

Trabajo Final

DeepMusicGenreClassifier

3 de julio de 2025

Procesamiento de Señales, Audio y Habla

Integrante	LU	Correo electrónico
Aaron Bernal Huanca	815/22	aaronbernal28@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
<http://www.exactas.uba.ar>

Resumen

Este informe aborda la clasificación de géneros musicales utilizando el dataset GTZAN (10 géneros, 100 canciones de 30 segundos cada uno). El objetivo es analizar y clasificar la música empleando diversas arquitecturas de redes neuronales. Se implementan métodos que incluyen el procesamiento secuencial RNNs, el análisis de espectrogramas (Mel, MFCC, CQT) con CNNs (1D), y la utilización de redes neuronales preentrenadas para la extracción de features relevantes (transfer learning) como Wav2vec2 y EnCodecMAE. Como criterios de evaluación se emplean la entropía cruzada y la precisión (accuracy) para determinar cuál de estas arquitecturas ofrece un mejor desempeño en la tarea de clasificación.

Dataset: <https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification/data>

Repositorio: <https://github.com/aaronbernal28/DeepMusicGenreClassifier>

1. Introducción

Se dispone de un conjunto de 100 canciones, cada una con una duración de 30 segundos, distribuidas equitativamente en 10 géneros musicales distintos. A partir de estas canciones se extraen distintas representaciones (embeddings o espectrogramas), que en adelante denominaremos features por simplicidad.

Los datos fueron divididos de manera estratificada en tres subconjuntos: un 10 % para evaluación final, y el 90 % restante se separó en 80 % para entrenamiento y 20 % para validación. La misma se realizó independientemente de las features.

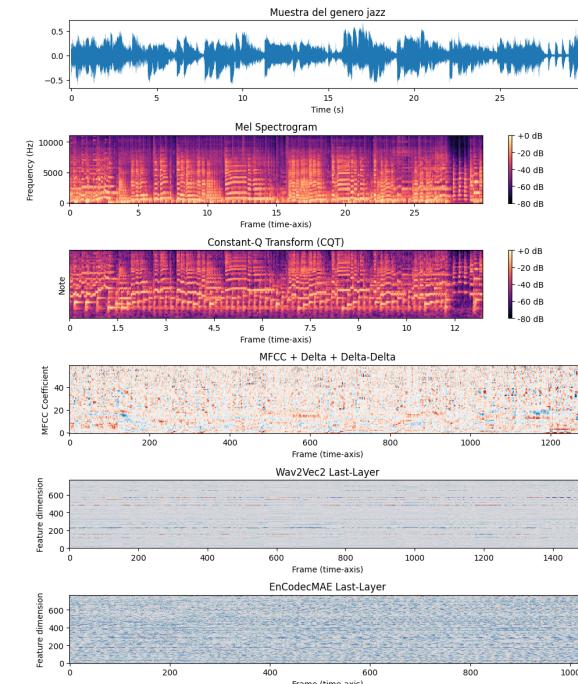


Figura 1: Distintas representaciones de una muestra del género jazz

La representación CQT está diseñada específicamente para capturar estructuras musicales como notas, acordes y sus armónicos ([2], [3]). Está orientada al análisis de audio musical; véase la Figura 1 para una muestra del género jazz.

Información adicional que puede ser de interés:

feature	dimension
Mel	80
MFCC	60
CQT	84
Wav2vec2	768
EnCodecMAE	768

Tabla 1: Dimensión de cada feature por frame o tiempo. 10 ms hop length y 25 ms de ventana donde corresponda, el resto de parámetros por default

Cabe aclarar que la extracción de las features de EnCodecMAE se realizó con los primeros 15 segundos (no 30s original) por cuestiones de memoria.

Finalmente, aunque la arquitectura se mantuvo idéntica para todas las representaciones, las diferencias en tamaño y consumo de memoria obligaron a ajustar algunos hiperparámetros. Por ejemplo, si la matriz CQT (84, 1290) permitía un batch size de 64, el embedding de Wav2Vec2 (1496, 768) requeriría batch size 28, lo cual implicaba los respectivos learning rates. De igual modo, el número de épocas se fijó de manera dinámica, deteniendo el entrenamiento cuando la pérdida de validación se estabilizaba o empezaba a alejarse significativamente de la del entrenamiento.

