Informe : Proyecto Clasificación de Audio

Firmin Cadot

17 Novembre 2023

1. Introducción

El procesamiento de señales de audio es una rama del aprendizaje automático y la inteligencia artificial que ha ganado una atención significativa en la última década. Con la proliferación de dispositivos inteligentes capaces de capturar y procesar sonido, la necesidad de algoritmos avanzados para comprender y clasificar señales de audio se ha vuelto imperativa. Este proyecto se sitúa en la intersección de la ingeniería de señales y la inteligencia artificial, buscando construir y afinar modelos de aprendizaje profundo capaces de identificar y clasificar una amplia gama de sonidos.

El campo de aplicación de la clasificación de audio es extenso y variado, abarcando desde el reconocimiento de voz hasta la categorización de sonidos ambientales o de instrumentos musicales. El propósito de este trabajo es desarrollar un modelo robusto y preciso que, mediante el uso de técnicas de procesamiento de señales y arquitecturas de redes neuronales, pueda predecir con exactitud la categoría de un sonido dado su representación acústica.

El presente informe resume el enfoque adoptado para enfrentar este desafío, detallando la estructura de los notebooks de Jupyter utilizados, la metodología aplicada en la solución, las iteraciones de desarrollo realizadas y los resultados obtenidos. A través de este documento, proporcionamos una visión comprensiva del proceso de diseño, implementación y evaluación de modelos de clasificación de audio utilizando dos tipos de modelos principales: redes neuronales convolucionales (CNN) y redes neuronales recurrentes (RNN), destacando nuestras contribuciones y aprendizajes en el proceso.

2. Descripción de la Estructura de los Notebooks

2.1. Notebook 1 - Análisis Exploratorio de Datos (eda.ipynb)

El notebook *eda.ipynb* está estructurado para facilitar el análisis exploratorio de datos (EDA) de señales de audio con el objetivo de comprender mejor sus características y preparar los datos para la modelización. A continuación, se detallan las secciones principales y sus funciones en el análisis:

Visualización de Datos en Diversas Formas : Esta sección está dedicada a la representación gráfica de las señales de audio, que es fundamental para entender sus características. Incluye:

- **Trama de Serie Temporal:** Muestra la amplitud de la señal a lo largo del tiempo, proporcionando la visualización más directa de la variación de las ondas sonoras.
- **Transformada de Fourier:** Convierte la señal del dominio del tiempo al dominio de la frecuencia, revelando los componentes frecuenciales y ayudando a comprender el tono y la tonalidad.
- **Coeficientes de Banco de Filtros:** Representan la energía de la señal en diferentes bandas de frecuencia y son utilizados comúnmente en sistemas de reconocimiento de voz.
- **Coeficientes Cepstrales de Frecuencias Mel (MFCCs):** Son ampliamente usados en el procesamiento de audio y voz, representando eficazmente el espectro de potencia a corto plazo de un sonido.

Cada tipo de visualización se genera a través de funciones dedicadas como `plot_signals`, `plot_fft`, `plot_fbank`, y `plot_mfccs`, las cuales son aplicadas a los diccionarios que almacenan los datos correspondientes: `signals`, `fft`, `fbank`, y `mfccs`.

Carga de Datos y Distribución de Clases : Se presenta el proceso de carga de datos desde un archivo CSV, donde se establece el índice basado en los nombres de los archivos de audio. Se realiza una exploración inicial para determinar la longitud de los audios y se calcula la distribución media de las clases, visualizada a través de un gráfico de pastel.

Detección del Umbral de Ruido : En esta parte se aborda la detección y eliminación de espacios sin señal o 'silencios' en los archivos de audio. Se introduce la función `envelope`, que aplica un umbral para limpiar el audio y reducir el ruido de fondo.

Generación de Archivos de Audio Limpios : Esta sección se centra en la preparación de los datos para el modelado, específicamente en la generación de archivos de audio "limpios". Aquí se describe el proceso de reducción del ruido de fondo y la eliminación de secciones silenciosas de los archivos de audio. Se utiliza la función `envelope` previamente descrita para aplicar un umbral y conservar sólo las partes del audio que contienen información relevante. Posteriormente, los audios procesados se almacenan en un directorio 'clean'.

2.2. Notebook 2 - Modelos (model.ipynb)

El segundo cuaderno Jupyter se dedica íntegramente a la configuración, entrenamiento y evaluación de modelos de aprendizaje automático para la clasificación de audio. A continuación se desglosan las secciones principales del cuaderno y las tareas que se llevan a cabo en cada una de ellas.

