Detección acústica de especies

Dante Bermúdez Marbán
bermudezmarbandante@gmail.com

Resumen

Rainforest Connection y Kaggle presentan el desafío de crear un modelo que estime la probabilidad de que una especie esté presente en un audio. Debido al desbalanceo entre verdaderos positivos y falsos positivos, se aplicaron tres técnicas de acrecentado de datos. Se probaron los espectrogramas con escala lineal y los que tienen escala de Mel como representaciones de los audios para un modelo convolucional. Se utilizó la métrica *Label Ranking Average Precision* para medir los desempeños, resultando con un valor de 0.7 usando el espectrograma con la escala de Mel.

1. Introducción

El cambio climático es uno de los problemas más grandes que enfrenta la humanidad. Este fenómeno se puede describir como el aumento de temperatura causado por el exceso de dióxido de carbono que generamos, lo cual crea el llamado "efecto invernadero". Si bien dicho efecto es natural y se ha presentado en el planeta por ciclos, la presencia humana lo ha acelerado. Un ejemplo de acción humana que ha contribuido a este fenómeno es la alarmante tala de árboles que se hace en selvas, lo cual es un problema ya que éstas nos ayudan a absorber gran parte del dióxido del carbono [1].

Dicho lo anterior, es obvio que es necesario tomar acciones para dejar de contribuir tanto al efecto invernadero, por lo que es necesario buscar indicadores que nos señalen el impacto que tiene el cambio climático en los hábitats. Uno bueno podría ser la presencia de animales endémicas en la selva, el problema es que resulta muy difícil ver a las especies, por lo que Rainforest Connection (RFCx), una organización, ha creado un sistema de monitoreo acústico, el cual tiene distintas finalidades, tales como prevenir la tala y caza ilegal, así como el que nos compete en este documento: detectar la presencia de alguna especie por medio de algún audio [2].

2. Descripción del problema

Rainforest Connection ha proporcionado a Kaggle un conjunto de datos que consta principalmente de audios donde se escuchan a distintas especies de ranas y pájaros de la selva. La finalidad es crear un sistema que permita identificar la presencia de dichas especies [3].

Para cada audio, se debe indicar la probabilidad de presencia, correspondiente a cada una de las 24 especies (una lista de las especies se puede consultar en [4]). Debido a que hay algunos audios en los que más de una especie está presente, este problema es de **clasificación multietiqueta**

3. Descripción de los datos

El conjunto de datos proporciona audios con verdaderos positivos y falsos positivos con la finalidad de robustecer el entrenamiento. Se incluyen dos archivos CSVs con las anotaciones: uno para verdaderos positivos, y otro para falsos positivos, el cual, para cada renglón, incluyen la siguiente información:

- **recording_id**: identificador del archivo de audio
- **species_id**: Un entero que va desde 0 hasta 23, el cual identifica a la especie.
- **songtype_id**: Identificador del tipo de sonido que el animal emite.
- **t_min**: Inicio de la anotación en donde se escuchó a la especie, en segundos
- **t_max**: Fin de la anotación en donde se escuchó a la especie, en segundos.
- **f_min**: Frecuencia más baja de la anotación en donde se escuchó a la especie, en hercios.
- **f_max**: Frecuencia más alta de la anotación en donde se escuchó a la especie, en hercios.

Se concatenaron los dos archivos, añadiendo las siguientes columnas

- **is_tp**: Indica si la anotación es un verdadero positivo.
- **duration**: La diferencia de **t_max** - **t_min**.
- **bandwidth**: La diferencia de **f_max** - **f_min**.

Al querer pivotar la tabla, de tal forma que cada renglón representara un audio, con 24 columnas (una para cada especie) con -1 si no aparece, 0 si es falso positivo y 1 si es verdadero positivo, se encontraron que hay algunos audios donde se anotó a la misma especie dos veces pero con tiempos y frecuencias diferentes, incluso la bandera que indica si es verdadero positivo o no eran diferentes. Un ejemplo de esto se puede ver en la figura 1, donde la especie 17 se repite dos veces, con intervalos diferentes y cuyos valores de `is_tp` son diferentes. En ete caso en particular, como los intervalos no se traslapan, es perfectamente plausible estas anotaciones, es decir, no necesariamente representan un error.

