción y aprendizaje sobre el tema del proyecto, estableciendo las bases para poder comenzar con el desarrollo. Al principio se repasaron algunos conceptos básicos y se investigaron diferentes formas para empezar a adquirir estos conocimientos. Por ejemplo *Google Scholar* se puede usar para buscar artículos científicos sobre investigaciones de este campo. También se visualizó parte de un curso de AP [NG], tras ese periodo de investigación, se comenzó con el desarrollo.

2. **Desarrollo**: Tras obtener la base teórica necesaria para el desarrollo del proyecto, se redujo la fase de investigación para dar paso al desarrollo. La investigación continuó, pero de manera más orientada a resolver cuestiones que surgían durante el desarrollo. En esta etapa, se intentó dominar mejor el lenguaje de programación *Python* y adaptarse al librerías como *Keras* o *TensorFlow*, que eran útiles para el desarrollo.

3. **Experimentación**: Durante la etapa de experimentación, tras el desarrollo de los primeros modelos, se iniciaron pruebas utilizando algunas de las bibliotecas mencionadas anteriormente. En este período, se realizó una comparativa entre los resultados obtenidos para realizar ajustes en los parámetros del modelo. Se implementaron técnicas de aumento de datos para evaluar y mejorar el modelo. Importante destacar que el desarrollo continuó a la vez que con la experimentación.

1.5. Estructura del Trabajo

El trabajo está organizado en 6 capítulos y 4 anexos con la siguiente estructura: El Capítulo 2 introduce algunos conceptos en el contexto del análisis de audio mediante técnicas de aprendizaje profundo, enfocado en la identificación y clasificación de contenido sensible.

El Capítulo 3 ofrece una revisión del estado del arte en la identificación de contenido sensible en audio y en vídeos mediante técnicas avanzadas de aprendizaje profundo y procesamiento de señales.

El Capítulo 4 describe la metodología para desarrollar y evaluar un modelo como el que se necesita en este trabajo. Se mencionan tecnologías clave, el conjunto de datos utilizado para el entrenamiento del modelo y en general la arquitectura del modelo. Concluye describiendo el flujo de trabajo que incluye etapas de entrada, preprocesamiento, aumento de datos, entrenamiento, evaluación de modelos y la implementación de una interfaz de usuario para pruebas de rendimiento en tiempo real.

En el Capítulo 5 se describen todos los experimentos que se han realizado con sus resultados.

El Capítulo 7 son conclusiones que se han conseguido a lo largo del trabajo, se mencionan limitaciones y posibles ideas que se puedan llegar a desarrollar en un futuro.

Los Capítulos 8 y 6 son las traducciones al inglés de la Introducción y de las Conclusiones.

Capítulo 2

Capítulo Contexto de la Investigación

Este capítulo aborda los conceptos teóricos de este trabajo, comenzando con una revisión de los fundamentos de la IA, hasta una revisión de los elementos esenciales para el análisis y procesamiento de audio.

2.1. Introducción a la Inteligencia Artificial

La IA es un campo de la ciencia que se enfoca en crear sistemas que realizan tareas que requieren inteligencia humana. Estas tareas incluyen, por ejemplo, aprendizaje, razonamiento, reconocimiento de voz y toma de decisiones. Los avances este campo han llevado al desarrollo de conceptos como el AA y el AP. Estos avances han transformado la forma en la que las máquinas pueden aprender, interpretar datos y tomar decisiones. [Int]. La Figura 2.1 muestra de manera gráfica como se engloban los diferentes campos de la inteligencia artificial

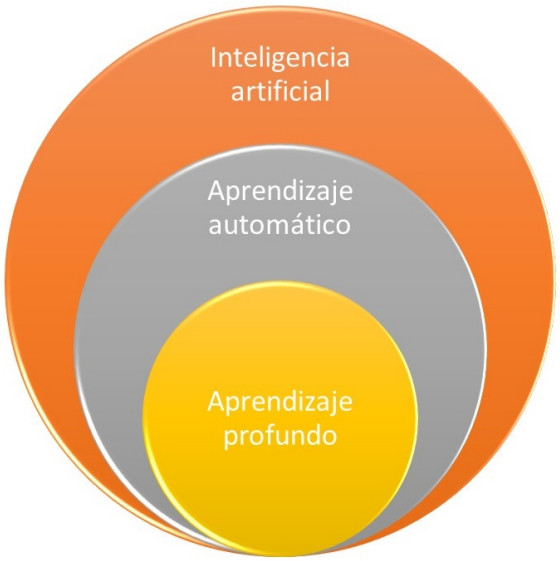

Figura 2.1: IA, AA y AP [Piq20]

2.2. Aprendizaje Profundo

En la última década, el mundo del AP ha experimentado una transformación en el ámbito de la IA, el cual se inspira en el cerebro humano y utiliza arquitecturas neuronales para extraer patrones y representaciones de conjuntos de datos. Esto ha permitido que las máquinas aprendan de forma automática, adaptándose a los datos y superando problemas que antes se consideraban insuperables.

2.2.0.1. Diferencia ente el Aprendizaje Profundo y Automático

El AP se basa en el uso de arquitecturas neuronales, compuestas por múltiples capas, para poder procesar datos. El AP ha demostrado su eficacia en áreas como reconocimiento de voz, traducción automática, conducción autónoma o diagnóstico médico. En cambio, el AA se basa en modelos más simples y depende en gran medida de la selección de características. El AA puede ser efectivo en tareas con grandes volúmenes de datos pero requiere un enfoque más manual en la extracción de características [DR21]. Como se puede ver en la Figura 2.2, a la hora de trabajar sobre un modelo basado en AA hay que centrarse más en la extracción de datos mientras que el AP se centra más en las capas de entrenamiento.

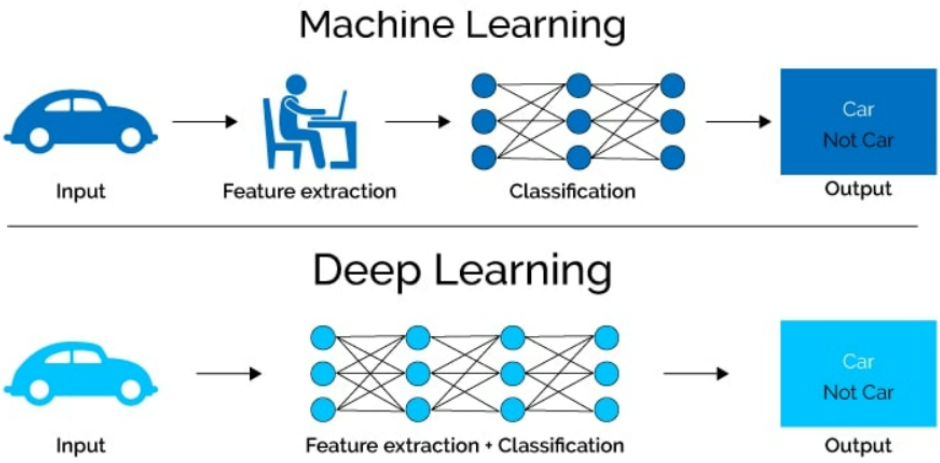

Figura 2.2: Diferencia entre AP y el AA [Sru24]

2.2.1. Redes Neuronales

Las redes neuronales son un elemento importante en el campo de la IA, están inspiradas en la estructura del cerebro humano, estas redes utilizan nodos conectados, o neuronas, para procesar información (Figura 2.3). Cada conexión entre neuronas tiene un "peso" que se ajusta durante el entrenamiento del modelo, lo que permite a la red aprender y adaptarse a patrones. Esto es efectivo en tareas como reconocimiento de patrones, procesamiento de imágenes y voz, contribuyendo al avance en la capacidad de las máquinas para comprender y analizar información de manera más sofisticada [LBH15].

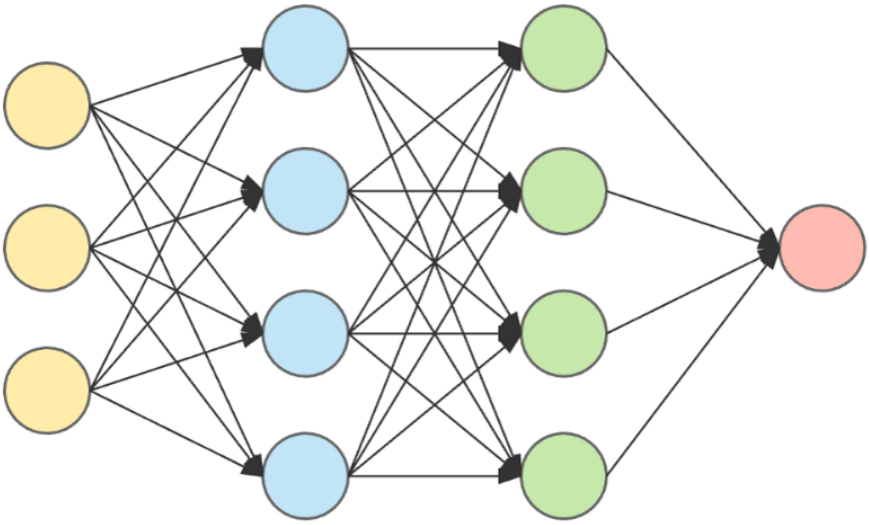

Figura 2.3: Entrada de datos, dos capas de la red y salida de datos de una red neuronal [cap]

2.2.1.1. Tipos de Redes Neuronales

Existen diversos tipos de redes neuronales, cada una diseñada para abordar diferentes desafíos. Las redes neuronales de propagación directa, del inglés (*Feedforward Neural Network* (FNN)) se tratan de una estructura con la información fluyendo en una dirección, desde la capa de entrada hasta la de salida, sin retroalimentación. Son ideales para tareas de clasificación simples.

Las Redes Neuronales Convolucionales, del inglés (*Convolutional Neural Networks* (CNN)), son un tipo de arquitectura de redes neuronales diseñadas para procesar datos con estructura de cuadrícula, como imágenes. La operación fundamental en una CNN es la convolución, que es una operación que implica deslizar un pequeño filtro llamado kernel sobre la entrada por ejemplo, una imagen, para realizar operaciones. Cada filtro detecta diferentes patrones, como bordes, texturas o características más complejas. Después de esta convolución, se aplica una función de activación como la Unidad Lineal Rectificada, del inglés (*Rectified Linear Unit* (ReLU)) para introducir no linealidades, es decir, si hay algún valor negativo, se convierte a 0, por lo que se introduce no linealidad en el modelo y permite a la red aprender representaciones más complejas y relevantes de los datos de entrada. Después, es común agregar capas de *pooling* para conservar las características más importantes. La operación de *pooling* implica tomar el valor máximo o promedio dentro de una región, reduciendo así la cantidad de información mientras conserva las características relevantes (Figura 2.4). Después de varias capas convolucionales y de *pooling*, se suelen agregar capas conectadas para realizar la clasificación final [cnna] [cnnb].

Las redes neuronales recurrentes, del inglés *Recurrent Neural Networks* (RNN) son perfectas para datos secuenciales, ya que tienen conexiones que les permiten recordar estados anteriores. Esto las hace valiosas para tareas como traducción automática o lenguaje natural. Las redes neuronales generativas, del inglés (*Generative Adversarial Network* (GAN)), en cambio, se emplean para crear datos nuevos que son prácticamente indistinguibles de los datos originales de entrenamiento. Estas redes son fundamentales en la generación de imágenes. Cada tipo de red neural ofrece diferentes tipos de fortalezas,

lo que permite abordar una amplia gama de problemas en el campo del AP [tip23].

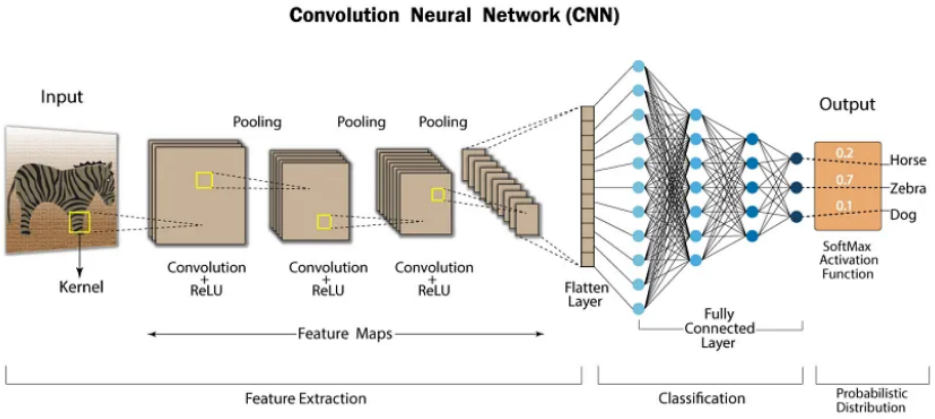

Figura 2.4: Proceso de extracción de características de una imagen [Haq]

2.3. Regresión y Clasificación

Las redes neuronales son herramientas que se aplican a una variedad de problemas, incluyendo regresión y clasificación. En la regresión, se entrenan para predecir valores numéricos continuos, como el precio de una casa. La salida se adapta para permitir que el modelo genere predicciones de datos nuevos. Por otro lado, en la clasificación, las redes neuronales son eficaces en clasificar diferentes categorías a datos, como en la clasificación de imágenes. En la Figura 2.5 se puede ver que mientras en la clasificación se trata de separar las instancias de datos en diferentes clases (en este caso 2) utilizando un límite de decisión, en la regresión se trata de ajustar una línea a los datos para predecir valores numéricos [Jen19].