Clase de Configuración La clase Config es una pieza clave en el proceso de clasificación de audio, funcionando como un almacén central para todas las configuraciones significativas del sistema. Esta encapsula los parámetros críticos para la manipulación y análisis de los datos de audio, como el modo de procesamiento ('conv' para el uso de redes convolucionales y 'time' para redes recurrentes) , y la especificación de los filtros Mel, que son vitales para la extracción de características representativas. Además, establece el tamaño de la ventana para la FFT, lo que determina la resolución en el dominio de la frecuencia, y la tasa de muestreo, que afecta la calidad y el detalle de la señal de audio procesada.

Construcción de los Modelos Se detallan los pasos para la construcción y el entrenamiento de dos tipos de modelos de redes neuronales: uno convolucional (CNN) y otro recurrente (LSTM).

Distribución de Clases en el Conjunto de Datos de Audio Limpio La distribución de las clases en nuestro conjunto de datos, compuesto por **300** muestras de audio originalmente almacenadas en el directorio 'clean', se analiza cuidadosamente para comprender la representación de cada tipo de instrumento. Mediante la generación de **26,404** muestras finales, basadas en la suma de las duraciones de los audios y un marco de tiempo seleccionado de 0.1 segundos, aseguramos una representación diversa y equilibrada en el entrenamiento de los modelos.

Configuración y Entrenamiento del Modelo :

- Preparación de Características
- División Entrenamiento/Prueba (0.7/0.3)
- Definición del Modelo
- Entrenamiento del Modelo
- Callbacks
- Guardado del Modelo
- Evaluación del Modelo

Visualización de Resultados Se incluyen secciones dedicadas a la visualización del entrenamiento del modelo convolucional, mostrando matrices de confusión, gráficos de precisión y pérdida, así como la curva característica de operación del receptor (ROC), que es fundamental para evaluar el rendimiento del modelo en tareas de clasificación.

3. Descripción de la Solución

La solución propuesta para la clasificación de señales de audio implica el uso de dos arquitecturas de aprendizaje profundo diferenciadas: una Red Neuronal Convolucional (CNN) y una Red Neuronal Recurrente (RNN) con células LSTM.

3.1. Arquitectura de la Red Neuronal Convolucional (CNN)

La CNN está configurada para identificar y aprender patrones espaciales en los espectrogramas de audio, que son transformaciones de las señales de audio a representaciones en el dominio de la frecuencia y el tiempo. La estructura de la CNN se describe a continuación:

- Capas convolucionales y de pooling que extraen características de alto nivel de los datos de entrada.
- Normalización por lotes para acelerar la convergencia y mejorar la generalización del modelo.
- Capas de regularización para evitar el sobreajuste mediante el uso de técnicas como la regularización L1 y L2 y la técnica de Dropout.
- Una capa de aplanamiento (Flatten) que convierte las matrices multidimensionales en un vector único que puede ser procesado por las capas densas.
- Capas densas que actúan como un clasificador sobre las características extraídas por las capas convolucionales.

La función de pérdida utilizada es la entropía cruzada categórica, que es adecuada para problemas de clasificación multiclase, y el optimizador es Adam, conocido por su eficacia en diversos problemas de aprendizaje profundo.

3.2. Arquitectura de la Red Neuronal Recurrente (RNN) con LSTM

La RNN con células LSTM está diseñada para capturar dependencias temporales y la secuencialidad en los datos de audio. Este modelo es particularmente adecuado para reconocer patrones a lo largo del tiempo en las señales de audio. La estructura de la RNN se presenta de la siguiente manera:

- Capas LSTM que procesan la secuencia de entrada mientras mantienen un estado interno que refleja el contexto temporal de los datos.
- Normalización por lotes y Dropout que mejoran la estabilidad y la eficacia del entrenamiento.
- Capas de tiempo distribuido (TimeDistributed) que aplican una capa densa a cada paso de tiempo de la secuencia, permitiendo que el modelo aprenda representaciones relevantes en cada punto temporal.
- Una capa de aplanamiento que prepara la salida de la RNN para la clasificación.
- Una capa densa final que utiliza la función de activación softmax para obtener las probabilidades de las distintas clases de audio.

Al igual que en la CNN, se utiliza la entropía cruzada categórica como función de pérdida y el optimizador Adam.