	recording_id	species_id	songtype_id	t_min	f_min	t_max	f_max	is_tp
56	0c48ed342	17	1	50.0320	1312.50	56.4853	3937.50	True
1617	0c48ed342	2	1	53.6640	468.75	55.4400	3000.00	False
1618	0c48ed342	19	1	47.4933	281.25	49.0453	2812.50	False
1619	0c48ed342	17	4	43.5413	1312.50	46.0747	7406.25	False

FIGURA 1: Ejemplo de anotaciones duplicadas para la especie 17 en el mismo audio

Por simplicidad, se busca que para cada audio, solamente exista una anotación por especie, por lo que si una especie tenía más de una anotación, se promediaban las anotaciones (los tiempos y frecuencias), siempre y cuando las anotaciones tuvieran el mismo valor para la variable de verdadero positivo. En el caso que hubieran valores diferentes para dicha variable, solamente se hacía la agregación para las anotaciones que tuvieran el valor de verdadero positivo.

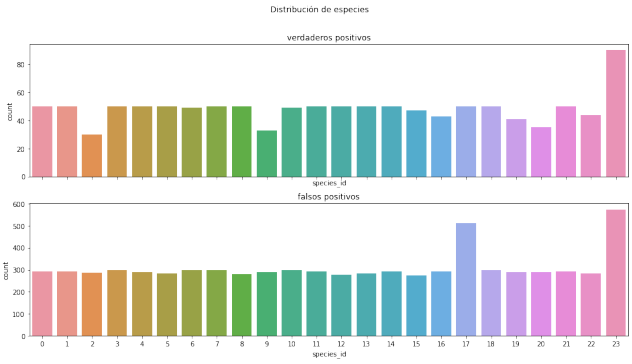

FIGURA 2: Distribución de las especies (verdaderos positivos arriba, falsos positivos abajo).

En la figura 2 podemos ver una gráfica de barras que representa la frecuencia de cada especie en el conjunto de datos de entrenamiento. Vemos que la especie 23 es la que predomina tanto en los verdaderos y falsos positivos. También podemos ver que, las otras especies están aproximadamente balanceadas en los falsos positivos (a excepción de la especie 23), mientras que en los verdaderos positivos, hay algunas especies con un conteo considerablemente menor.

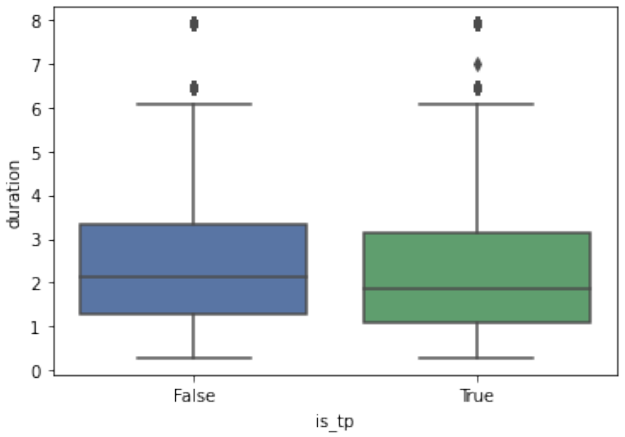

FIGURA 3: Diagramas de caja para la duración por tipo de positivo

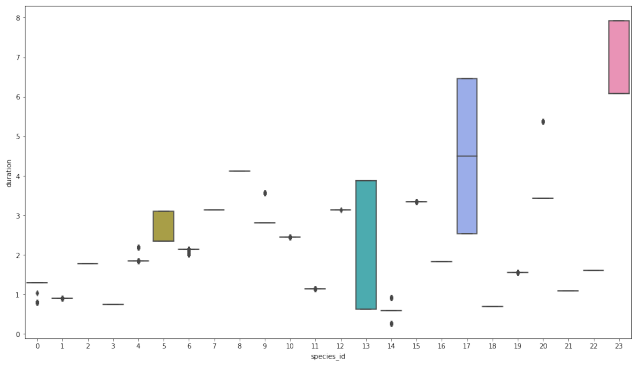

FIGURA 4: Diagramas de caja para la duración por especie

En la figura 3 podemos ver diagramas de caja para la duración de las anotaciones según si son verdaderos positivos o falsos positivos. La conclusión que se puede obtener es que las duraciones son muy parecidas, por lo que la duración no distingue entre los verdaderos y falsos positivos. Lo que si distingue la duración son el tipo de especies, tal como se puede ver en la figura 4, en donde cada especie tiene