2. Desarrollo

2.1. Random Forest

Este modelo, para cada feature, toma la media respecto la secuencia o tiempo, es decir, pasamos de $(T, \text{feature_dim})$ a $(, \text{feature_dim})$. Seguidamente, se entrena clásico un Random Forest con 100 ensambles.

Característica	Accuracy
Mel	0.27
MFCC	0.14
CQT	0.32
Wav2vec2	0.66
EnCodecMAE	0.79

Tabla 2: Performance de cada feature tomando la media

EnCodecMAE da buenos resultados comparable con los siguientes modelos y nos sirve como referencia (Tabla 2). Considerando que se tiene vectores de dimensión fija $\text{feature.dim}_{\text{EnCodecMAE}} = 768$ y buena performance, proyectamos a dos dimensiones con la ayuda de UMAP entender como se distribuyen las clases.

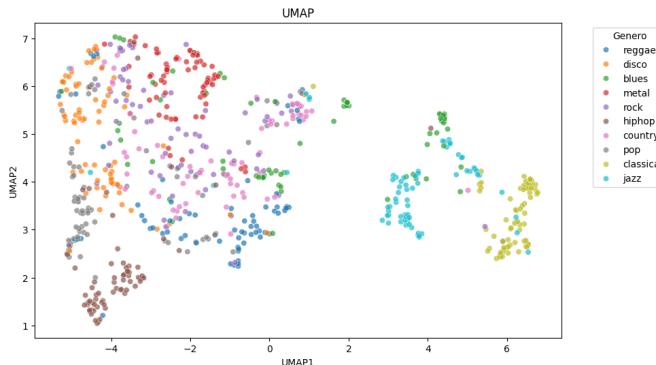


Figura 2: Proyección UMAP a 2D

Existe una clara separación no perfecta para algunas clases como se puede observar en 2, por ejemplo entre *hiphop*, *jazz* y *classical*. Además, *classical* esta más alejado al resto. Esto se corresponde con el propósito de las representaciones, en particular, que sean discriminativas y capturar relaciones semánticas.

2.2. Multilayer perceptron (MLP)

Presentamos un aproach naive donde extraemos las primeras 2049 frecuencias de la serie de Fourier de cada muestra (Tabla 3).

Layer Type	Input Size	Output Size
Input Layer	2049	—
Hidden Block 1	2049	1024
... 2	1024	512
... 3	512	256
... 4	256	128
... 5	128	64
Output Layer	64	10

Tabla 3: Arquitectura de la red con activacion Gelu, dropout 0.5 y batchnorm 1D. Cantidad de parametros: 2,801,098

Como se puede ver en la Figura 3, el modelo sobreajusta rápidamente y la loss de los datos de test se estanca. Luego de 100 épocas alcanza un accuracy del 0.49.

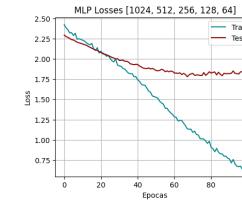


Figura 3: Curva de aprendizaje de la red

2.3. Redes recurrentes

En esta sección se describen las dos arquitecturas que obtuvieron los mejores desempeños. En todos los casos, se considera a las representaciones Mel, CQT y MFCC como secuencias de vectores de características con dimensiones especificadas en la Tabla 1.

El primero modelo emplea LSTM y el segundo una variación de la misma: GRU. Ambos tienen como output un estado oculto de dimensión $(2 * \text{hidden_size}, T)$. Luego, se aplica un mecanismo simple de atención para finalmente obtener una salida de la dimensión de la cantidad de clases, en nuestro caso, 10. Véase Figura 4.

2.3.1. LSTM

A continuación se presenta un esquema detallado de la arquitectura basada en LSTM:

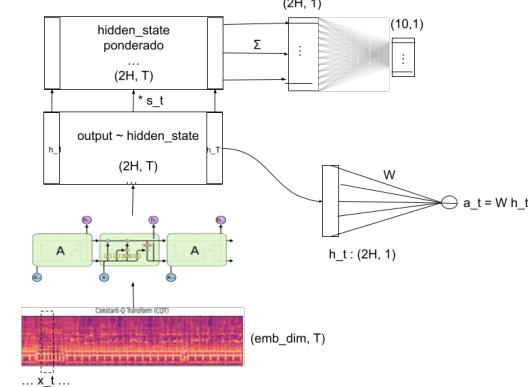


Figura 4: Esquema del modelo con LSTM bidireccional donde $(s_1, \dots, s_T) = \text{softmax}(a_1, \dots, a_T)$ y $H = 16$