3.3. Preprocesamiento

El preprocesamiento de señales de audio comienza con la función **envelope**, que genera una máscara para distinguir los segmentos relevantes del ruido de fondo mediante un umbral específico. Utilizando **librosa**, cada pista es cargada y remuestreada a 16 kHz. La máscara se aplica a continuación, filtrando las partes donde la media móvil excede dicho umbral. Este paso esencial minimiza elementos superfluos como silencios o ruido estático, optimizando así la calidad del audio para la extracción de características y el entrenamiento del modelo.

3.4. Extracción de Características

La extracción de características es un paso crítico en la clasificación de señales de audio y se lleva a cabo mediante la función build randfeat. Este proceso convierte las señales de audio brutas en un formato que los modelos de aprendizaje automático pueden procesar de manera más eficiente.

La función **BuildRandFeat** construye un conjunto de características extrayendo coeficientes cepstrales de frecuencias mel (MFCCs) de segmentos de audio seleccionados aleatoriamente. Se eligen clases de audio con probabilidades proporcionales a su distribución en el conjunto de datos y, para cada archivo de audio seleccionado, se extrae un segmento aleatorio de longitud fija (**config.step**). Los MFCCs se calculan utilizando un número específico de coeficientes (**config.nfeat**), un número determinado de filtros mel (**config.nfilt**) y un tamaño de ventana para la Transformada Rápida de Fourier (**config.nfft**).

El resultado es un conjunto de valores que representan la energía espectral en el espacio de las frecuencias mel. Estos MFCCs son normalizados para que todos los valores estén dentro del rango de 0 a 1, ajustando los datos entre los valores mínimos y máximos encontrados. Esta normalización es crucial para el rendimiento y la estabilidad en el entrenamiento de modelos de aprendizaje automático.

3.5. Dimensiones de los datos

En cuanto a los datos, **Xtrain** y **Xtest** tienen formas de $(18482, 13, 9, 1)$ y $(7922, 13, 9, 1)$ respectivamente. Estas dimensiones se desglosan de la siguiente manera:

- El primer número (18482 y 7922) representa el número de muestras en los conjuntos de entrenamiento y prueba.
- El segundo número (13) representa el número de coeficientes MFCC (características de audio) extraídos de cada segmento de audio.
- El tercer número (9) corresponde a las dimensiones temporales, es decir, el número de tramas o frames temporales para cada muestra.
- El último número (1) indica un canal adicional, utilizado en modelos convolucionales (*conv*), similar a los canales de color en el procesamiento de imágenes. Para modelos recurrentes (*time*), X se transforma en un tensor 3D, adecuado para el procesamiento secuencial.

Finalmente, **Ytrain** y **Ytest** tienen formas de $(18482, 10)$ y $(7922, 10)$ respectivamente. Aquí, el primer número representa el número de muestras y el segundo indica la codificación "one-hot" de 10 clases posibles. Esta estructura facilita la clasificación en los modelos de aprendizaje automático.

4. Iteraciones y Resultados

En esta sección, documentamos las distintas iteraciones realizadas durante el desarrollo de la solución, enfocándonos en los ajustes de la arquitectura, el preprocesamiento y la extracción de características. Cada iteración busca optimizar el rendimiento de los modelos convolucional y recurrente.

Iteración con Batch Size de 1 (50 Epochs)

Observamos que con un batch size de 1, los modelos se volvieron inestables y no lograron aprender de manera efectiva. Los resultados fueron notablemente peores en comparación con batch sizes mayores.

Modelo	Acc	Loss	Val Acc	Val Loss	Test Acc	Test Loss
Conv	0.926	2.315	0.605	2.929	0.606	2.725
LSTM	0.128	22.991	0.143	2.301	0.153	2.300