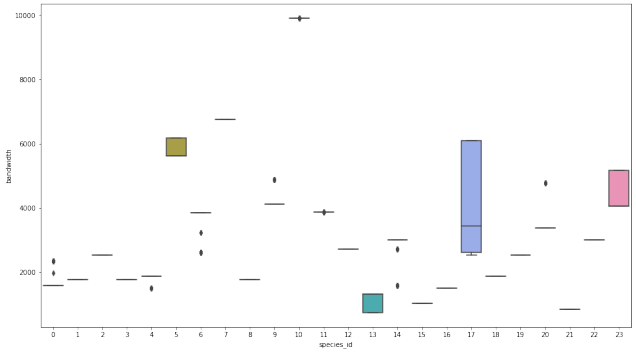

FIGURA 5: Diagramas de caja para el ancho de banda por especie

una duración en concreto. Una conclusión similar se puede obtener del ancho de banda al inspeccionar la figura 5.

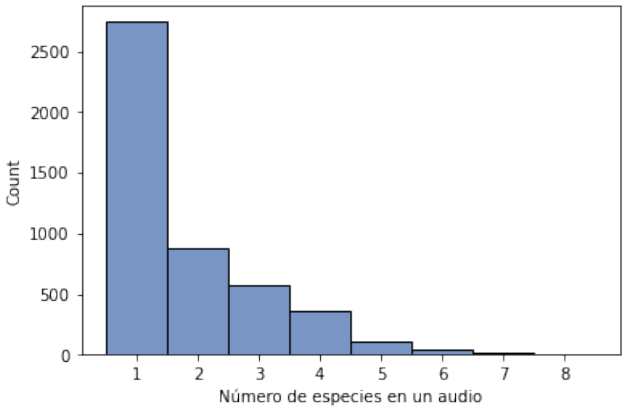

FIGURA 6: Histograma de especies por audio

Como se mencionó, es un problema de clasificación multietiqueta, por lo que más de una especie puede estar presente en el audio, lo cual se ejemplifica en el histograma de la figura 6. La mayoría de casos solamente hay una especie, pero hay casos de hasta 8 especies en un audio.

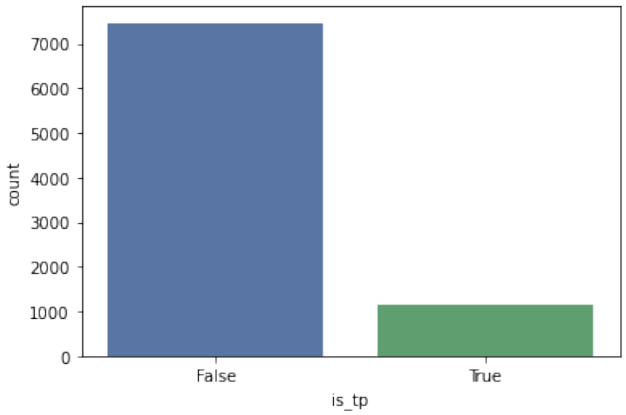

FIGURA 7: Distribución entre verdaderos y falsos positivos

Un aspecto a considerar es la proporción entre verdaderos positivos y falsos positivos, la cual se puede ver en la figura 7 un claro desbalanceo, con 7459 falsos positivos y 1161 verdaderos positivos.

Respecto a los audios, se cuentan con 4727 de entrenamiento y 1992 de predicción. Todos los audios duran un minuto. En la figura 8 se puede ver tanto la gráfica de forma de onda como el espectrograma, con las anotaciones correspondientes (líneas verticales en la gráfica de forma de onda, cuadros verdes en el espectrograma) . En este ejemplo también se puede ver que hay traslapes.