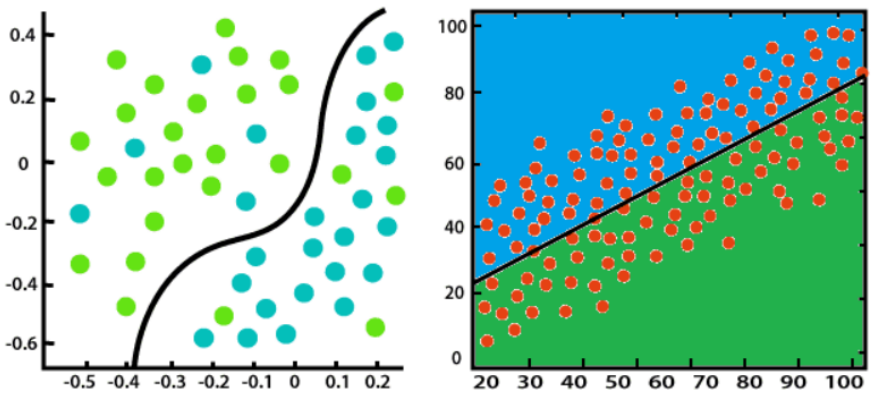

Figura 2.5: Diferencias de una función de clasificación y una de regresión [reg]

2.4. Aprendizaje Supervisado y No Supervisado

En el AS, los modelos se entrenan utilizando conjuntos de datos etiquetados, donde cada dato tiene una entrada emparejada con su correspondiente salida. El objetivo es aprender la relación entre las entradas y las salidas para más tarde hacer predicciones en datos no conocidos por el modelo. Por otro lado, en el ANS, los modelos no tienen información sobre las salidas deseadas de los datos. Técnicas como el *clustering* o la reducción de dimensionalidad son comunes en el ANS[SSK20] .En la Figura 2.6 se ve claramente la diferencia en la entrada de los datos, donde en el no supervisado el modelo no va a saber las etiquetas de ellos, sino que extraerá sus características

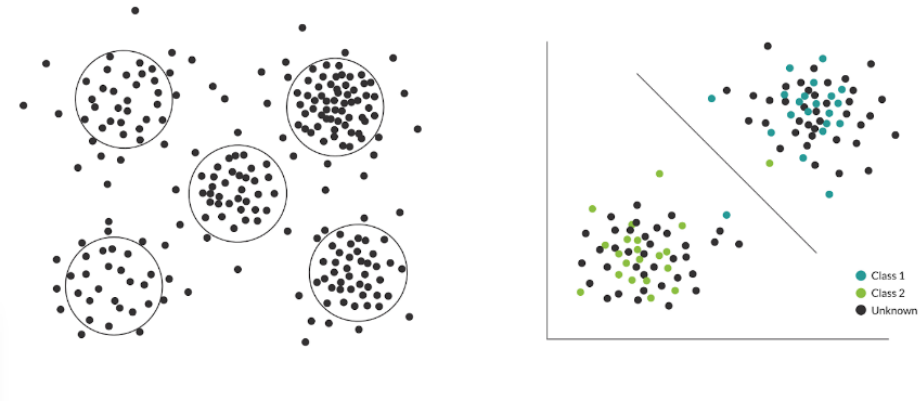

Figura 2.6: AS y ANS [Edw]

2.5. Estrategia de Partición del Conjunto de Datos

Para entrenar un modelo utilizando las diferentes arquitecturas mencionadas anteriormente, la preparación del conjunto de datos es importante. En general, el conjunto de datos se divide en tres partes para llevar a cabo el proceso de entrenamiento, como se puede ver en la Figura 2.7. La primera parte es el conjunto de entrenamiento, que se utiliza para entrenar el modelo, para que aprenda patrones y relaciones entre las entradas y salidas. La segunda es el conjunto de validación, empleado durante el entrenamiento para ajustar los parámetros del modelo y prevenir el sobreajuste, que ocurre cuando el modelo se ajusta demasiado bien a los datos de entrenamiento, resultando en que cuando le llegan datos nuevos, pierde eficacia. Finalmente, el tercer componente es el conjunto de prueba, que se usa para realizar una evaluación del rendimiento del modelo [Gre22]. Esta división del conjunto de datos facilita el desarrollo de modelos precisos.

2.6. Evaluación de Modelos

En cuanto a la evaluación de estos modelos, implica el uso de diversas métricas para medir su rendimiento. La exactitud cuantifica el porcentaje de predicciones correctas, mientras que precisión y *recall* son útiles en problemas de clasificación desbalanceada. El *F1-Score* [Roh22] combina precisión y *recall* para un equilibrio. La matriz de confusión ofrece una visión detallada del rendimiento. En clasificación binaria, la curva

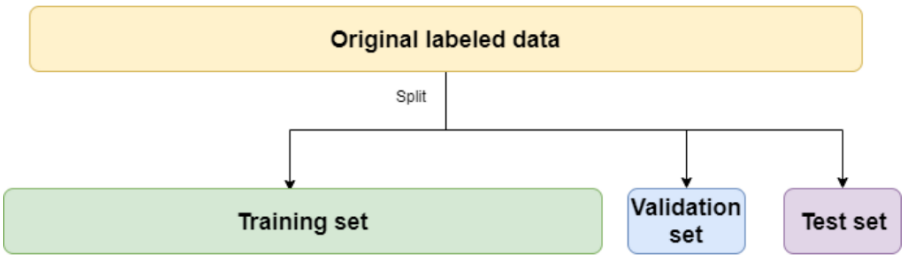

Figura 2.7: Cómo se suele dividir un conjunto de datos[Kea95]

Características Operativas del Receptor, del inglés (ROC) y el Área Bajo la Curva de Características Operativas del Receptor,del inglés (*Area Under the Receiver Operating Characteristic Curve* (AUC-ROC)) evalúan la capacidad de discriminación del modelo. Para problemas de regresión, se emplean el Error Cuadrático Medio, del inglés (*Mean Squared Error* (MSE)) y el Error Absoluto Medio, del inglés(*Mean Absolute Error* (MAE)) para medir el error entre predicciones y valores reales, cada una de estas métricas están explicadas a continuación. La elección de métricas depende del tipo de problema y los objetivos del modelo, siendo esencial considerar múltiples métricas para una evaluación completa del modelo [Roh22][met].

A continuación se explican algunas métricas más:

- **Verdaderos Positivos, del inglés (*True Positives* (TP))**: Indica el número de instancias que fueron correctamente identificadas como positivas por el modelo.
- **Verdaderos Negativos, del inglés (*True Negatives* (TN))**: Representa el número de instancias que fueron correctamente identificadas como negativas por el modelo.
- **Falsos Positivos, del inglés (*False Positives* (FP))**: Ocurren cuando el modelo predice incorrectamente una instancia como positiva siendo en realidad negativa.
- **Falsos Negativos, del inglés (*False Negatives* (FN))**: Ocurren cuando el modelo predice incorrectamente una instancia como negativa siendo en realidad positiva.

- **Exactitud (*Accuracy*):** Indica el porcentaje de predicciones correctas del modelo, se mide mediante esta fórmula[met]:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- **Precisión (*Precision*) y *Recall*:** Útiles en clasificación desbalanceada, miden la calidad de los TP y la capacidad del modelo para identificar todas las instancias positivas, respectivamente[Roh22].

$$\textit{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\textit{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- ***F1-Score*:** Combina precisión y *recall* para un equilibrio.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Matriz de Confusión:** Ofrece una visión detallada del rendimiento del modelo[Roh22] de la cantidad de verdaderos positivos y negativos y falsos positivos y negativos. (Figura 2.8).
- **Curva ROC y AUC-ROC:** Evalúan la capacidad de discriminación del modelo en clasificación binaria [met]. En la Figura 2.9 se muestra un ejemplo de la curva ROC.
- **Error Cuadrático Medio (MSE) y Error Absoluto Medio (MAE):** Medidas de discrepancia en problemas de regresión [met].

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right|$$

Figura 2.8: Matriz de confusión [Arc19]

2.7. Otras Estrategias en la Evaluación de Modelos

Otras estrategias en la evaluación de modelos en AP incluyen la técnica de validación cruzada (*cross-validation*). Esta técnica ayuda a evaluar la capacidad del modelo al dividir el conjunto de datos en múltiples subconjuntos. En cada iteración, se utiliza un subconjunto diferente como conjunto de prueba y el resto como conjunto de entrenamiento. Esto proporciona una evaluación más robusta del rendimiento del modelo [Tea24].

Para garantizar la generalización del modelo, se emplea un proceso de validación cruzada de *k-folds*. Esto significa dividir el conjunto de datos en k subconjuntos mutuamente excluyentes, entrenar el modelo k veces utilizando diferentes combinaciones de datos de entrenamiento y validación, y luego promediar los resultados de rendimiento obtenidos en cada iteración.

El aumento de datos es otra técnica valiosa. Consiste en aplicar transformaciones como rotaciones, zoom o cambios de brillo a las muestras de entrenamiento, lo que aumenta la diversidad del conjunto de datos y mejora la capacidad del modelo para generalizar a nuevas entradas de datos.

Al analizar los FP y los FN, se obtiene una comprensión más profunda del rendimiento del modelo. Permiten ajustar el umbral de decisión del modelo para abordar desequilibrios en la clasificación [set20].

El umbral óptimo es un valor que decide la asignación de instancias a clases según las predicciones de probabilidad del modelo. Ajustar este umbral es importante para equilibrar las tasas de FP y FN, especialmente en casos de desequilibrios. La curva ROC ayuda a evaluar el rendimiento del modelo en distintos umbrales, mostrando la relación entre la sensibilidad y la especificidad. El umbral óptimo depende de los requisitos del problema y puede ajustarse para optimizar el rendimiento del modelo.

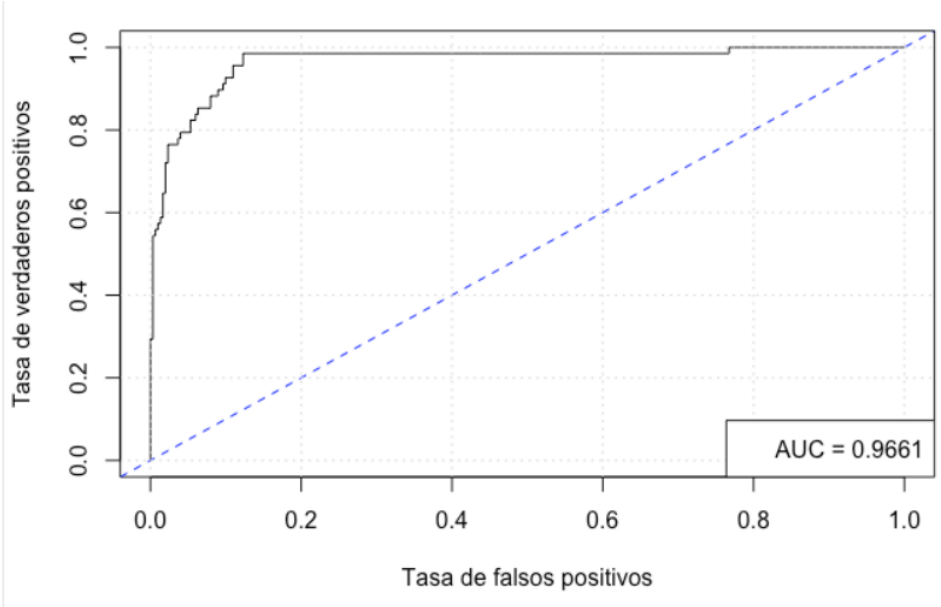

Figura 2.9: Gráfica de una ROC para un modelo de regresión [Val]

2.8. Análisis de Audio

Un audio es generado por variaciones en la presión del aire y puede representarse midiendo la intensidad de estas variaciones en función del tiempo. Los sonidos a menudo siguen patrones periódicos, pero pueden ser complejos al combinar señales de diferentes frecuencias (Figura 2.10).

2.8.1. Digitalización del Audio

Para digitalizar el audio, se convierte en una serie de números midiendo su amplitud en intervalos de tiempo llamados muestras. La preparación de datos para modelos de AP solía depender de técnicas tradicionales, pero con el paso del tiempo, se pueden convertir los datos de segmentos de audio en imágenes y utilizar arquitecturas como las mencionadas anteriormente [BT20].

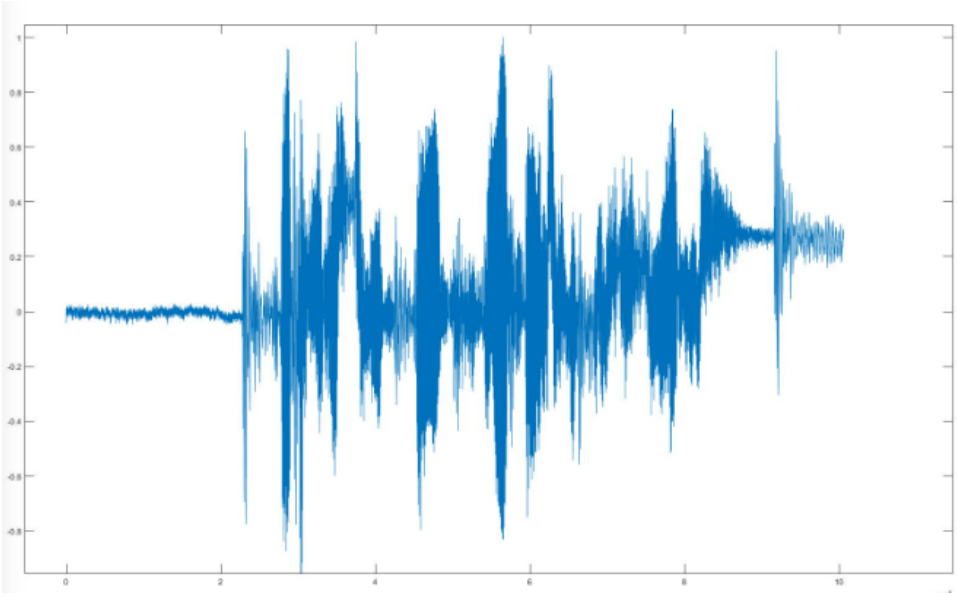

Figura 2.10: Señal de audio [int22].

2.8.2. Espectrogramas

Los sonidos se pueden representar mediante espectrogramas (Fig 2.11), que son fotografías del espectro de frecuencias en un tiempo determinado. Para construirlos, se utilizan transformadas de Fourier, que descomponen la señal en sus componentes de frecuencia.

Los parámetros importantes son:

- **Número de mels**: Número de bandas en la escala de Mel. Una mayor cantidad proporciona una representación más detallada.
- **Número de fft**: Tamaño de la ventana para la Transformada de Fourier. Un valor mayor mejora la resolución en frecuencia.