Table 1: Resultados con Batch Size de 1

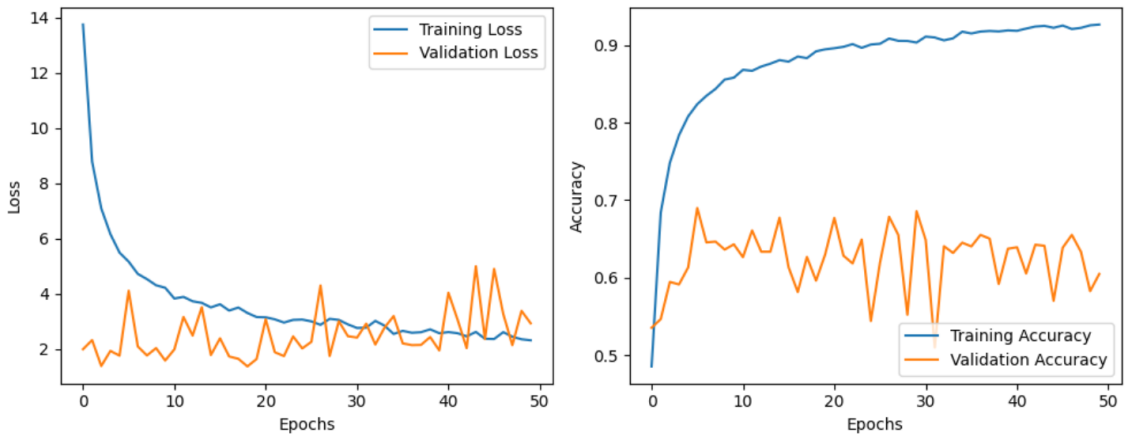

Figure 1: Curva de Pérdida y Precisión para Modelo Conv (Batch Size = 1)

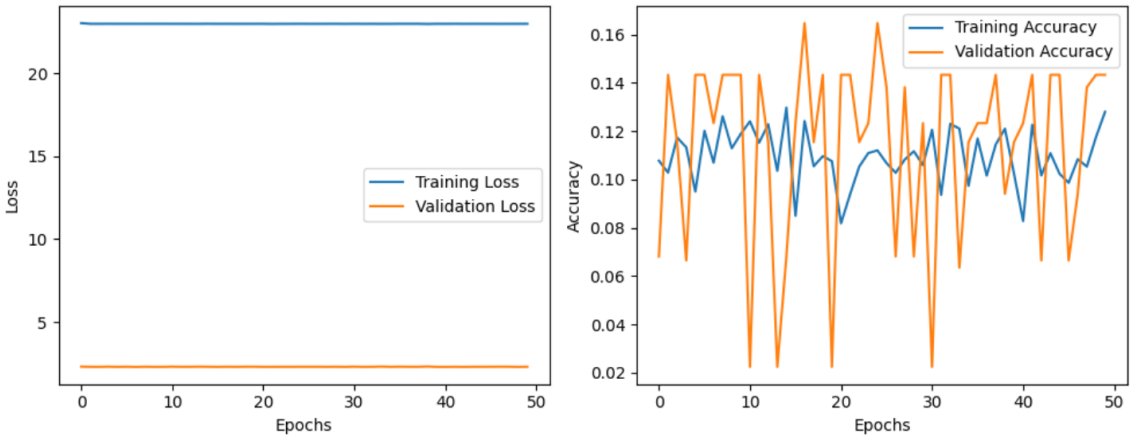

Figure 2: Curva de Pérdida y Precisión para Modelo Recurrente (Batch Size = 1)

Iteración con Batch Size de 32 (50 Epochs)

Con un batch size de 32, los modelos mejoraron significativamente, mostrando un aprendizaje estable y resultados más precisos. Esto se debe a que un batch size más grande proporciona una estimación más precisa del gradiente, pero con menos actualizaciones por época. Los modelos con batch size de 32 lograron un equilibrio entre la eficiencia computacional y la precisión de las actualizaciones de los pesos. A continuación, se presentan los resultados de estas iteraciones en detalle:

Modelo	Acc	Loss	Val Acc	Val Loss	Test Acc	Test Loss
Conv	0.969	0.875	0.957	0.190	0.956	0.206
LSTM	0.960	1.104	0.954	0.196	0.951	0.214
Conv sin reg	0.972	0.755	0.964	0.150	0.962	0.163
LSTM sin reg	0.968	0.850	0.909	0.299	0.909	0.331
Conv sin norm	0.963	1.143	0.956	0.243	0.949	0.283
LSTM sin norm	0.944	1.426	0.933	0.291	0.933	0.269

Table 2: Resultados con Batch Size de 32

Sin Regularización (con Normalización de Batch): Los modelos que se ejecutaron sin regularizadores pero manteniendo la normalización de batch también presentaron un buen rendimiento. Esto sugiere que, para este caso específico, la presencia de la normalización de batch puede compensar parcialmente la ausencia de otros mecanismos de regularización. A pesar de la eliminación de técnicas regulares como el dropout o la regularización L2, la normalización de batch contribuyó a mantener la estabilidad del proceso de aprendizaje y evitar el sobreajuste, lo cual se refleja en los resultados satisfactorios obtenidos.