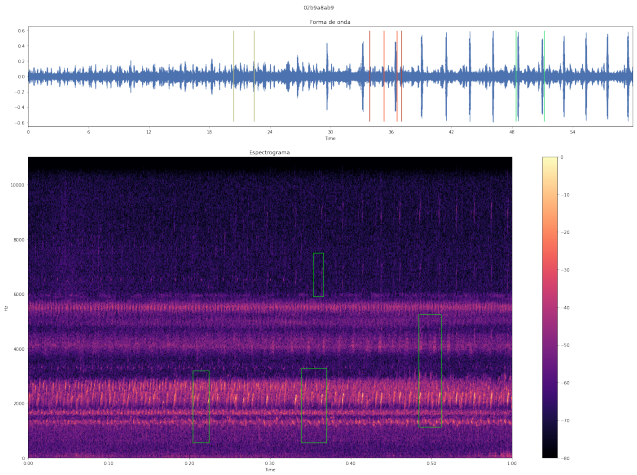

FIGURA 8: Ejemplo de audio

4. Metodología

4.1. Acrecentamiento de datos

Debido al desbalanceo expuesto en la figura 7 de falsos positivos frente a verdaderos positivos, lo primero que se hizo fue hacer un acrecentamiento de datos a los audios que tuvieran mayor número de verdaderos positivos que falsos positivos. De los 4727 audios de entrenamiento, 777 cumplían con esta característica, y a estos audios se les aplicó tres técnicas

- Agregar ruido gaussiano: consiste en agregar ruido blanco al audio. Si consideramos que σ es la desviación estándar de un audio, entonces

  $$\text{Nuevo audio} = \text{audio} + \alpha\epsilon$$

  Donde $\epsilon \sim \mathcal{N}(0, \sigma)$ y α es una constante que se eligió también de manera aleatoria tal que $\alpha \sim \mathcal{U}(0, 1)$.

- Desplazar audio: Consiste en desplazar la onda en el tiempo, ya sea por la izquierda (atrasar el audio) o por la derehca (adelantar). Aquellos elementos que lleguen a tiempos posteriores del minuto del audio, o inferiores al inicio del audio (segundo cero), se ingresan del otro lado.

  En este caso en particular, la dirección y cantidad de segundos del desplazamiento era determinado también de manera aleatoria, procurando que las anotaciones desplazadas no fueran recortadas.

- Cambio de tono: Consiste en mover la onda una cantidad de semitonos. El nuevo audio tiene la misma duración, pero se escuchará mas agudo o más grave. En este caso, el número de semitonos era determinado de manera uniforme entre $\{-2, -1, 1, 2\}$.

Los audios de entrenamiento y de prueba tenían diferentes frecuencias de muestreo (*sampling rate*), por lo que en este

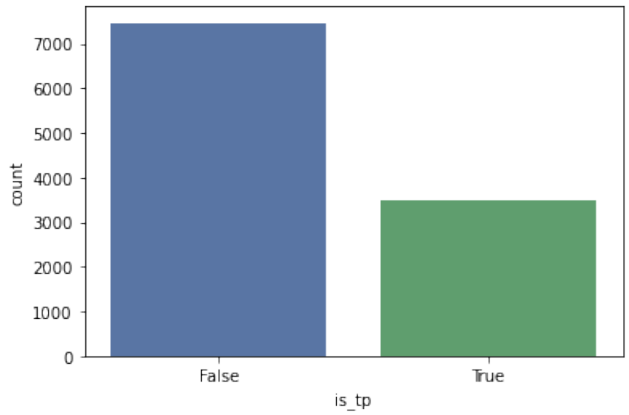

FIGURA 9: Distribución entre verdaderos y falsos positivos después del acrecentamiento de datos

proceso, se aprovechó para leer todos los audios y remuestrearlos a una misma frecuencia de muestreo.

Después de aplicar estas técnicas, se obtuvo un total de 7034 audios, donde se mantuvieron los 7459 falsos positivos y se aumentaron los verdaderos positivos a 3483.

4.2. Espectrograma

Para esta tarea de clasificación multietiqueta de audios, se optó por diseñar una arquitectura orientada a imágenes, en este caso, estamos hablando de espectrogramas.

Se contemplaron dos tipos de espectrogramas:

- Espectrograma: Basado en transformadas rápidas de Fourier. Para este caso se optó por hacer 400 transformadas, creando una imagen con 201 bins [1]. El ejemplo de la figura 8 corresponde a uno de estos casos.
- Mel: Corresponde a obtener el espectrograma descrito en el punto anterior, y luego transformar las frecuencias en frecuencias de mel[2]. En [5], se obtuvieron buenos resultados con esta representación, y para este caso, se optó por usar 64 filtros de mel.