- ***Hop length***: Número de muestras entre los inicios de ventanas consecutivas. Un valor menor mejora la resolución.

Los modelos de AP para audio convierten datos de audio en espectrogramas, los procesan con CNNs y generan predicciones, abordando problemas como clasificación, separación o segmentación de audio [BT20].

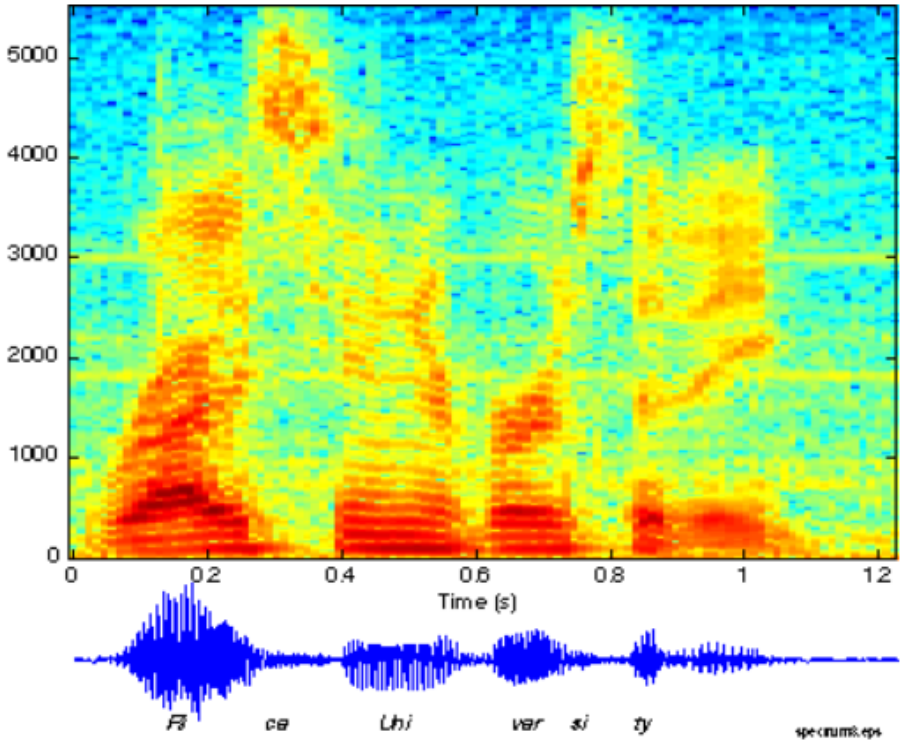

Figura 2.11: Espectrograma de una señal de voz [int17]

2.8.3. Espectrogramas de Mel

Los espectrogramas de Mel son un tipo de espectrogramas, basados en el concepto de la escala mel. La escala mel es una escala de tono que se basa en la respuesta del oído humano a diferentes frecuencias. A diferencia de la escala convencional, la escala mel refleja de manera más precisa cómo percibimos las diferencias de tono. Esta escala se utiliza para "mapear" las frecuencias del espectro de audio en intervalos mel. La relación entre la frecuencia en Hz (f) y la frecuencia en mel (m) se puede expresar mediante la siguiente fórmula:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.1}$$

El resultado final es una representación visual que destaca las características fundamentales del sonido (Figura 2.12). Esta técnica es valiosa para un modelo en una clasificación de audio, ya que captura de manera efectiva las características distintivas de diferentes fuentes sonoras [Ket21][JPA23].

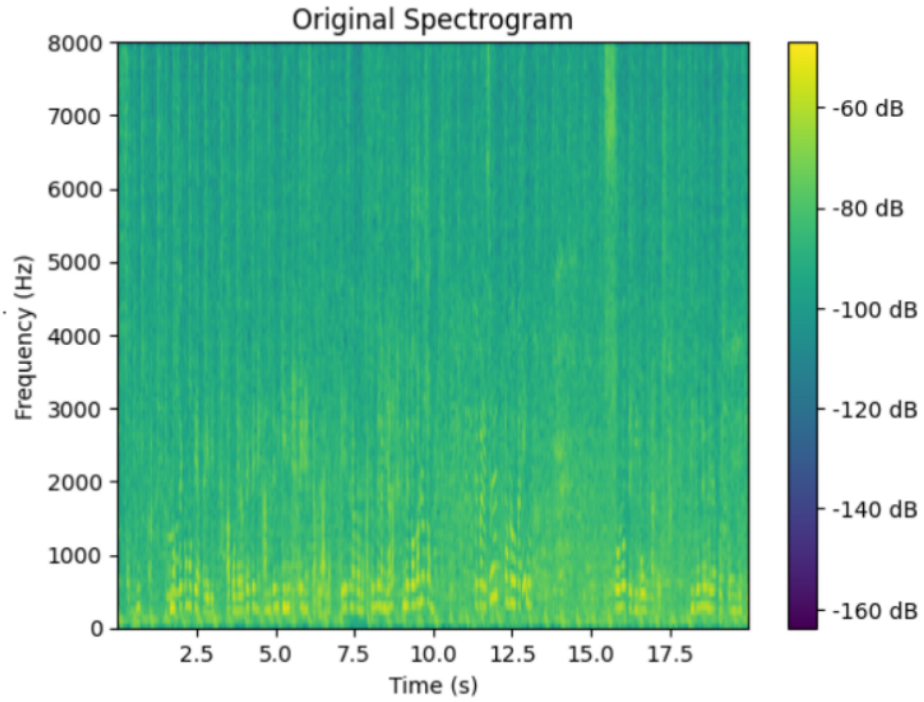

Figura 2.12: Espectrograma de Mel

2.8.3.1. Diferencia entre Espectrogramas de Mel y Log Mel

La diferencia principal entre los espectrogramas de Mel y los espectrogramas de Log Mel (Figura 2.13) es cómo se representan las magnitudes de las frecuencias. Mientras que los espectrogramas de Mel calculan la energía en bandas de frecuencia Mel, en los espectrogramas de Log Mel se aplica una transformación logarítmica, lo que comprime la información y resalta las diferencias perceptuales en el audio [Lel20].

La transformación logarítmica se puede ver en la siguiente fórmula:

$$\text{Log-Mel}(m) = \log(1 + \text{Mel}(m)) \tag{2.2}$$

Esta transformación logarítmica hace que los espectrogramas de Log Mel sean más efectivos para capturar características tonales esenciales y reducir la redundancia de datos, lo que los hace ampliamente utilizados en tareas de procesamiento de audio y análisis de sonido [Ket21].

2.8.3.2. Coeficientes Cepstrales de Frecuencia Mel

Los Coeficientes Cepstrales de Frecuencia Mel, del inglés (MFCCs) [MBE10] son derivados de la escala mel, estos coeficientes capturan las características espectrales del sonido de manera eficiente para su uso en modelos de AP. Los MFCCs se obtienen a través de un proceso que implica la aplicación de la transformada de coseno discreta. Estos coeficientes proporcionan información sobre las características más relevantes del espectro de audio, eliminando redundancias y destacando aspectos fundamentales. La utilización de MFCCs es otro enfoque para realizar tareas AP con audio, ya que ofrecen

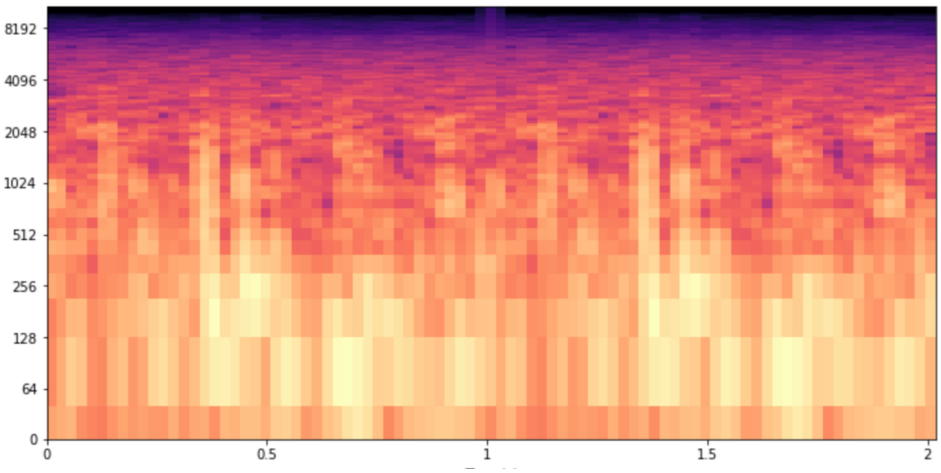

Figura 2.13: Espectrograma de Log Mel [MBE10]

una representación compacta de las características tonales de la señal de audio a lo largo del tiempo (Figura 2.14).

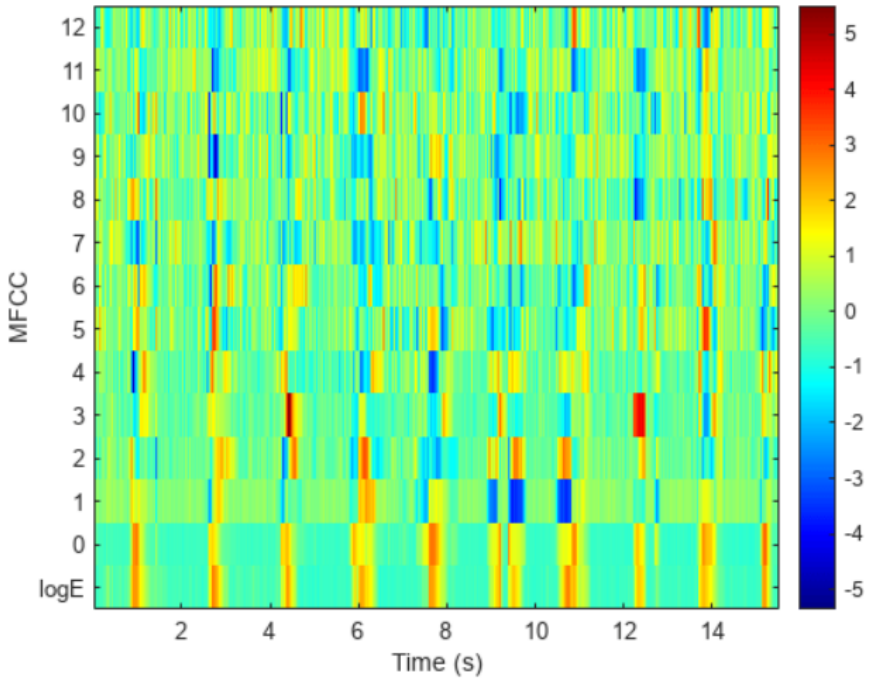

Figura 2.14: Imagen de un MFCCs

Capítulo 3

Estado del Arte

La identificación de contenido sensible en audio representa un desafío considerable en la actualidad, requiriendo el empleo de técnicas de AP especializadas. Esta revisión del estado del arte tiene como objetivo profundizar en las investigaciones más recientes sobre este tema, enfocándose en la utilización de metodologías de AP para abordar el trabajo.

3.1. Objetivos de la Revisión

- **Exploración de Técnicas Emergentes:** Investigar las arquitecturas más avanzadas de aprendizaje profundo aplicadas a la detección de contenido sensible en audio.
- **Evaluación de Desarrollos Recientes:** Examinar los avances más actuales en la detección de contenido sensible, en audio o vídeo, resaltando progresos tanto tecnológicos como metodológicos.
- **Identificación de Desafíos Actuales:** Discutir los problemas continuos y las limitaciones de las técnicas actuales utilizadas para detectar contenido sensible en audio.

A través de este análisis, se busca no solo entender el panorama actual de la investigación sino fomentar la investigación y el desarrollo futuros en este campo.

3.2. Técnica de Combinación de Espectrogramas

Lovenia et al. [LLF22] realizaron la detección de contenido pornográfico en audio utilizando (MFCCs) y espectrogramas log mel y mediante la extracción de información de un conjunto de datos llamado *Pornography-800* [ATC+13], que consta de 400 vídeos pornográficos y 400 no pornográficos. Emplearon arquitecturas de redes neuronales tanto de FNN como de CNN, siendo el modelo CNN, especialmente con espectrogramas Log Mel y una duración de segmento de 60 segundos en vez de 20, este último demostró un rendimiento superior en el *F1-score*. Se exploró varios métodos de predicción de segmento a audio, destacando la eficacia del método de votación. Los resultados se pueden ver en la Tabla 3.1.

Tabla 3.1: Comparación de *F1-score* por modelo y tipo de característica

Modelo	**Conjunto**	**Validación *F1-Score***	**Test *F1-Score***
FFNN	mfcc-20	89,56 %	90,25 %
FFNN	mfcc-60	91,89 %	90,99 %
FFNN	log-mel-20	92,88 %	92,49 %
FFNN	log-mel-60	93,72 %	93,42 %
CNN	mfcc-20	93,38 %	92,46 %
CNN	mfcc-60	93,46 %	93,92 %
CNN	log-mel-20	95,73 %	93,70 %
CNN	log-mel-60	95,38 %	94,89 %

3.3. Técnica Empleando Random Forest

Liu et al. [LLLZ23] proponen un método para la detección de contenido pornográfico en audio. Introducen una característica complementaria que combina espectrogramas de log mel, MFCCs y Coeficiente Cepstral de Frecuencia *Gammatone*, del inglés(*Gammatone Frequency Cepstrum Coefficient* (GFCC)). Su enfoque utiliza la arquitectura Red de Transformadores de Fusión de Doble Ruta, del inglés(*Dual-Path Fused Transformer Network* (DPFTNet)) y Random Forest[Bre01] para la clasificación.DPFTNet aborda la complejidad computacional mediante procesamiento en paralelo e incorpora *Batch Normalization* y *Patch Embedding*. El método logra un rendimiento superior, aprovechando un conjunto de datos de 13.338 muestras de audio. Las métricas de evaluación incluyen precisión y *F1-score*, demostrando su eficacia. El trabajo futuro tiene como objetivo mejorar el modelo para entornos ruidosos y la detección en tiempo real,enfocándose también en la optimización de la velocidad. Ver Tabla 3.2

Tabla 3.2: Comparación de resultados por métricas empleando Random Forest

Método	**Clasificación**	**Resultados**	
	Precisión	**Precisión**	**F1-score**
LMS	83,13 %	92,90 %	93,33 %
MFCC	72,78 %	89,05 %	89,64 %
GFCC	64,79 %	86,10 %	87,47 %
Modelo Propuesto	**84,02 %**	**93,20 %**	**93,56 %**

3.4. Técnica Empleando Refinamientos

Zhou et al. [ZWL+22] aborda la detección de contenido pornográfico en archivos de audio proponiendo un modelo basado en CNN y la investigación de refinamientos

como mecanismos de atención, métodos de *pooling*, *label smoothing*, *warmup* y *knowledge distillation*. En comparación con los otros trabajos el enfoque se distingue por su énfasis en refinamientos para la detección de audio, como mecanismos de atención o métodos de *pooling*. El artículo presenta una evaluación exhaustiva en un conjunto de datos abundante(224.127 audios pornográficos y 274.206 audios normales), logrando una precisión del 97,19 % gracias a la combinación de refinamientos. En la Tabla 3.3 se pueden observar los diferentes experimentos y resultados.