Sin Normalización Batch: La omisión de la normalización batch resultó en una disminución del rendimiento, lo cual resalta su importancia para la estabilidad del modelo durante el entrenamiento.

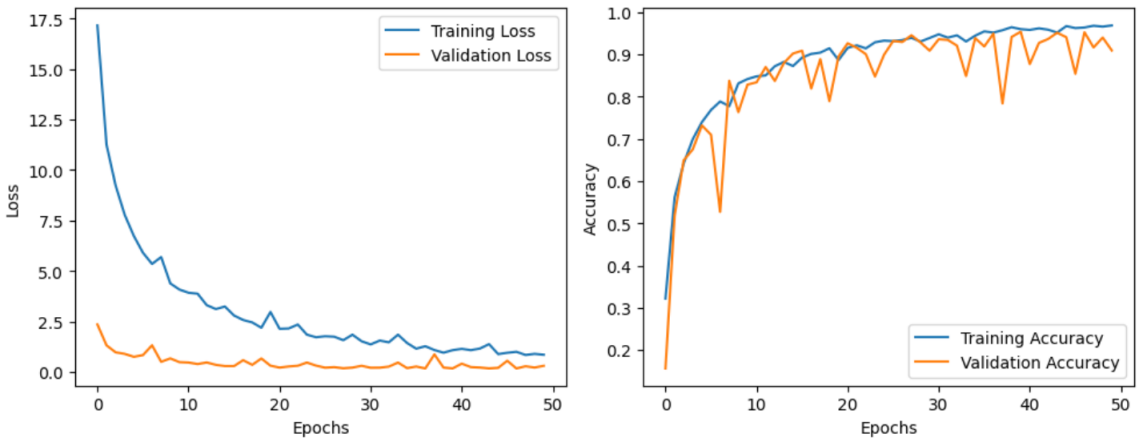

Figure 3: Curva de Pérdida y Precisión para Modelo Conv (Batch Size = 32) sin Regularización

Iteración con Batch Size de 16 (50 Epochs)

Reducir el batch size a 16 mantuvo un alto rendimiento en los modelos, lo que demuestra que un tamaño de lote más pequeño puede ser suficiente para lograr resultados precisos. Esta iteración resalta la eficacia de un tamaño de batch moderado, equilibrando entre la precisión del gradiente y el número de actualizaciones por época.

A continuación, se presentan los resultados detallados de estas iteraciones:

Modelo	Acc	Loss	Val Acc	Val Loss	Test Acc	Test Loss
Conv	0.981	0.588	0.965	0.231	0.964	0.219
LSTM	0.967	0.917	0.952	0.254	0.948	0.251
Conv sin reg	0.979	0.602	0.965	0.165	0.961	0.179
LSTM sin reg	0.968	0.866	0.938	0.222	0.939	0.237
Conv sin norm	0.971	0.960	0.937	0.325	0.936	0.326
LSTM sin norm	0.954	1.356	0.936	0.281	0.937	0.280

Table 3: Resultados con Batch Size de 16

Los modelos sin regularización pero con normalización de batch mostraron un buen rendimiento, lo que sugiere que la normalización de batch puede compensar parcialmente la falta de otros mecanismos de regularización, manteniendo la estabilidad del aprendizaje y evitando el sobreajuste.

Sin embargo, al eliminar la normalización de batch, se observó una disminución en el rendimiento, destacando la importancia de esta técnica para mantener la estabilidad y la eficacia del modelo durante el entrenamiento.