4.3. Modelo

Primeramente, se definió un bloque convolucional, el cual se define por el número de canales de entrada y el número de filtros deseados (canales de salida), y consiste de una capa convolucional 2D, con un tamaño de 3×3 para el kernel, luego se pasa por la función de activación ReLU, después se pasa por una capa de *max pooling* 2d, cuya ventana de muestreo es de 2×2, y finalmente se aplica deserción (*dropout*), con una probabilidad de 0.5 en donde se desactivan las neuronas. Este bloque se puede apreciar en la figura 10.

[1]El número de bins se obtiene como $\lfloor n_{fft}/2 \rfloor + 1$

[2]La escala de mel se caracteriza por ir cambiando de manera perceptual

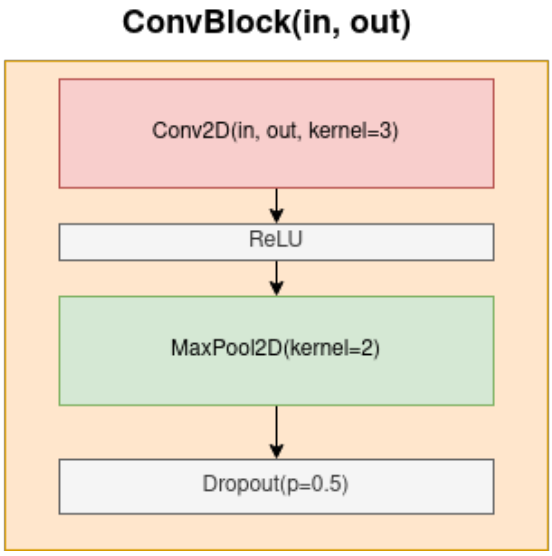

FIGURA 10: Definición de bloque convolucional

Una vez definido el bloque convolucional, podemos definir la arquitectura de la red, la cual consiste en apilar tres de estos bloques, usando 20 filtros para el primero, 40 para el segundo y 60 para el tercero, y luego viene la parte del clasificador, donde se hace un aplanado de los mapas de características, se pasa por una capa completamente conectada con 256 neuronas ocultas, y finalmente la última capa con 24 neuronas, que corresponde al número de especies que tenemos. La definición de esta red se puede ver en la figura 11. Nótese que en la primera capa completamente conectada, se deja como variable las dimensiones de la imagen que se obtienen tras las convoluciones. Esto es porque varía según el tipo de espectrograma que se use.

4.4. Esquema de entrenamiento

Para el conjunto con el que se tiene etiquetas, se dividió de manera aleatoria en 80 % entrenamiento y 20 % prueba.

Como función de pérdida, se utilizó la entropía cruzada binaria, sin embargo, esta función se personalizó de tal manera que solamente contribuyeran aquellas especies en el audio que aparecieran como falso positivo o verdadero positivo. La ventaja de modelar el problema así es que se hace uso de la información de los falsos positivos para robustecer los casos cuando la especie no está presente y se evita el desbalanceo de etiquetas para aquellas especies que no tienen anotaciones (la cual es muchísima más grande que la de verdaderos y falsos positivos juntos).

Como métrica para medir el desempeño, se utilizó la métrica *Label Ranking Average Precision*, la cual es la que se pidió utilizar en el desafío de Kaggle, y queda definida como

$$\text{LRAP}(y, \hat{f}) = \frac{1}{n_{\text{muestras}}} \sum_{i=1}^{n_{\text{muestras}}} \frac{1}{||y_i||_0} \sum_{j:y_{ij}=1} \frac{|\mathcal{L}_{ij}|}{\text{rank}_{ij}} \quad (1)$$