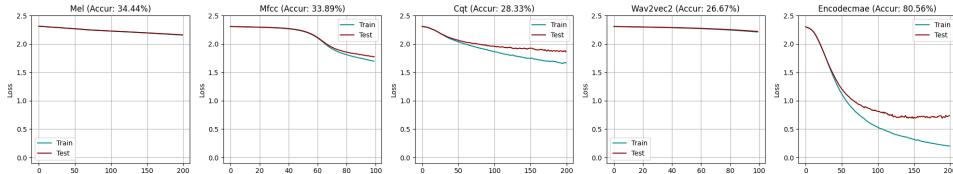


Figura 5: Curvas de aprendizaje del LSTM

A partir de las curvas de pérdida y accuracy obtenidas para cada tipo de representación 5, podemos afirmar que EnCodecmae alcanza una performance buena y se estanca. El resto de las features tienen un desempeño pobre con un aprendizaje muy lento.

2.3.2. GRU

La arquitectura es igual que en esquema 4, sin embargo, se reemplaza la unidad de recurrencia LSTM por GRU. Ademas el *hidden_size* pasa a ser 32.

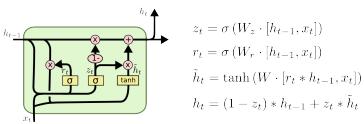


Figura 6: Esquema de GRU

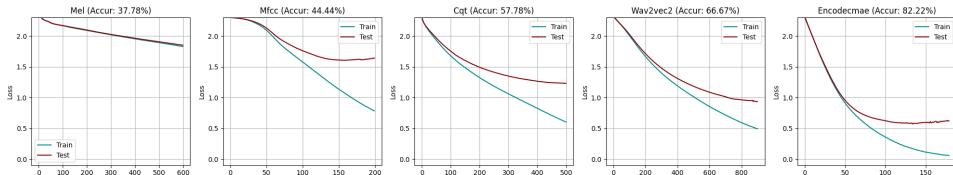


Figura 7: Curvas de aprendizaje de GRU

Como se puede observar en Figura 7, el rendimiento mejora de forma constante al avanzar desde los spectrogramas tradicionales hacia los embeddings auto-supervisados. Con Mel obtenemos un 37 % de aciertos, MFCC escala al 44 % y CQT alcanza el 58 %. Al incorporar representaciones de Wav2Vec2, la precisión sube al 67 %, mientras que EnCodecMAE sobresale con más del 82 %. Esta brecha entre los spectrogramas y las representaciones se puede deber a su capacidad de capturar rasgos musicales de alto nivel. Cabe aclarar que se experimento añadir métodos de regularización como dropout pero no produjeron mejoras adicionales.

2.3.3. Redes convolucionales 2D

MEL y CQT son las únicas representaciones que muestran diversos patrones que valen la pena explorar con redes convolucionales 2D. Para ello, figura 8a, proponemos una arquitectura sencilla con:

- Convoluciones con kernel de 3×3 , seguido de max-pooling 2×2 .
- Progresión de filtros de 32 a 64 canales.
- Average global pooling para reducir la dimensión espacial a 64.
- Capa fully connected final de $64 \rightarrow 10$ (número de clases).

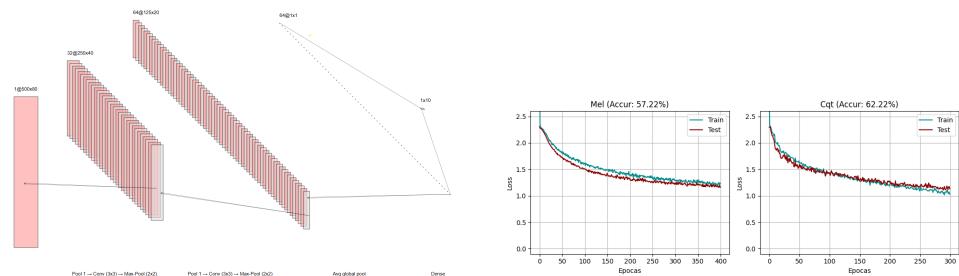


Figura 8: (a) Arquitectura de la CNN2D con filtros de 32 a 64 y pooling global. (b) Evolución de la pérdida en entrenamiento y validación.

Los resultados de 8b son razonables, sin embargo, existe un brecha de perdida $\mathcal{L} = 1$ que no se puede superar o requiere muchísimas épocas.