- ***Coordinate Attention* (CA)**: Mecanismo de atención que mejora la captura de características espaciales [HZF21].
- ***Label Smoothing* (LS)**: Técnica de regularización que suaviza las etiquetas para evitar sobreajuste [SVI+16].
- ***Max Pooling* (MP)**: Técnica que reduce la dimensionalidad seleccionando el valor máximo en una ventana de características.
- ***Gradual Warmup* (GW)**: Estrategia que incrementa gradualmente la tasa de aprendizaje al inicio del entrenamiento.
- ***Knowledge Distillation* (KD)**: Transferencia de conocimiento de un modelo grande a uno más pequeño para mejorar eficiencia [HVD+15].

Tabla 3.3: Resultados de precisión por método y características

CA	MP	LS	GW	KD	Precisión
					92,46 %
✓					95,63 %
	✓				93,64 %
		✓			92,98 %
			✓		93,26 %
				✓	93,66 %
✓	✓				96,25 %
✓	✓	✓			96,50 %
✓	✓	✓	✓		96,98 %
✓	✓	✓	✓	✓	97,19 %

3.5. Técnica Empleando Fusión Multinivel

Wang et al.[WZW+20] propone un método de fusión multinivel de características profundas multimodales para el reconocimiento de streamers pornográficos en vídeos en vivo. Se extraen características de texto mediante una red basada en Representaciones de Codificador Bidireccional de Transformadores, del ingés *Bidirectional Encoder*

Representations From Transformers (BERT), también se utiliza una arquitectura que integra CNN y Redes Neuronales Bidireccionales con Puertas Recurrentes (Bi-GRU) para procesar las características auditivas. Se realiza una serie de experimentos utilizando un conjunto de datos propio que, tras la limpieza de datos, contiene 2528 vídeos en vivo de streamers pornográficos para validar la eficacia del método propuesto.

3.6. Técnica Empleando MFCCs y KNN

Banaeeyan et al. [BAKL+19] aprovechan el audio para detectar contenido sensible en escenas oscuras utilizando técnicas como tonos y MFCCs así como el entrenamiento de modelos basados en K-Vecinos Más Cercanos,del inglés(KNN). Se utilizó un conjunto de datos de vídeo presentado por Lopes et al.[LAP+09], que incluía 179 muestras de vídeos. Los resultados realizados en un conjunto de datos creado a partir de un conjunto de vídeos pornográficos existente, demuestran la viabilidad del enfoque con una precisión del 88,40 % , un *F1-score* del 85,20 % y un *Area Under the Curve* (AUC) del 95 %. Se identificó que el KNN medio ofreció el mejor rendimiento en términos de *recall*, *F1-score* y AUC. No obstante, otros clasificadores también demostraron resultados comparables, destacando la importancia de considerar diversos enfoques en futuras investigaciones en esta área. En la Tabla 3.4, se puede observar la comparación de métricas para diferentes métodos KNN.

Tabla 3.4: Comparación de métricas para diferentes métodos KNN

Método	**Precisión**	**Recall**	**F1-Score**	**Accuracy**	**AUC**
Coarse KNN	72.10 %	84.76 %	77.92 %	83.00 %	92 %
Cosine KNN	85.02 %	83.44 %	84.22 %	86.70 %	94 %
Cubic KNN	83.04 %	85.55 %	84.27 %	87.10 %	94 %
Fine KNN	83.22 %	86.04 %	84.61 %	87.40 %	87 %
Medium KNN	84.32 %	86.09 %	85.20 %	87.80 %	95 %
Weighted KNN	83.68 %	85.79 %	84.72 %	88.40 %	87 %

3.7. Técnica Evaluación de RCSF

Lim et al. [LCHL11] proponen la característica de espectro en forma de curva repetida, del ingés: *Repeated Curve Shape Spectrum* (RCSF). El conjunto de datos incluye 6.269 clips de audio clasificados en obscenos y no obscenos, con 1.200 vídeos pornográficos y generales para evaluación. La (RCSF) alcanza un rendimiento destacado con un F1-score del 96,6 %, precisión del 98,2 %, y tasa de *recall* del 95,2 %, demostrando la eficacia de clasificar vídeos X exclusivamente con características de audio. La (RCSF) representa la variación temporal del espectro de frecuencia mediante una transformada de coseno discreta aplicada a MFCCs, evaluando la eficacia de diferentes órdenes de coeficientes para construir vectores de características RCSF.

3.8. Técnica de Optimización con EfficientNet y BiLSTM

Yousaf et al. [YN22] utiliza una arquitectura de CNN pre entrenada,*EfficientNet-B7* ,luego utilizan una red neuronal de Memoria Bidireccional a Largo y Corto Plazo *Bidirectional Long Short-Term Memory* (BiLSTM) para aprender representaciones efectivas de vídeo y realizar clasificación incorporando un mecanismo de atención para destacar la información más relevante. El conjunto de datos manualmente etiquetado consta de 111.156 clips de caricaturas de YouTube. Los resultados experimentales muestran que *EfficientNet-BiLSTM* sin atención supera a la versión con atención (95,66 % frente a 95,30 %).

3.9. Técnica de Detección Multimodal Empleando AudioVGG

Freitas et al. [AdFBGC20] propone un enfoque multimodal para la detección de contenido pornográfico en vídeos, centrándose específicamente en el análisis de audio. Se emplean CNN diseñadas para extraer características de secuencias de audio. El modelo consta de dos módulos principales: un extractor de características de audio y un clasificador. El modelo es el "AudioVGG" basado en la arquitectura VGG16 [SZ15], adaptado para secuencias de audio. Los modelos explorados fueron los Secuenciales: Memoria a Largo Corto Plazo, del inglés (*Long Short-Term Memory* (LSTM)) y No secuenciales: Máquinas de Soporte Vectorial, del inglés(*Support Vector Machines* (SVM)), KNN y Perceptrón Multicapa, del inglés(*Multilayer Perceptron* (MLP)). La base de datos consistió en*Pornography-2k* [MAP+16] combinada con una muestra de 1.976 vídeos pornográficos de la Base de Datos Xvideos. Dividido en 90 % entrenamiento, 5 % validación y 5 % prueba. Se evaluaron modelos con métricas como Precisión, *Recall* y *F1-score*, el modelo LSTM destacó con un *F1-score* medio del 99 % , superando otros métodos como SVM y KNN.

3.10. Conclusión

La revisión exhaustiva del estado del arte en la detección de contenido pornográfico en audio da una diversidad de enfoques y metodologías innovadoras. Cada artículo aborda el tema de diferentes formas, aprovechando avances en técnicas de procesamiento de señales de audio y modelos de aprendizaje profundo. Estos artículos proporcionan una panorámica completa de las estrategias actuales para abordar la detección de contenido pornográfico en audio.

Capítulo 4

Detección de Contenido Sensible en Audio

Este capítulo describe la metodología para desarrollar y evaluar un modelo de aprendizaje profundo que clasifique el contenido en categorías sensible y segura. La tarea requiere un enfoque detallado desde la selección y preparación de datos hasta la implementación del modelo y su evaluación en entornos reales. Se presentan etapas clave como la definición de objetivos, las tecnologías utilizadas, la preparación de datos, la arquitectura del modelo y el flujo de trabajo, desde la entrada hasta la evaluación.

4.1. Objetivos

Esta investigación se enfocó en establecer objetivos claros y medibles que guíen el desarrollo y la evaluación del modelo propuesto. Los objetivos específicos son:

- **Desarrollar un modelo de AP** que sea capaz de distinguir eficazmente entre contenido sensible y seguro en pistas de audio o en vídeos. Este modelo deberá alcanzar un umbral de precisión y sensibilidad específico, tras las primeras pruebas se estableció que un modelo de más de un 85 de precisión en la clasificación se consideraría existoso.

- **Optimizar el preprocesamiento de audio** para mejorar la calidad y la relevancia de las características extraídas que permitirán al modelo ser más eficiente. Esto incluye la exploración y aplicación de diferentes técnicas como la creación de espectrogramas o el ajuste de datos de audio, por ejemplo, rellenar el audio con silencios.

- **Implementar un sistema de validación cruzada** para asegurar la generalización del modelo. El modelo debe ser robusto y capaz de generalizar a nuevos datos no vistos durante el entrenamiento, lo que será evaluado a través de una validación cruzada de *k-folds* .

- **Evaluar el modelo con técnicas de aumento de datos:** El objetivo es implementar y comparar diferentes estrategias de aumento de datos, como la modificación de las muestras existentes. Estas técnicas de aumento de datos tienen

como fin equilibrar la representación de ambas clases en el entrenamiento del modelo, mejorando así su capacidad de generalización frente a nuevos datos.

- **Desarrollar una interfaz:** que permita probar el modelo.

4.2. Algunas Tecnologías Utilizadas

En el desarrollo de este trabajo, se ha empleado varias tecnologías y bibliotecas destacadas en el campo del procesamiento de datos y el aprendizaje profundo. Estas herramientas han sido fundamentales para llevar a cabo el análisis de audio, la construcción de modelos y el procesamiento de datos. A continuación, se detalla las más importantes:

- **Python**: El lenguaje de programación principal utilizado, es muy utilizado en la comunidad científica y de investigación por su extensa biblioteca de paquetes para ciencia de datos. En la Figura 4.1 se puede ver el logo del lenguaje. [VR91].

Figura 4.1: Logo de Python [pyt]

- **TensorFlow**: Proporciona un entorno para el entrenamiento de modelos de AP, permitiendo la manipulación eficiente de grandes conjuntos de datos y operaciones complejas sobre tensores (Figura 4.2) [ABC+15].

Figura 4.2: Logo de Tensorflow [tfl]

- **Keras**: Una biblioteca de alto nivel para el aprendizaje profundo, que funciona como interfaz para TensorFlow, facilitando la creación y el entrenamiento de modelos de

Figura 4.3: Logo de Keras [ker].

redes neuronales. Se utiliza para diseñar y experimentar con la arquitectura de las redes convolucionales utilizadas en este proyecto 4.3) [C+15].

- **Librosa**: Especializada en el análisis de música y audio, esta biblioteca es empleada para la extracción de características de audio, incluyendo la conversión de audio a espectrogramas y la extracción de MFCCs, que son esenciales para el análisis realizado (Figura 4.4) [MRL15].

Figura 4.4: Logo de Librosa [lib].

- **Pandas**: Utilizada para la manipulación y análisis de grandes conjuntos de datos, especialmente útil para manejar y procesar los datos y resultados obtenidos de los modelos (Figura 4.5) [McK10].

Figura 4.5: Logo de Pandas [pan].

- **Matplotlib**: Es una librería de visualización en Python para la creación de gráficos estáticos, animados e interactivos en la ciencia de datos. Permite visualizar de manera efectiva los resultados de análisis y modelos(Figura 4.6) [H+03].
- **Scikit-Learn**: Proporciona herramientas simples y eficientes para el análisis predictivo. Es fundamental para el preprocesamiento de datos, la selección de

Figura 4.6: Logo de Matplotlib [mat].

características, y la evaluación de modelos a través de técnicas como la validación cruzada (Figura4.7) [PVG11].

Figura 4.7: Logo de Scikit-Learn [gra].

- **Gradio**: Una herramienta capaz de crear aplicaciones web en Python. Se utiliza para hacer interfaces de usuario para los modelos (Figura 4.8) [stra].

Estas tecnologías, además de muchas otras, proporcionan una base sólida para el desarrollo de este trabajo.

4.3. Conjunto de Datos

Para el conjunto de datos se seleccionó *Pornography-2k* [MAP+16] , compuesto por 2000 vídeos, divididos equitativamente entre materiales de índole sensible y seguro, sumando en total 140 horas de contenido audiovisual.

Una parte significativa del contenido sensible proviene de redes de vídeos de uso general. Este repertorio incluye una gama de grabaciones que oscilan entre los seis segundos y los 33 minutos de duración.

Figura 4.8: Logo de Gradio [strb].

El conjunto de datos *Pornography-2k* se caracteriza por su diversidad, con varios géneros pornográficos, reflejando así un amplio espectro de comportamientos y etnias. Este enfoque asegura que el conjunto de datos no solo sea extenso en cantidad, sino también en variedad.

En cuanto al contenido no pornográfico del conjunto de datos *Pornography-2k*, consta de muestras aleatorias de vídeos de propósito general como tutoriales, vídeos musicales, personas hablando, etc.

4.4. Preprocesamiento

En cuanto al preprocesamiento de los vídeos de *Pornography-2k* [MAP+16], se extrajeron las pistas de audio directamente de los vídeos en segmentos de 20 segundos. Si se trataba del final de un vídeo, la duración podía ser menor, pero nunca excedía esos segundos. Las extracciones se realizaron a 44 kHz utilizando la biblioteca librosa. Cada audio fue etiquetado según su contenido: 0 para seguro y 1 para sensible.