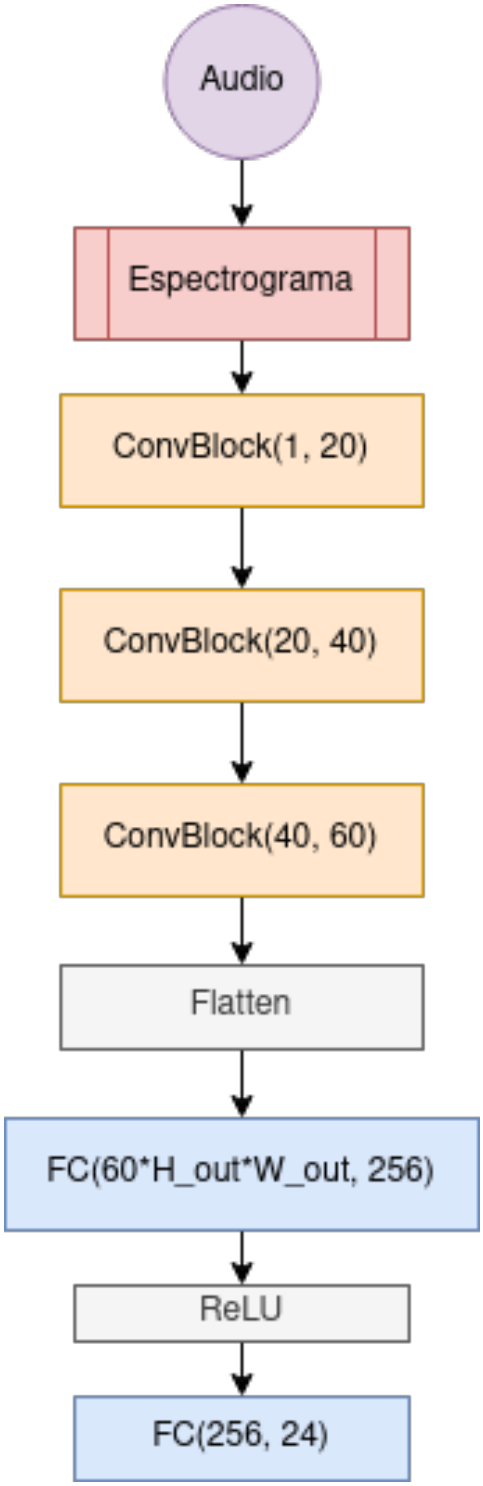

FIGURA 11: Definición del modelo

donde $y \in \{0,1\}^{n_{\text{muestras}} \times n_{\text{etiquetas}}}$, $\hat{f} \in \mathbb{R}^{n_{\text{muestras}} \times n_{\text{etiquetas}}}$, $\mathcal{L}_{ij} = \{k : y_{ik} = 1, \hat{f}_{ik} \geq \hat{f}_{ij}\}$, $\text{rank}_{ij} = |\{k : \hat{f}_{ik} \geq \hat{f}_{ij}\}|$, y $||\cdot||_0$ es la "norma" ℓ_0[3] que cuenta el número de elementos distintos de cero. La pregunta que responde esta métrica es: para cada anotación, que fracción de mis predicciones que están mejor *rankeados* son verdaderos [6].

El modelo se entrenó por 5 épocas, con un tamaño de lote igual a 4 y se usó el descenso del gradiente estocástico como método de optimización.

[3]No se considera norma porque no cumple con la desigualdad del triángulo

LRAP	Train	Test
Espectrograma	0.657	0.630
Mel	0.723	0.701

TABLA 1: Resultados

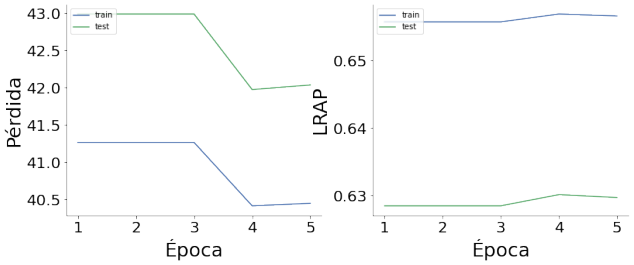

FIGURA 12: Desempeño del modelo usando espectrogramas

5. Resultados

En la tabla 1 se pueden ver los resultados de las dos representaciones tiempo-frecuencia propuestos, utilizando la métrica LRAP como comparativa

En las figuras 12 y 13 podemos ver la evolución del desempeño para cada enfoque. se puede ver al lado izquierdo, la gráfica de la pérdida y del lado derecho, de la métrica LRAP.