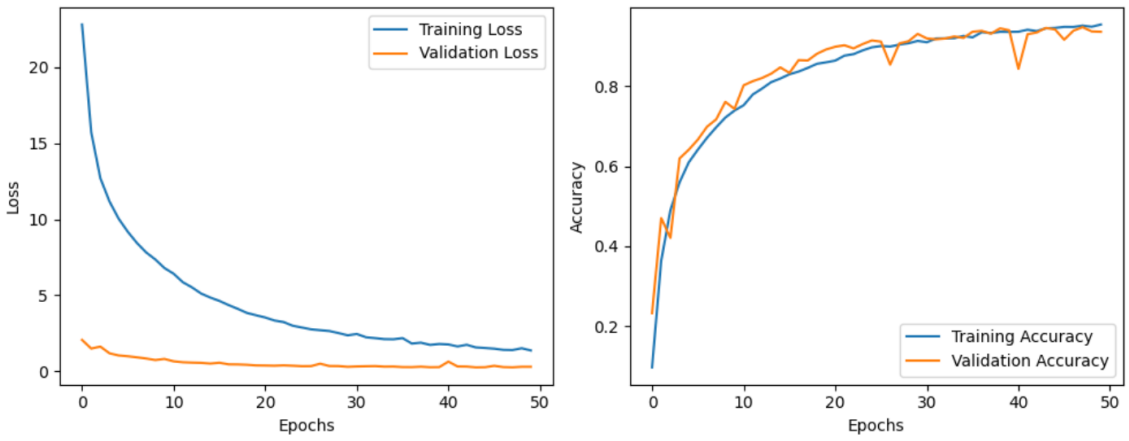

Figure 4: Curva de Pérdida y Precisión para Modelo Recurente (Batch Size = 16) sin normalización de batch

5. Nuestra Mejor Solución

En esta sección, presentamos los resultados alcanzados con lo que consideramos nuestra mejor solución en el proyecto de clasificación de audio. Hemos optado por un modelo de red neuronal convolucional (Conv), configurado con un tamaño de lote (batch size) de 16 y entrenado durante 50 épocas. Para mejorar aún más su rendimiento, hemos implementado técnicas de normalización de lotes (batch normalization) y regularización.

A continuación, se muestran los resultados detallados de este modelo :

Modelo	**Acc**	**Loss**	**Val Acc**	**Val Loss**	**Test Acc**
Conv (Batch Size 16)	0.981	0.588	0.965	0.231	0.964

Table 4: Resultados del modelo Conv con Batch Size de 16

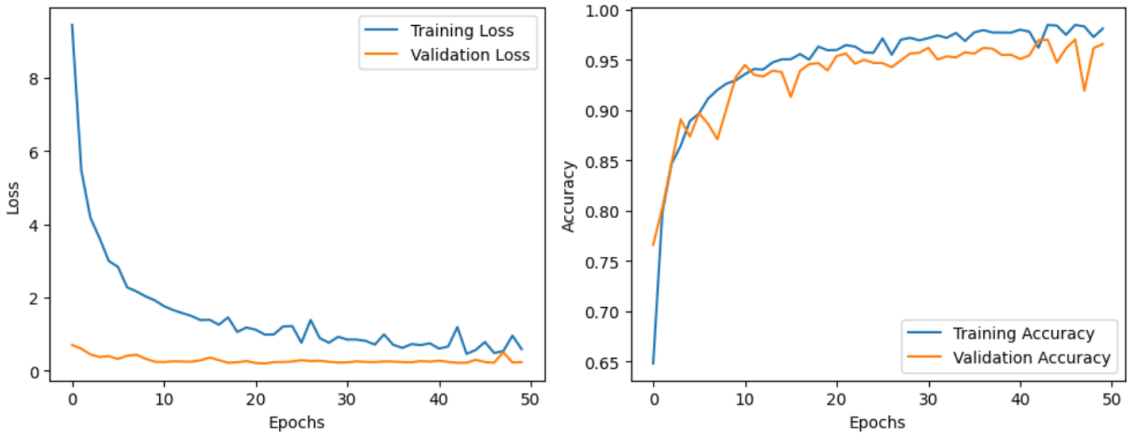

Figure 5: Curva de Pérdida y Precisión para Modelo Conv (Batch Size = 16)

Además, presentamos la curva ROC para evaluar la capacidad discriminativa del modelo en la clasificación multiclase:

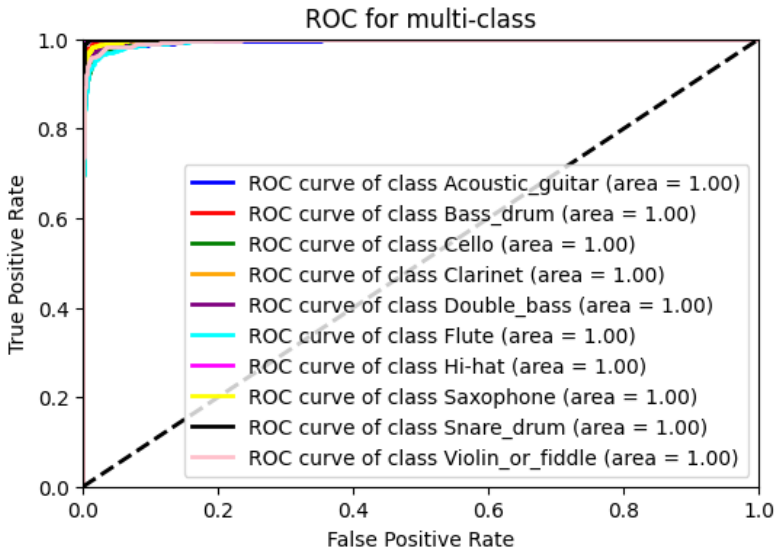

Figure 6: Curva ROC del modelo Conv con Batch Size de 16

Finalmente, la matriz de confusión adjunta ofrece una visión detallada de la precisión del modelo en la clasificación de las distintas categorías:

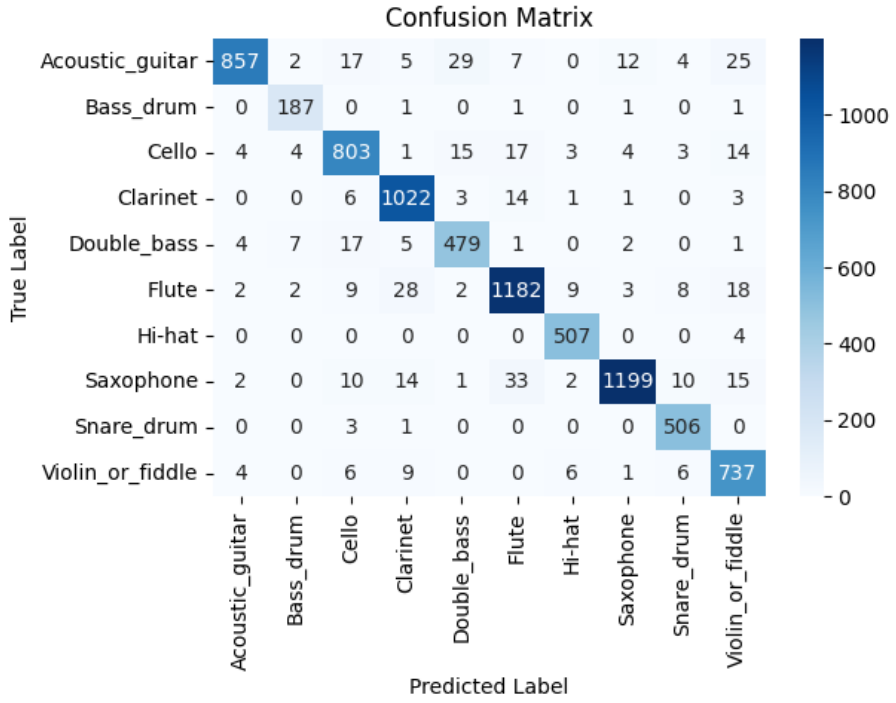

Figure 7: Matriz de confusión para el modelo Conv con Batch Size de 16

Estos resultados y visualizaciones evidencian la eficacia del modelo Conv con un Batch Size de 16, mostrando un alto grado de precisión y capacidad para generalizar bien en datos no vistos.

6. Conclusión

En conclusión, este proyecto nos ha permitido adentrarnos en el mundo de la clasificación de señales de audio mediante redes neuronales, explorando tanto las convolucionales como las recurrentes. A lo largo del proyecto, hemos aprendido sobre la importancia de ajustar cuidadosamente los hiperparámetros y las técnicas de preprocesamiento. Aunque nuestro modelo Conv con un tamaño de lote de 16 ha mostrado resultados prometedores, comprendemos que todavía hay mucho espacio para el crecimiento y la mejora.

Las visualizaciones, como las curvas ROC y las matrices de confusión, no solo han enriquecido nuestro entendimiento del modelo sino también han subrayado la profundidad y complejidad del campo del procesamiento de audio. Mirando hacia el futuro, vemos muchas oportunidades para expandir nuestro conocimiento y habilidades, experimentando con más datos, diversas categorías de audio y arquitecturas de redes más complejas.

Este proyecto ha sido una valiosa experiencia de aprendizaje, ofreciéndonos una base sólida sobre la cual podemos construir en nuestros futuros estudios y proyectos en el ámbito del aprendizaje automático y la inteligencia artificial.