Tras la extracción de los audios, se detectó que algunos tenían fallos, lo cual se debía a que algunos vídeos no tenían sonido y fallaban a la hora de entrenar el modelo, por lo que se eliminaron del conjunto de datos 23 vídeos. Estos estaban distribuidos de manera casi equitativa entre las dos categorías, por lo que su eliminación no tuvo un impacto en el trabajo.Del total de 2000 vídeos, se obtuvieron 26257 audios, de los cuales 7797 fueron seguros y 18460 sensibles, reflejando una mayor duración promedio de los vídeos de contenido sensible. En la Tabla 4.1 se pueden ver los contenidos de una forma más visual.

Tabla 4.1: Conjunto de datos

	Seguro	**Sensible**	**Total**
Videos	989	988	1.977
Audios	18.460	7.797	26.257

4.4.1. División de Audio desde una Carpeta de Vídeos

El algoritmo 1 toma como entrada una carpeta que contiene vídeos en los formatos `.mp4`, `.avi` o `.mkv`. Luego, para cada vídeo:

1. Crea una carpeta de salida específica para el nombre base del vídeo.
2. Divide el vídeo en segmentos de duración máxima de 20 segundos y extrae el audio de cada segmento.
3. Los segmentos de audio se guardan en formato `.wav` con nombres específicos que reflejan la duración y un índice para cada segmento.

La librería *Moviepy* [Zul14] se encarga de cargar el vídeo y obtener la duración, mientras que librosa permite procesar y extraer segmentos del audio.

4.4.2. División de audio desde un DataFrame

El algoritmo 2 trabaja con un `DataFrame` que contiene rutas de vídeo y etiquetas asociadas. El proceso sigue estos pasos:

1. Crea una carpeta de salida según la etiqueta de cada vídeo (`0` para seguro, `1` para sensible).
2. Extrae el audio en segmentos de 20 segundos (o menos si es el final del vídeo), siguiendo una estructura similar al primer algoritmo.

Se pueden ver mejor estos algoritmos más abajo.

4.5. Arquitectura

La arquitectura del sistema se fundamenta en el AP, utilizando principalmente CNN. Este tipo de red es eficiente en el procesamiento de datos visuales y auditivos, reflejado en la mayoría de los artículos revisados, que resaltan su uso en sistemas similares.

4.5.1. Selección de la Arquitectura

Se ha elegido implementar CNN por su capacidad para procesar datos de espectrogramas de audio, que al fin y al cabo son imágenes. Las CNN son particularmente eficientes para identificar patrones en estos datos, lo cual es importante para lograr una clasificación precisa.

4.5.2. Procesamiento de Entrada

La transformación de la señal de audio comienza con la aplicación de la Transformada de Fourier de corto tiempo (Figura 4.9) , del inglés (*Short-time Fourier transform* (STFT)) para convertir la señal temporal en una representación de frecuencia. A continuación, se extraen espectrogramas de log-mel, que proporcionan una representación más relevante del

Algoritmo 1: Proceso de división de audio desde una carpeta de vídeos

```
1:  procedure DividirAudioDesdeCarpeta(carpeta_videos, carpeta_salida,
    duracion_maxima)
2:      Verificar y Crear Carpeta: if carpeta_salida no existe then
3:
          Crear carpeta_salida
4:
5:      Procesar Todos los Videos: for cada archivo_video en carpeta_videos do
6:
          ruta_video = unir_ruta(carpeta_videos, archivo_video) if archivo_video no
    termina con {.mp4, .avi, .mkv} then
7:
          continuar                                  ▷ Ignorar archivos que no son videos
8:
9:      nombre_base = quitar_extension(archivo_video)
10:     carpeta_salida_video = unir_ruta(carpeta_salida, nombre_base)
          if carpeta_salida_video no existe then
11:
          Crear carpeta_salida_video
12:
13:     Obtener Duración del Video y Dividir en Segmentos:
14:     clip_video = cargar_video(ruta_video)
15:     duracion_video = obtener_duracion(clip_video)
          for i desde 0 hasta (duracion_video / duracion_maxima) + 1 do
16:
          tiempo_inicio = i * duracion_maxima
17:     tiempo_fin = min((i + 1) * duracion_maxima, duracion_video)
18:     duracion = redondear(tiempo_fin - tiempo_inicio)
19:     nombre_audio_salida = nombre_base _duracion_s_ i .wav
20:     ruta_audio_salida = unir_ruta(carpeta_salida_video, nombre_audio_salida)
21:     Extraer y Guardar Segmento de Audio:
22:     segmento_audio = obtener_segmento_audio(clip_video, tiempo_inicio, tiempo_fin)
23:     escribir_audio(segmento_audio, ruta_audio_salida, codec="pcm_s16le", fps=16000)
24:
25:
26: end procedure
```

Algoritmo 2: Proceso de división de audio desde un dataFrame

```
procedure DividirAudioDesdeDataFrame(df, carpeta_salida, duracion_maxima)
    Verificar y Crear Carpeta: if carpeta_salida no existe then
        Crear carpeta_salida
    Procesar Todos los Videos en el DataFrame: for cada fila en df do
        ruta_video = fila['ruta_video']
    etiqueta_video = fila['etiqueta_video']
    nombre_base = quitar_extension(ruta_video)
    Determinar Carpeta de Audio Según la Etiqueta: if etiqueta_video es 0 then
        carpeta_audio_salida = unir_ruta(carpeta_salida, 'aPorn')
    etiqueta_video es 1
    carpeta_audio_salida = unir_ruta(carpeta_salida, 'aNonPorn') else
        lanzar ValueError: {Etiqueta de video no válida}
        if carpeta_audio_salida no existe then
        Crear carpeta_audio_salida
Obtener Duración del Video y Dividir en Segmentos:
clip_video = cargar_video(ruta_video)
duracion_video = obtener_duracion(clip_video)
        for i desde 0 hasta (duracion_video / duracion_maxima) + 1 do
        tiempo_inicio = i * duracion_maxima
tiempo_fin = min((i + 1) * duracion_maxima, duracion_video)
duracion = redondear(tiempo_fin - tiempo_inicio)
nombre_audio_salida = nombre_base _duracion_s_ i .wav
ruta_audio_salida = unir_ruta(carpeta_audio_salida, nombre_audio_salida)
Extraer y Guardar Segmento de Audio:
segmento_audio = obtener_segmento_audio(clip_video, tiempo_inicio, tiempo_fin)
escribir_audio(segmento_audio, ruta_audio_salida, codec="pcm_s16le", fps=16000)


end procedure
```

audio al imitar la respuesta del oído humano. Tras los espectrogramas de log-mel se pueden obtener los MFCCs, que son características ampliamente reconocidas por su eficacia en la representación de sonidos para modelos.

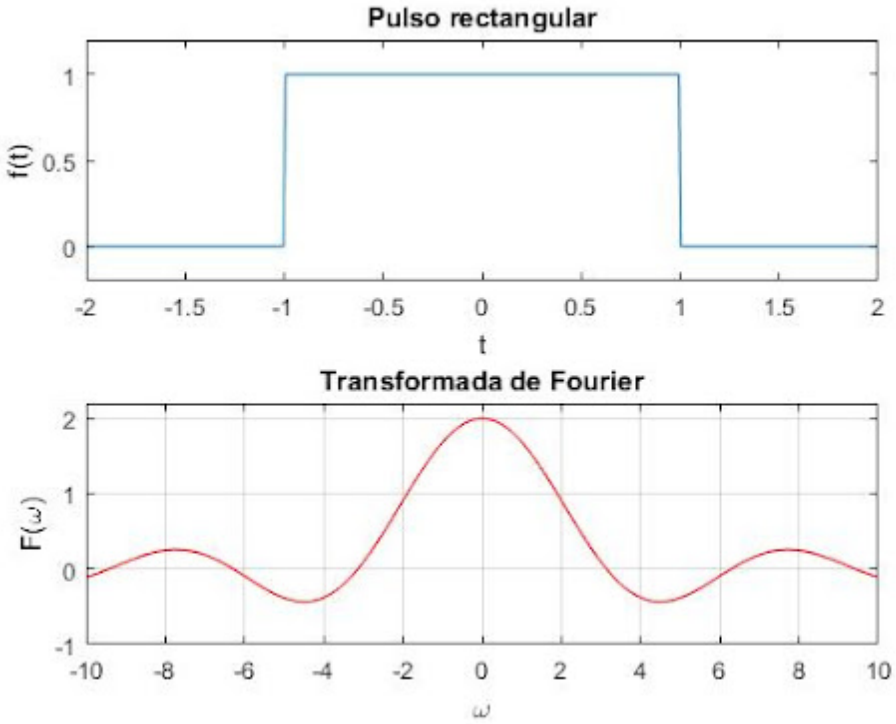

Figura 4.9: Diferencia entre transformada de Fourier y un pulso rectangular [sft].

4.5.3. Estructura de la Red

Se emplean múltiples capas convolucionales para detectar patrones en los espectrogramas. Estas capas están seguidas de capas de *pooling* que reducen la dimensionalidad de los datos, permitiendo un enfoque más generalizado y menos propenso al sobreajuste.

Capas de Normalización y Activación: Se utiliza *Batch Normalization*, que es una técnica para mejorar la velocidad, estabilidad y rendimiento del entrenamiento y se usan funciones de activación ReLU para mejorar la eficiencia del entrenamiento y la no linealidad de la red.

4.5.4. Salida y Clasificación

Después de las capas convolucionales, se incluyen una o varias capas para la clasificación final. La capa de salida utiliza una función de activación *softmax* para categorizar las entradas en contenido sensible o seguro.

4.5.5. Extracción del Umbral Óptimo

Para encontrar el umbral óptimo en la clasificación de contenido, se llevó a cabo el siguiente proceso:

1. **Conversión de Probabilidades a NumPy:** Convertir las probabilidades combinadas en un array de NumPy para facilitar la manipulación.
2. **Probabilidad de la Clase Positiva:** Extraer las probabilidades correspondientes a la clase positiva para cada predicción.
3. **Curva de Precisión:** Usando las etiquetas verdaderas y las probabilidades, calcular las curvas de precisión y recall, junto con los correspondientes umbrales.
4. **Cálculo del *F1-Score*:** Calcular el *F1-Score* para cada umbral, evaluando la relación entre precisión y *recall*.
5. **Selección del Mejor Umbral:** Identificar el umbral que maximiza el *F1-Score*, lo cual permitirá equilibrar precisión y *recall* para una clasificación óptima.

Este proceso proporciona un umbral óptimo para separar las clases, maximizando el rendimiento global del modelo (Figura 4.10).

4.5.6. Optimización

Se emplea el optimizador de Estimación de Momentos Adaptativos: *Adam* para el ajuste de los pesos de la red mediante tasas de aprendizaje adaptativas. El optimizador *Adam* es utilizado por su capacidad para ajustar automáticamente las tasas de aprendizaje durante el entrenamiento [KB14].

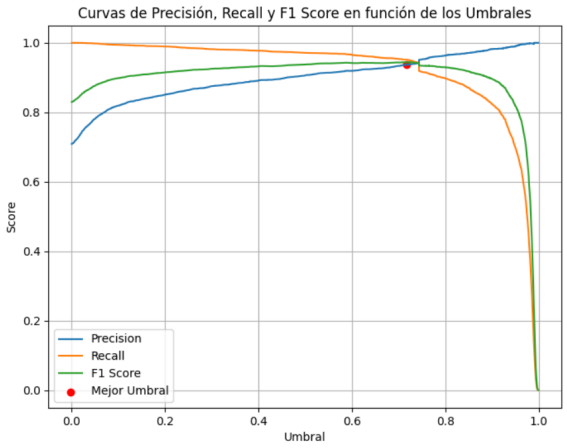

Figura 4.10: Umbral Óptimo.

La arquitectura basada en CNN demuestra eficiencia en la clasificación de datos de audio transformados a espectrogramas de log-mel y MFCCs. Las capas convolucionales capturan patrones, mientras que las capas de *pooling* y funciones de activación ReLU permiten una clasificación eficiente. El uso de *Adam* como optimizador proporciona un ajuste rápido de los pesos, permitiendo tasas de aprendizaje adaptativas para mejorar la precisión del modelo en la detección de contenido.

4.6. Flujo de Trabajo

El flujo de trabajo propuesto consta de varias etapas clave que se detallan a continuación:

1. **Entrada de la señal de audio:** El proceso inicia con la recepción de señales de audio, las cuales son extraídas de vídeos, como se ha mencionado anteriormente. Esta etapa importante puesto que es donde se recopila el material que más tarde será procesado y analizado por el modelo.

2. **Preprocesamiento de la señal:** Una vez cargada la señal de audio, comienza el preprocesamiento, el cual consiste en transformar la señal de audio en un espectrograma mediante STFT. Este paso es fundamental para convertir las señales de audio en una representación visual que refleje las frecuencias. Para mantener consistencia en el procesamiento, los espectrogramas se estandarizan a una duración de 20 segundos. Si un segmento de audio es menor a este tiempo, se realiza un relleno para asegurar que todos los espectrogramas tengan el mismo tamaño, facilitando así la clasificación.

3. **Aumento de datos:** Para mejorar la robustez del modelo, se implementan técnicas de aumento de datos. Incluyen la generación de muestras y la modificación de muestras existentes. Este proceso se realiza para equilibrar la representación de las clases (contenido seguro y sensible) dentro del conjunto de entrenamiento. El aumento de datos ayuda a que el modelo aprenda a identificar y clasificar correctamente el contenido bajo diversas variaciones, imitando así las condiciones que podría encontrar en aplicaciones del mundo real.

4. **Entrenamiento del modelo:** Tras completar el preprocesamiento, se procede al entrenamiento del modelo utilizando una arquitectura de CNN. Durante esta fase, el modelo aprende a identificar patrones distintivos en los espectrogramas. Se emplean métricas de rendimiento como precisión, sensibilidad y el propio aumento de datos para evaluar y ajustar el desempeño del modelo. Estas métricas son esenciales para asegurar que el modelo no solo se ajuste a los datos de entrenamiento, sino que también posea la capacidad de generalizar bien a nuevos datos no vistos.