Kaggle permite subir intentos de predicción, a pesar de que la competencia ya haya terminado, por lo que se utilizó el modelo con la representación de Mel para realizar las predicciones en el conjunto de audios que no tienen anotaciones. El score final fue de 0.337, tal como se ve en la figura 14.

gulo

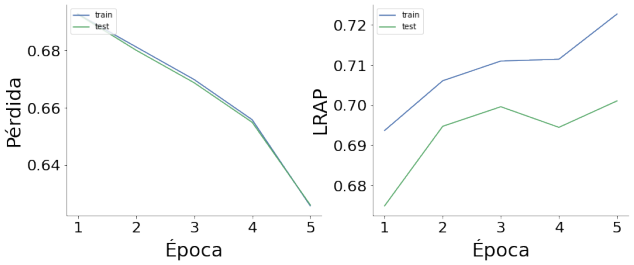

FIGURA 13: Desempeño del modelo usando espectrogramas en escala de mel

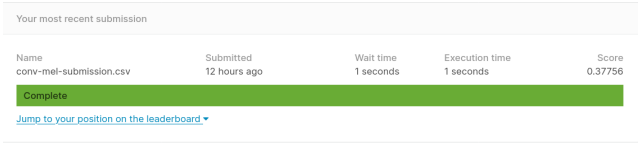

FIGURA 14: Puntuación final de Kaggle

6. Conclusiones

De la tabla 1 podemos ver que, tal como sugieren en [5], el espectrograma con la escala de Mel otorga mejores resultados que la otra representación, aparte de que la imagen resultante es de menores dimensiones, lo que se traduce en menos tiempo de procesamiento y entrenamiento.

En la figura 13, vemos que es posible mejorar el modelo si se entrena por más épocas, ya que no se presenta evidencia de sobreajuste.

Respecto a la puntuación final otorgada por Kaggle, el resultado no fue el esperado. Una posible explicación es que los audios para predicción fueran anotados de manera diferente, tal como se platica en algunos foros de Kaggle referente a la competencia. No obstante, eso no detuvo a un equipo para obtener el primer lugar con una puntuación de 0.98.

El problema más grande al que se enfrentó en la realización de este proyecto, eran los largos tiempos que toma trabajar con audios: las técnicas de aumentado de datos, así como remuestrear todos los audios de entrenamiento a un misma frecuencia de muestreo, tomó alrededor de cuatro horas. Asimismo, entrenar el modelo por cinco épocas toma aproximadamente dos horas, sin embargo, a la hora de hacer las predicciones, no tomó más de 15 minutos, por lo que es posible que el proceso se esté ralentizando a la hora de crear el vector de etiquetas.

Dicho lo anterior, una forma notable de mejorar el proyecto es haciendo más eficiente la creación del vector de etiquetas para un entrenamiento más rápido (entre más rápido uno puede experimentar, más rápido se pueden llegar a mejores resultados).

Una vez hecho lo anterior, o incluso, ignorando lo anterior (suponiendo que el tiempo no es un problema), hay dos formas de expandir el proyecto:

- Explorando otras representaciones tiempo-frecuencia, por ejemplo, los coeficientes ceptrales en las frecuencias de Mel (MFCC)
- Usar arquitecturas del estado del arte, por ejemplo, ResNet18.

El código de este proyecto se puede consultar en el repositorio [7].

Referencias

[1] Jessica Pink. *3 ways climate change affects tropical rainforests*. https://www.conservation.org/blog/3-ways-climate-change-affects-tropical-rainforests/. 26 de junio del 2018.

[2] Rainforest Connection. *Our work*. https://rfcx.org/our_work. 2021.

[3] *Rainforest Connection*. *Rainforest Connection Species Audio Detection*. https://www.kaggle.com/c/rfcx-species-audio-detection. Recuperado el 11-06-2021.

[4] *Jack L. (anfitirión de la competencia)*. *List of Species*. https://www.kaggle.com/c/rfcx-species-audio-detection/discussion/238216. Recuperado el 11-06-2021.

[5] Muhammad Huzaifah. "Comparison of Time-Frequency Representations for Environmental Sound Classification using Convolutional Neural Networks". En: *CoRR* abs/1706.07156 (2017). arXiv: 1706.07156. URL: http://arxiv.org/abs/1706.07156.

[6] Scikit learn. *Label ranking average precision*. https://scikit-learn.org/stable/modules/model_evaluation.html#label-ranking-average-precision. 2021.

[7] D Bermudez. *Repositorio del proyecto*. 2021. URL: %5Curl % 7Bhttps : / / github . com / DanteBM / Aprendizaje-Profundo/tree/main/proyecto%7D (visitado 12-06-2021).