5. **Evaluación de Modelos:**

   a) **Comparación de Modelos:**
   - Se cargan y evalúan dos modelos con diferentes características: MFCCs y espectrogramas de Log Mel, utilizando un mismo conjunto de datos de prueba.
   - Se comparan las métricas de cada modelo para identificar fortalezas y debilidades de cada uno.

   b) **Probabilidades Combinadas:**
   - Se combinan las probabilidades de los modelos promediándolas para suavizar las predicciones.
   - Se recalculan las métricas (precisión, *recall* , *F1-score*) para evaluar el rendimiento de la combinación.

   c) **Análisis de FP y FN:**

- Se identifican los FP y FN.
- Se estudian patrones comunes que causan errores para optimizar el umbral de decisión.
- Se ajustan los umbrales de decisión basados en las métricas de rendimiento para optimizar el balance entre precisión y *recall*.

6. **Interfaz de usuario para pruebas de rendimiento:** Finalmente, el modelo entrenado se implementa en una interfaz de usuario desarrollada con Gradio. Esta interfaz permite a los usuarios cargar nuevos archivos de audio y obtener resultados de clasificación en tiempo real.

   Para facilitar el acceso y la utilización del sistema propuesto, se ha desarrollado una interfaz de usuario intuitiva utilizando la plataforma Streamlit. Esta interfaz proporciona una experiencia amigable que permite a usuarios cargar fácilmente listas de audios o vídeos directamente desde su dispositivo.

   Una vez que los archivos multimedia son cargados en la interfaz, el sistema los procesa utilizando el modelo integrado. Este modelo analiza los datos de entrada para identificar patrones asociados con contenido sensible.

   Los resultados obtenidos del análisis son presentados de manera clara y concisa en la misma interfaz de usuario. Información sobre si se detectó contenido sensible en los archivos cargados Figura 4.11.

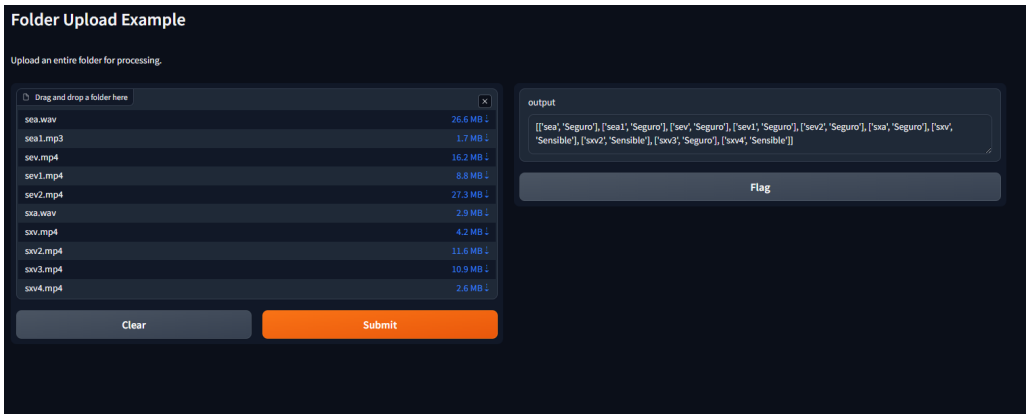

Figura 4.11: Interfaz para realizar pruebas.

En la Figura 4.12 se puede ver el flujo de trabajo descrito de manera simplificada, pero puede ayudar a entenderlo.

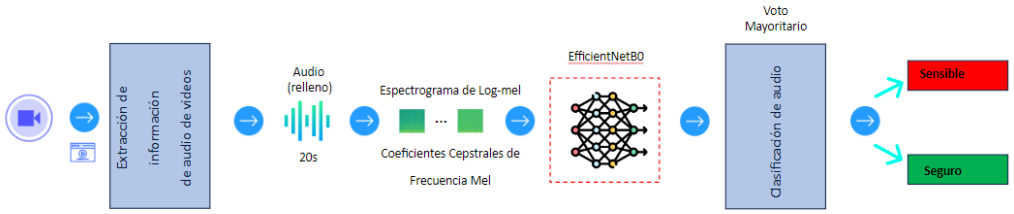

Figura 4.12: Flujo de trabajo simplificado

Capítulo 5

Experimentos y Resultados

Este capítulo detalla los experimentos realizados para evaluar la eficacia de modelos conseguidos.

El objetivo de estos experimentos no solo era evaluar la capacidad de los modelos para clasificar con precisión el contenido, también explorar cómo diferentes configuraciones y técnicas de procesamiento, como el aumento de datos o la selección de parámetros, pueden influir en la precisión de estos modelos.

5.1. Configuración de los Experimentos

Los experimentos se llevaron a cabo utilizando diferentes configuraciones de modelos y técnicas para optimizar la clasificación de contenido sensible y seguro en audios. Se emplearon varias arquitecturas de redes neuronales, incluyendo *ConvNext* o *EfficientNet*.

Inicialmente, se realizó una partición aleatoria del conjunto de datos de vídeos en un 80/20, asignando el 80 % de los datos al entrenamiento y el 20 % a la prueba y luego se obtuvieron los audios correspondientes. Esta hizo para que el modelo pueda entrenarse en diferentes muestras del conjunto de datos.

Para asegurar la generalización del modelo y evitar el sobreajuste, se implementó una validación cruzada con cinco *folds*, ayudando a maximizar el uso de los datos. Se seleccionaron cinco *folds* (igual que en la primera división se obtuvieron los índices de los vídeos de entrenamiento y validación y posteriormente los audios correspondientes a cada uno de estos grupos.) recomendaciones estándar para lograr un buen equilibrio de los datos.

5.2. Experimentos Iniciales

Se comenzó con una configuración de red donde solo la última capa de el modelo preentreando *EfficientNet* estaba habilitada para el entrenamiento, mientras que las capas anteriores permanecían "congeladas", es decir, sus pesos no se modificaban. Esto permitió al modelo ajustar inicialmente solo los parámetros de la capa de salida.

Posteriormente, se procedió a *"descongelar"* gradualmente más capas del modelo. El descongelamiento de capas es un enfoque común, donde se comienza el entrenamiento

congelando la mayoría de las capas de un modelo pre-entrenado, para luego ir habilitando más capas para ir ajustando a medida que el entrenamiento avanza. Este método es útil para ajustar mejor los modelos a conjuntos de datos.

Tras algunos experimentos iniciales se configuraron algunos hiperparámetros que se conservaron a lo largo de los experimentos:

5.2.1. Descripción de algunos Hiperparámetros Clave

- **Tasa de aprendizaje (*Learning rate*)**: El tamaño del paso en la actualización de los pesos. Un valor pequeño puede llevar a un aprendizaje lento, mientras que un valor demasiado alto puede provocar que el modelo no converja.
- **Tamaño de lote (*Batch size*)**: Cantidad de muestras procesadas antes de actualizar el modelo.
- **Decaimiento de peso (*Weight decay*)**: Técnica de regularización que ayuda a prevenir el sobreajuste al añadir una penalización en los pesos grandes durante la optimización.
- **Número de épocas (*Num epochs*)**: Número de veces que el algoritmo de aprendizaje trabaja a través de todo el conjunto de datos. Más épocas pueden mejorar el aprendizaje hasta cierto punto antes de que empiece el sobreajuste.
- **Número de mels**: Especifica el número de bandas de frecuencia Mel usadas para transformar la señal de audio en el espectrograma Mel,más Mels proporcionan mayor detalle pero también incrementan la complejidad computacional.
- **Parada temprana (*Early stopping*)**: Es una forma de regularización utilizada para evitar el sobreajuste durante el entrenamiento. Detiene el entrenamiento tan pronto como deje de mejorar.

Los valores están en la Tabla 5.1:

5.2.2. Validación Cruzada con Una Época

Se realizó una prueba inicial de validación cruzada con solo una época para evaluar la estabilidad inicial del modelo:

- Resultados por *fold*: [89,19 %, 86,86 %, 89,98 %, 89,64 %, 87,40 %]
- Media: 88,62 %
- Varianza: 0,02 %
- Desviación estándar: 0,13 %

5.2.3. Evaluación 30 Épocas

Se extendió el entrenamiento a 30 épocas para observar la mejora en el rendimiento:

Tabla 5.1: Hiperparámetros coincidentes

Nombre	Valor
Tamaño de Imagen	224
Tamaño del Lote	16
Tasa de Muestreo	16.000
Tasa de Aprendizaje	0,0001
Decaimiento de Peso	0,0001
Número de Mels	26
Número de Épocas	30
Número de Capas Descongeladas	20
Parada temprana	5
Optimizador	Adam

- Resultados por *fold*: [91,38 %, 93,88 %, 91,61 %, 92,77 %, 91,19 %]
- Media: 92,17 %
- Varianza: 0,01 %
- Desviación estándar: 0,10 %

5.3. Resultados con EfficientNet

Se implementó un modelo *EfficientNet* para comparar su eficacia:

- Resultados por *fold*: [91,77 %, 93,26 %, 91,49 %, 92,70 %, 90,08 %]
- Media: 91,86 %
- Varianza: 0,01 %
- Desviación estándar: 0,11 %

5.4. Experimentos de Aumento de Datos

Inicialmente, se exploraron diferentes técnicas de aumento de datos utilizando la biblioteca *audiomentations* [Kar]. Se experimentaron técnicas como el cambio de tono, aumento de ruido y cambios de velocidad, entre otros.

Sin embargo, a lo largo de los experimentos, se descubrió que añadir ruido blanco resultaba ser la más efectivo en términos de mejorar las métricas. El ruido blanco es un tipo de ruido producido por una señal que contiene todas las frecuencias en igual medida.

La implementación del aumento de ruido blanco se realizó mediante la librería de TensorFlow. El proceso consiste en generar una muestra de ruido blanco que se suma al audio original, lo cual puede ser controlado por un parámetro que determina la intensidad del ruido a añadir. Durante los experimentos, se probaron diferentes niveles de intensidad para encontrar el balance adecuado que maximizara el rendimiento del modelo sin distorsionar demasiado el contenido original del audio. En la Figura 5.1 se puede ver como se representa el ruido blanco en una imagen.

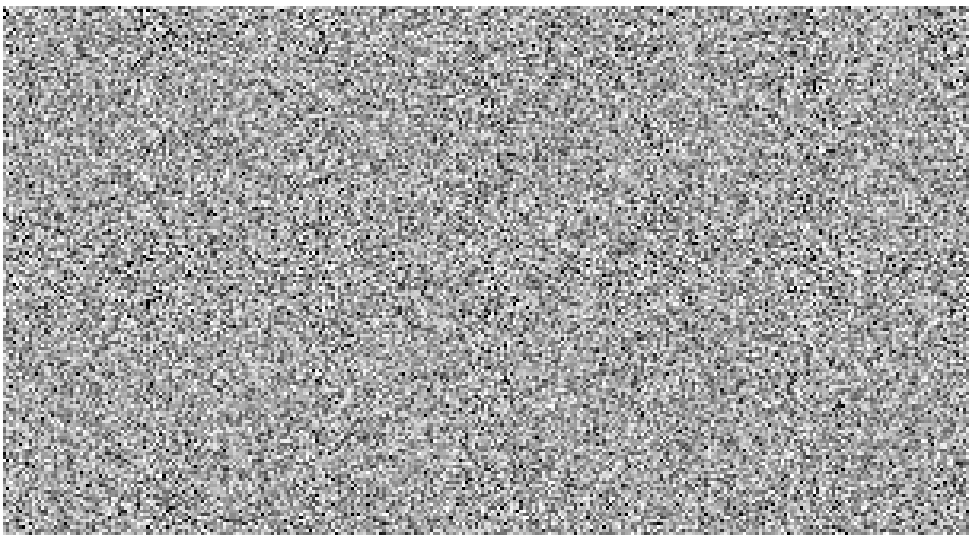

Figura 5.1: Imagen de un ruido blanco [Tre23].

5.4.1. Resultados del Aumento de Datos

El uso de ruido blanco demostró ser efectivo al mejorar la capacidad del modelo para generalizar a partir de los datos de entrenamiento. Los experimentos mostraron que, bajo las mismas condiciones de prueba y parámetros de modelo, el ruido blanco superaba consistentemente las otras técnicas de aumento probadas.

5.5. Análisis de Errores

Se realizó un análisis detallado de los falsos positivos y falsos negativos para entender las limitaciones del modelo:

5.5.1. Falsos Positivos

Los falsos positivos involucraron principalmente:

- **Silencio y Música de Ritmo Rápido:** Segmentos donde predominaba el silencio o música de ritmo rápido, como en los archivos *vNonPorn000063* (silencio completo) y *vNonPorn000215* (tic-tacs rápidos), fueron mal clasificados con probabilidades alrededor del 60 %. El modelo podría estar confundiendo la falta de características sonoras claras con contenido sensible.

- **Conversaciones Rápidas:** Audios con diálogos rápidos también fueron frecuentemente mal clasificados, ejemplificados por *vNonPorn000309* y *vNonPorn000314*, sugiriendo dificultades del modelo para distinguir entre ritmos rápidos y contenido sensible.
- **Sonidos de agua:** El modelo a menudo confunde los sonidos de agua con contenido sensible.
- **Contenido Explícitamente Sensible:** Audios claramente sensibles como *vPorn000254* fueron identificados correctamente con altas probabilidades (70-90 %), demostrando efectividad en condiciones claras.
- **Ruido de Fondo:** Combinaciones de contenido sensible con ruidos de fondo, como *vPorn000859*, mostraron menores probabilidades (50-60 %).

5.6. Pruebas con Nuevas Configuraciones

5.6.1. ConvNextTiny con Aumento de Ruido Blanco

Para evaluar la eficacia de otras arquitecturas, se introdujo el modelo *convnextTiny* como una alternativa a *efficientNet*. Esta configuración incluyó la técnica de aumento de datos con ruido blancos. El objetivo era verificar si el nuevo modelo, combinado con la mejora en los datos, podría superar los resultados obtenidos anteriormente. Sin embargo, se optó por seguir utilizando *efficientNet*.

- Resultados por iteración: [90,43 %, 91,47 %, 91,47 %, 91,29 %]
- Media: 92,16 %
- Varianza: 0,0017 %
- Desviación estándar: 0,36 %

5.7. Evaluación Final y Selección de los Modelos

Se configuraron dos modelos distintos con el objetivo de evaluar su eficacia. Para ello, se tomaron en cuenta las conclusiones obtenidas en experimentos anteriores, permitiendo seleccionar las configuraciones más eficaces, como el uso de ruido blanco para el aumento de datos y la optimización mediante *Adam*, ajustando sus parámetros para maximizar los resultados. Los modelos se desarrollaron sobre la base de diferentes características distintivas:

El macro y ponderado se utilizan para ofrecer dos perspectivas distintas de agregación de las métricas de precisión, *recall* y *F1-Score*. El macro se refiere a la media de las métricas calculadas independientemente para cada clase y luego promediadas sin tener en cuenta el soporte de cada clase. En cambio, el ponderado, calcula la media para cada clase ponderadas por el soporte de cada una.

- **Modelo 2.1**: Se utilizó espectrogramas de Log Mel. En la Tabla 5.2 se ven los resultados del entrenamiento del modelo, en la Tabla 5.3 su matriz de confusión.

Tabla 5.2: Tabla de resultados del modelo usando espectrogramas de Mel

Modelo Log Mel	Precisión	Sensibilidad	F1-Score	Soporte
Seguro	87 %	81 %	84 %	1.562
Sensible	93 %	95 %	94 %	3.805
Precisión			91 %	5.367
Macro	90 %	88 %	89 %	5.367
Ponderado	91 %	91 %	91 %	5.367

Tabla 5.3: Matriz de confusión del modelo usando espectrogramas de Mel

	Predicción seguro	Predicción sensible
Seguro	1.272	290
Sensible	197	3.608

- **Modelo 2.2**: Se utilizó MFCCs.En la Tabla 5.4 se ven los resultados del entrenamiento del modelo, en la Tabla 5.5 su matriz de confusión.

Tabla 5.4: Tabla de resultados del modelo usando MFCCs

Modelo MFCC	Precisión	Sensibilidad	F1-Score	Soporte
Seguro	91 %	78 %	84 %	1.562
Sensible	91 %	97 %	94 %	3.805
Precisión			91 %	5.367
Macro	91 %	87 %	89 %	5.367
Ponderado	91 %	91 %	91 %	5.367

5.8. Evaluación de Modelos

5.8.1. Evaluación Individual de Modelos

- Se realizaron pruebas de validación cruzada para ambos modelos, primero con una sola época para determinar la estabilidad inicial y luego extendiendo a 30 épocas para observar mejoras en el rendimiento.

- Se utilizó la técnica de detención temprana para evitar el sobreajuste y mejorar la eficiencia del entrenamiento.

Tabla 5.5: Matriz de confusión del modelo usando MFCCs

	Predicción seguro	**Predicción sensible**
Seguro	1.213	349
Sensible	115	3.690

5.8.2. Comparación de Modelos

- Ambos modelos se evaluaron bajo las mismas condiciones de prueba para asegurar una comparación justa.
- Se calculó la precisión, el *recall* y el *F1-Score* para cada modelo, y se identificaron los falsos positivos y falsos negativos para entender las limitaciones específicas de cada configuración.

5.8.3. Combinación de Probabilidades de Modelos

- Después de obtener las predicciones de cada modelo, se experimentó con la combinación de sus probabilidades para cada clase, promediando los resultados de ambos modelos. Esta técnica es una forma de *ensembling*, donde se utilizan múltiples modelos de aprendizaje para mejorar la robustez y precisión de las predicciones finales.
- Esta técnica buscaba aprovechar las fortalezas de ambos modelos y mejorar la precisión general en la clasificación. Los métodos de *ensembling* son conocidos por su capacidad para reducir el riesgo de sobreajuste y mejorar la generalización sobre datos no vistosa [Die00].

5.9. Umbral Óptimo

Para maximizar la precisión del modelo en la clasificación de contenido, se llevó a cabo un análisis exhaustivo para encontrar el umbral óptimo utilizando las predicciones combinadas de ambos modelos. El mejor umbral fue 60 %.:

5.10. Resultados

- Los resultados individuales mostraron un rendimiento sólido, con el Modelo 2.1 alcanzando una precisión máxima del 90.93 % y el Modelo 2.2 mejorando ligeramente a un 91.35 %.
- La combinación de probabilidades mejoró aún más la precisión a un 92.06 %, demostrando que la integración de las salidas de múltiples modelos puede ser una estrategia efectiva para la clasificación de audios.

En la Tabla 5.6 se ven los resultados del entrenamiento del modelo final, en la Tabla 5.7 su matriz de confusión.

Tabla 5.6: Tabla de resultados del modelo usando espectrogramas de Mel

Modelo MFCC	Precisión	Sensibilidad	F1-Score	Soporte
Seguro	89 %	83 %	86 %	1.562
Sensible	93 %	96 %	94 %	3.805
Precisión			92 %	5.367
Macro	91 %	89 %	90 %	5.367
Ponderado	92 %	92 %	92 %	5.367

Tabla 5.7: Matriz de confusión del modelo usando espectrogramas de Log Mel

	Predicción seguro	Predicción sensible
Seguro	82,84 %	17,16 %
Sensible	4,15 %	95,85 %

5.11. Análisis Final y Conclusiones

- El análisis de errores reveló que ciertos tipos de audios, como música de ritmo rápido y habla rápida, eran propensos a ser clasificados incorrectamente, lo que indica áreas de mejora para futuras iteraciones del modelo.
- La combinación de las predicciones de ambos modelos no solo mejoró la precisión sino también la robustez del sistema frente a variaciones en los datos de entrada.

5.12. Conclusión

Los experimentos demostraron que el enfoque de combinación de técnicas y ajustes de modelo puede mejorar significativamente la precisión en la clasificación de contenido sensible en audios. Las pruebas futuras incluirán la exploración de más arquitecturas y técnicas de aumento de datos para continuar mejorando el rendimiento del modelo.

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

Los resultados de los experimentos con los modelos 2.1 y 2.2 demuestran la efectividad de los métodos utilizados. La precisión alcanzada, superior al 90 %, indica que se ha alcanzado un alto nivel de eficacia en la clasificación.

El uso de diferentes características de las imágenes sacadas de los audios, como los espectrogramas de Mel y MFCCs, ha permitido a los modelos capturar detalles que ayudan a la clasificación del contenido. La combinación de probabilidades de ambos modelos ha mejorado aún más la precisión hasta un 92,06 %, validando positivamente la estrategia de combinar los modelos.

6.1.1. Viabilidad del Modelo

El modelo ha demostrado ser efectivo con archivos de audio y de vídeo, al convertir estos últimos en segmentos de audio, permitiendo que el modelo se adapte a diferentes tipos de entradas. También cabría la posibilidad de usar este modelo con otros como modelos de clasificación de contenido sensible en vídeos o imágenes.

6.1.2. Limitaciones

A pesar de sus puntos fuertes, el modelo presenta algunas limitaciones que deben abordarse en futuros desarrollos:

- **Clasificación errónea de sonidos de agua:** El modelo a menudo confunde los sonidos de agua con contenido sensible, lo que puede llevar a FP.
- **Precisión no absoluta:** Aunque el modelo es efectivo, no alcanza una precisión perfecta, lo que indica la necesidad de ajustes continuos y mejoras en las técnicas de AP.
- **Dependencia de la calidad del audio:** La efectividad del modelo puede verse comprometida por la calidad del audio en los vídeos, aunque en el aumento de datos se ha intentado “desmejorar el sonido” con ruido blanco para evitar esto.

Obviamente, cuanto mayor sea la calidad del audio, mejores serán los resultados del modelo.

6.2. Trabajo Futuro

6.2.1. Análisis de Falsos Positivos y Negativos

Reducción de FP y FN, especialmente en casos donde la música de fondo rápida, la conversación rápida o el caso del sonido del agua mencionado anteriormente pueden confundir al modelo. El análisis detallado de estos errores puede ser importante para el desarrollo de modelos más robustos.

6.2.2. Incorporación de Temporalidad

La temporalidad en secuencias de vídeos y audios completos puede ofrecer una dimensión adicional de análisis, permitiendo a los modelos entender mejor el contexto y la continuidad del contenido audiovisual. En lugar de dividir el vídeo o audio en segmentos individuales para su análisis, se propone que el modelo procese el contenido completo. Esta aproximación puede mejorar la precisión al identificar el contenido de vídeos y audios, ya que se preserva la coherencia y el flujo natural del contenido, proporcionando una mejor comprensión.

6.2.3. Uso de Tecnologías de Incrustación

La experimentación con tecnologías de incrustación avanzadas, como pasar de onda a vector, presenta una oportunidad prometedora para capturar características más profundas del contenido auditivo. Las incrustaciones pueden proporcionar una representación más detallada que podría ser muy útil para clasificar con precisión el contenido. Aunque se realizaron pruebas preliminares, finalmente no se escogió para este proyecto. Sin embargo, su potencial sugiere que futuras investigaciones podrían beneficiarse de su implementación [vec24].

6.2.4. Mejora del Conjunto de Datos

Una revisión y expansión del conjunto de datos que se ha usado en el trabajo puede ser clave para aumentar la robustez del modelo. Ampliar datos, puede ayudar a disminuir la tasa de FP y FN. Esta actualización del conjunto de datos también debería considerar incorporar muestras que cubran casos atípicos como el caso del agua.

6.3. Consideraciones Finales

El desarrollo del modelo de clasificación de contenido sensible en audio ha demostrado ser una herramienta eficaz, con resultados que superan el 90 % de precisión en las pruebas realizadas.

La metodología adoptada, que incluye la combinación de predicciones de múltiples modelos para mejorar la precisión global, refleja una estrategia bien pensada y ejecutada. Sin embargo, es crucial reconocer las limitaciones actuales del modelo, como la clasificación errónea de ciertos sonidos o la dependencia de la calidad del audio.

Este estudio subraya la necesidad de seguir evolucionando y adaptando las herramientas utilizadas a nuevos retos y escenarios de uso, para mantener su relevancia y eficacia en un mundo digital en constante cambio.

6.4. Perspectivas de Futuro

De cara al futuro, hay varias direcciones prometedoras para expandir y enriquecer este trabajo. Una de ellas es la adaptación del modelo para que sea compatible con una gama más amplia de dispositivos de captura y bajo diversas condiciones ambientales, lo que aumentaría significativamente su robustez y aplicabilidad en escenarios reales.

Finalmente, extender la investigación para incluir nuevas formas de procesamiento y representación del audio, como la tecnología de pasar de onda a vector y otras técnicas de incrustación avanzadas, podría proporcionar medios aún más eficaces para capturar las complejidades del contenido audio que necesita ser clasificado, abriendo la puerta a un sistema más refinado y sensible a los contextos variados del contenido sensible.

Capítulo 7

Introduction

7.1. Motivation

Currently, access to new technologies has changed the way content is created, shared, and consumed on the internet. This has brought benefits but has also facilitated the dissemination of illegal content, such as child pornography. Some recent reports highlight an increase in this type of content.

The State Attorney General's Office has expressed concern about the increase in the production of child pornography, highlighting the urgency of improving technological tools to combat these crimes [not23]. Furthermore, the year 2023 has recorded a record number of detections of child sexual abuse in Spain [elp].

One of the most challenging aspects is that videos of this type of content do not always show explicit nudity, making it difficult to detect using image-based methods. Audio, which is often overlooked, can contain clues about the video's content.

This work aims to develop technology capable of classifying sensitive content through audio analysis. The motivation is to provide more effective tools to combat the dissemination of these materials, ensuring a safer digital environment for minors.

7.2. Context

This Final Degree Project is part of a research project titled *Child protection centred strategies to fight against sexual abuse and exploitation – ALUNA*, approved by the European Commission under the ISF-2021-TF1-AG-CYBER call, grant agreement number 101084929. The GASS Group of the Complutense University of Madrid (*Analysis, Security and Systems Group*, `https://gass.ucm.es`, group 910623 of the catalog of recognized research groups by UCM) participates as the project coordinator.

7.3. Research Objective

This study focuses on the development and implementation of technologies for detecting and classifying sensitive content in audio tracks. Typically, the detection of this type of content focuses on images, but audio can provide another approach, especially

in cases where visual content may be difficult to detect.

The research explores the application of AP techniques to classify audio fragments extracted from videos, seeking patterns or characteristics that can detect this content.

The development of these models aims not only to reduce the amount of harmful content online but also to improve the automation of detecting such content.

7.4. Work Plan

The development of this work was carried out in three phases:

1. **Research**: The first part of the work was an introduction and learning phase about the project topic, laying the foundations to begin development. Initially, some basic concepts were reviewed, and different ways to start acquiring this knowledge were investigated. For example, *Google Scholar* can be used to search for scientific articles on research in this field. Part of an AP course was also viewed [NG]. After this research period, development began.

2. **Development**: After obtaining the necessary theoretical foundation for the project, the research phase was reduced to give way to development. Research continued but in a more targeted manner to resolve issues arising during development. In this stage, efforts were made to better master the *Python* programming language and adapt to libraries such as *Keras* and *TensorFlow*, which were useful for development.

3. **Experimentation**: During the experimentation stage, after developing the initial models, tests were conducted using some of the aforementioned libraries. In this period, a comparison of the results obtained was made to adjust the model parameters. Data augmentation techniques were implemented to evaluate and improve the model. It is important to note that development continued alongside experimentation.

7.5. Work Structure

The work is organized into 6 chapters and 4 appendices with the following structure: Chapter 2 introduces some concepts in the context of audio analysis using deep learning techniques, focusing on identifying and classifying sensitive content.

Chapter 3 provides a review of the state of the art in identifying sensitive content in audio and videos using advanced deep learning and signal processing techniques.

Chapter 4 describes the methodology for developing and evaluating a model needed for this work. Key technologies, the dataset used for training the model, and the overall model architecture are mentioned. It concludes by describing the workflow, including input stages, preprocessing, data augmentation, training, model evaluation, and the implementation of a user interface for real-time performance testing.

Chapter 5 describes all the experiments carried out along with their results.

Chapter 7 presents the conclusions achieved throughout the work, mentioning limitations and possible ideas that could be developed in the future.

Chapters 8 and 6 are translations into English of the Introduction and Conclusions.

Capítulo 8

Conclusions and Future Work

8.1. Conclusions

The results from experiments with models 2.1 and 2.2 demonstrate the effectiveness of the methods used. The accuracy achieved, over 90 %, indicates a high level of efficacy in classification.

The use of different features extracted from audio images, such as Mel Spectrograms and MFCCs, has enabled the models to capture details that aid in content classification. Combining probabilities from both models further improved the accuracy to 92.06 %, positively validating the strategy of model combination.

8.1.1. Model Viability

The model has proven to be effective with audio and video files, converting the latter into audio segments, allowing the model to adapt to different types of inputs. There is also the possibility of using this model with others, such as models for classifying sensitive content in videos or images.

8.1.2. Limitations

Despite its strengths, the model presents some limitations that need to be addressed in future developments:

- **Misclassification of water sounds:** The model often confuses water sounds with sensitive content, which can lead to FP.
- **Non-absolute accuracy:** While the model is effective, it does not achieve perfect accuracy, indicating the need for continuous adjustments and improvements in AP techniques.
- **Dependence on audio quality:** The effectiveness of the model can be compromised by the quality of the audio in videos, although data augmentation has attempted to ”degrade the sound”with white noise to avoid this. Obviously, the higher the audio quality, the better the results of the model.

8.2. Future Work

8.2.1. Analysis of False Positives and Negatives

Reduction of FP and FN, especially in cases where fast background music, rapid conversation, or the mentioned case of water sound may confuse the model. A detailed analysis of these errors can be important for the development of more robust models.

8.2.2. Incorporation of Temporality

Temporality in sequences of complete videos and audios can offer an additional dimension of analysis, allowing models to better understand the context and continuity of audiovisual content. Instead of dividing the video or audio into individual segments for analysis, it is proposed that the model processes the complete content. This approach can improve accuracy in identifying the content of videos and audios, as it preserves the coherence and natural flow of the content, providing a better understanding.

8.2.3. Use of Embedding Technologies

Experimentation with advanced embedding technologies, such as transitioning from wave to vector, referred to as , presents a promising opportunity to capture deeper features of auditory content. Embeddings can provide a more detailed representation that could be very useful for accurately classifying content. Although preliminary tests were conducted with , it was ultimately not chosen for this project. However, its potential suggests that future research could benefit from its implementation [vec24].

8.2.4. Improvement of the Dataset

A review and expansion of the dataset used in the work can be key to increasing the robustness of the model. Expanding data can help to decrease the rate of FP and FN. This update of the dataset should also consider incorporating samples that cover outlier cases like the case of water.

8.3. Final Considerations

The development of the model for classifying sensitive content in audio has proven to be an effective tool, with results exceeding 90 % accuracy in the tests performed.

The methodology adopted, which includes combining predictions from multiple models to improve overall accuracy, reflects a well-thought-out and executed strategy. However, it is crucial to recognize the current limitations of the model, such as misclassification of certain sounds or dependence on audio quality.

This study underscores the need to continue evolving and adapting the tools used to new challenges and usage scenarios, to maintain their relevance and effectiveness in a constantly changing digital world.

8.4. Future Perspectives

Looking ahead, there are several promising directions to expand and enrich this work. One is adapting the model to be compatible with a wider range of capture devices and under various environmental conditions, which would significantly increase its robustness and applicability in real-world scenarios.

Finally, extending research to include new forms of processing and representation of audio, such as the technology and other advanced embedding techniques, could provide even more effective means to capture the complexities of the audio content that needs to be classified, opening the door to a more refined and context-sensitive system.

Bibliografía

[ABC+15] Martín Abadi, Paul Barham, Jianmin Chen, et al. TensorFlow: Una Plataforma de Aprendizaje Automático de Código Abierto. https://www.tensorflow.org, 2015.

[AdFBGC20] Pedro Almeida de Freitas, Antonio Busson, Alan Guedes, and Sérgio Colcher. A Deep Learning Approach to Detect Pornography Videos in Educational Repositories. 11 2020.

[Arc19] Juan Ignacio Barrios Arce. Matriz de confusión bianria. https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/, 2019.

[ATC+13] Sandra Avila, Nicolas Thome, Matthieu Cord, Eduardo Valle, and Arnaldo De A. Araújo. Pooling in Image Representation: The Visual Codeword Point of View. *Computer Vision and Image Understanding*, 117(5):453–465, 2013.

[BAKL+19] Rasoul Banaeeyan, Hezerul Abdul Karim, Mohd-Haris Lye, Mohammad Faizal Ahmad Fauzi, Sarina Mansor, and John See. Acoustic Pornography Recognition using Fused Pitch and Mel-Frequency Cepstrum Coefficients. *International Journal of Technology*, 10:1335, 11 2019.

[Bre01] L. Breiman. Random Forest. *Machine Learning*, 45:5–32, 2001.

[BT20] Ankita Bose and BK Tripathy. Deep learning for audio signal classification. *Deep learning research and applications*, pages 105–136, 2020.

[C+15] François Chollet et al. Keras: The Python Deep Learning API. https://keras.io, 2015.

[cap] Capas de una Red Neuronal.

[cnna] Definición y Funcionamiento de las Redes Neuronales Convolucionales. https://datascientest.com/es/convolutional-neural-network-es.

[cnnb] Definición de las Redes Convolucionales. https://www.ibm.com/es-es/topics/convolutional-neural-networks.

[Die00] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[DR21] Jorge Díaz-Ramírez. Aprendizaje automático y aprendizaje profundo. *Ingeniare. Revista chilena de ingeniería*, 29(2):180–181, 2021.

[Edw] Edward Wu. Aprendizaje Supervisado y No Supervisado. https://www.extrahop.com/blog/supervised-vs-unsupervised-machine-learning-for-network-threat-detection.

[elp] Noticia de elperiodico. https://efe.com/espana/2024-04-09/agresiones-sexuales-infancia-ninos-adolescentes-anar/.

[gra] Logo de Gradio. https://seeklogo.com/vector-logo/515012/gradio-icon.

[Gre22] Rudeus Greyrat. Introducción a los Conjuntos de Datos. https://barcelonageeks.com/conjuntos-de-entrenamiento-prueba-y-validacion/, 2022.

[H+03] John D. Hunter et al. Matplotlib: Visualización con Python. https://matplotlib.org, 2003.

[Haq] Kh. Nafizul Haque. Extracción de características de una Red neuronal Convolucional. https://www.linkedin.com/pulse/what-convolutional-neural-network-cnn-deep-learning-nafiz-shahriar/.

[HVD+15] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[HZF21] Qibin Hou, Daquan Zhou, and Jiashi Feng. Coordinate attention for efficient mobile network design. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13713–13722, 2021.

[Int] Introducción a la Inteligencia Artificial. https://www.ibm.com/es-es/topics/artificial-intelligence.

[int17] Espectrograma de una senal de vo. https://www.ece.rice.edu/~dhj/courses/elec241/spectrogram.html, 2017.

[int22] Señal de Audio Digitalizada. https://dsp.stackexchange.com/questions/28001/how-can-i-center-an-audio-signal, 2022.

[Jen19] Ashlin Jenifa. Diferencia entre Clasificación y Regresión. https://geekflare.com/es/regression-vs-classification/, 2019.

[JPA23] D Joshi, J Pareek, and P Ambatkar. Comparative study of mfcc and mel spectrogram for raga classification using cnn. *Indian J Sci Technol*, 16(11):816–822, 2023.

[Kar] Ola Karlsson. Audiomentations: Una biblioteca de Python para el Aumento de Datos de Audio. https://github.com/iver56/audiomentations.

[KB14] Diederik P. Kingma and Jimmy Ba. Adam: Un Método Para La Optimización Estocástica. 2014.

[Kea95] Michael Kearns. A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split. *Advances in neural information processing systems*, 8, 1995.

[ker] Logo de Keras. https://keras.io/.

[Ket21] Ketan Doshi. Espectrogramas de Mel. https://ketanhdoshi.github.io/Audio-Mel/, January 2021.

[LAP+09] Ana Lopes, Sandra Avila, Anderson Peixoto, Rodrigo Oliveira, and Marcelo Coelho. Nude Detection in Video Using Bag-of-Visual-Features, 10 2009.

[LBH15] Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.

[LCHL11] JaeDeok Lim, ByeongCheol Choi, SeungWan Han, and ChoelHoon Lee. Automatic Classification of X-rated Videos using Obscene Sound Analysis based on a Repeated Curve-like Spectrum Feature, 2011.

[Lel20] Leland Roberts. Entendiendo los Espectrogramas de Mel. https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53, 2020.

[lib] Logo de Librosa. https://github.com/librosa/librosa.

[LLF22] Holy Lovenia, Dessi Puji Lestari, and Rita Frieske. What Did I Just Hear? Detecting Pornographic Sounds in Adult Videos Using Neural Networks. http://dx.doi.org/10.1145/3561212.3561244, September 2022. AudioMostly 2022, ACM.

[LLLZ23] Shangfeng Liu, Ruwei Li, Qiuyan Li, and Jingyu Zhao. Porn streamer audio recognition based on deep learning and random forest. *Applied Intelligence*, 53(15):18857–18867, 2023.

[MAP+16] Daniel Moreira, Sandra Avila, Mauricio Perez, Daniel Moraes, Vanessa Testoni, Eduardo Valle, Siome Goldenstein, and Anderson Rocha. Pornography classification: The hidden clues in video space–time. *Forensic Science International*, 268:46–61, 2016.

[mat] Logo de Matplotlib. https://brandfetch.com/matplotlib.org.

[MBE10] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*, 2010.

[McK10] Wes McKinney. Potentes Estructuras de Datos para el Análisis de Datos, Series Temporales y Estadísticas. https://pandas.pydata.org, 2010.

[met] Métricas de Evaluación. https://bookdown.org/keilor_rojas/CienciaDatos/aprendizaje-autom%C3%A1tico-machine-learning.html.

[MRL15] Brian McFee, Colin Raffel, and Dawen y otros Liang. Librosa: Análisis de Señales de Audio y Música en Python. https://librosa.org, 2015.

[NG] Andrew NG. Curso de Coursera de Andrew Ng. https://www.coursera.org/specializations/deep-learning.

[not23] Noticia del periódico 20 minutos. https://www.20minutos.es/noticia/5174107/0/fiscalia-teme-inteligencia-artificial-incremente-pornografia-infantil/, 2023.

[pan] Logo de Pandas. https://pandas.pydata.org/about/citing.html.

[Piq20] Víctor Yepes Piqueras. Inteligencia artificial, Aprendizaje Automático y Profundo. https://victoryepes.blogs.upv.es/2020/09/15/, 2020.

[PVG11] Fabian Pedregosa, Gaël Varoquaux, and Alexandre y otros Gramfort. Scikit-Learn: Aprendizaje Automático en Python. https://scikit-learn.org, 2011.

[pyt] Logo de Python. https://www.clipsafari.com/clips/o248484-python-logo.

[reg] Algoritmos Supervisados: Clasificación y Regresión. https://tutorialforbeginner.com/regression-vs-classification-in-machine-learning.

[Roh22] Rohit Kundu. Métricas de Evaluación. https://www.v7labs.com/blog/f1-score-guide, 2022.

[set20] Conjuntos de Entrenamiento, Test y Validación. https://www.aprendemachinelearning.com/sets-de-entrenamiento-test-validacion-cruzada/, 2020.

[sft] Transformada de Fourier. http://www.sc.ehu.es/sbweb/fisica3/simbolico/fourier/fourier_1.html.

[Sru24] Sruthy. Diferencias entre Aprendizaje Automático y Profundo. https://www.softwaretestinghelp.com/data-mining-vs-machine-learning-vs-ai/, 2024.

[SSK20] Ritu Sharma, Kavya Sharma, and Apurva Khanna. Study of supervised learning and unsupervised learning. *International Journal for Research in Applied Science and Engineering Technology*, 8(6):588–593, 2020.

[stra] Herramienta Streamlit. https://streamlit.io/.

[strb] Logo de Streamlit. https://www.infocentric.com.au/2024/04/12/unlocking-the-power-of-data-visualisation-with-snowflake-streamlit/.

[SVI+16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[SZ15] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, 2015.

[Tea24] Team Gyata. Validación Cruzada de Aprendizaje Automático. https://www.gyata.ai/es/machine-learning/machine-learning-cross-validation/, 2024.

[tfl] Logo de Tensorflow. https://www.tensorflow.org/extras/tensorflow_brand_guidelines.pdf.

[tip23] Tipos de Redes Neuronales. https://www.youtube.com/watch?v=_D2RvMN_xCA&ab_channel=AtAGlance%21, 2023.

[Tre23] Luis Diego Pérez Trejo. Imagen de cómo se representa el ruido blanco. https://escuchamexico.iteso.mx/el-ruido-blanco-un-auxiliar-para-el-descanso/, 2023.

[Val] Alberto Torrejón Valenzuela. Curva ROC. https://www.rpubs.com/albtorval/AUC.

[vec24] Procesamiento de audios mediante vectores. https://ai.meta.com/research/impact/wav2vec/, 2024.

[VR91] Guido Van Rossum. Lenguaje de Programación Python. https://www.python.org, 1991.

[WZW+20] Liyuan Wang, Jing Zhang, Meng Wang, Jimiao Tian, and Li Zhuo. Multilevel fusion of multimodal deep features for porn streamer recognition in live video. *Pattern Recognition Letters*, 140:150–157, 2020.

[YN22] Kanwal Yousaf and Tabassam Nawaz. A Deep Learning-Based Approach for Inappropriate Content Detection and Classification of YouTube Videos. *IEEE Access*, 10:16283–16298, 2022.

[Zul14] Zulko. MoviePy: Editar Vídeos con Python. https://zulko.github.io/moviepy/, 2014.

[ZWL+22] Lifeng Zhou, Kaifeng Wei, Yuke Li, Yiya Hao, Weiqiang Yang, and Haoqi Zhu. Acoustic Pornography Recognition Using Convolutional Neural Networks and Bag of Refinements, 2022.