2.3.4. Redes convolucionales 1D

A continuación se detalla la arquitectura:

- Conv1d de feature_len → 128 (kernel=3, padding=1), LeakyReLU y MaxPool1d (k=3, s=3).
- Conv1d de 128→256 (kernel=3, padding=1), LeakyReLU y MaxPool1d (k=3, s=3).
- Conv1d de 256→512 (kernel=3, padding=1), LeakyReLU y MaxPool1d (k=3, s=3).
- AdaptiveAvgPool1d a tamaño 1, reshape a $(B, 512)$ y Dropout($p = 0.5$).
- Linear de 512→128 + LeakyReLU, seguido de Linear de 128→10.

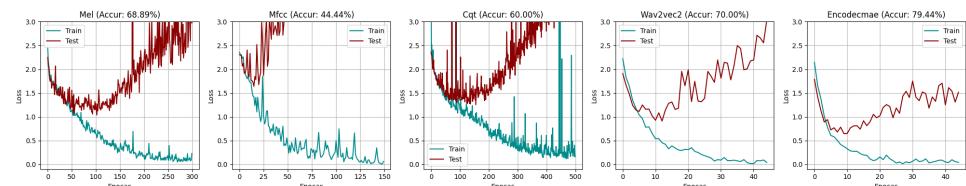


Figura 9: Curvas de aprendizaje del CNN 1D

En general, se obtiene una performance comparable o mejor que las anteriores. Sin embargo, MEL, MFCC y CQT muestran gran inestabilidad mientras todos sobreajustan rápidamente.

2.4. Encoder-Transformer

La idea detrás de este enfoque es aportar contexto a las representaciones que no son embeddings y ajustar ligeramente las que sí lo son. Para ello, empleamos un de $N = 4$ capas de encoder Transformer (véase Figura 10) y $dim_feedforward = 16$.

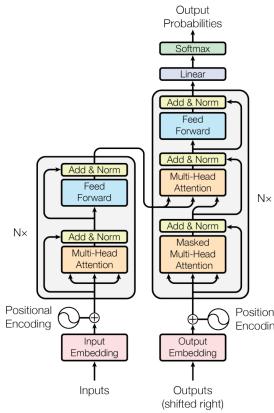


Figura 10: Esquema del Transformer

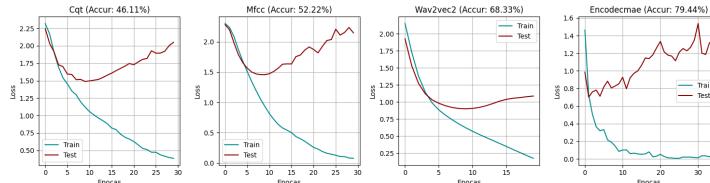


Figura 11: Curvas de aprendizaje del Transformer

En las curvas de la Figura 11 se observa que el Transformer logra mejorar la estabilidad respecto a las CNN1D en CQT y MFCC, con precisiones finales de 46 % y 52 % respectivamente, moderando el sobreajuste. Con Wav2Vec2 alcanza un 68 % y con EnCodecMAE mantiene un sólido 79 %. Notar MEL no fue incluido, esto se debe a cuestiones de hardware.

Observación, $dim_feedforward = 16$ genera un cuello de botella, se experimentó con $dim_feedforward = 2 \times emb_dim$ como recomienda la literatura sin embargo los resultados no fueron superiores.

3. Evaluación

El mejor modelo se obtiene de la mano de la unidad de recurrencia GRU 2.3.2. A continuación se lo pone a prueba con los datos de evaluación.

Genero	Precision	Recall	F1-score	Support
blues	0.90	0.90	0.90	10
classical	1.00	1.00	1.00	10
country	0.73	0.80	0.76	10
disco	0.90	0.90	0.90	10
hiphop	0.75	0.90	0.82	10
jazz	1.00	1.00	1.00	10
metal	0.91	1.00	0.95	10
pop	0.88	0.70	0.78	10
reggae	0.82	0.90	0.86	10
rock	0.57	0.40	0.47	10
macro avg	0.85	0.85	0.84	100
accuracy	0.85	-	-	100

Tabla 4: Performance del mejor modelo

La mejor versión (Tabla 4), presenta un desempeño global de 85 % de accuracy, con un F1-score medio de 0.84.

Las clases *classical* y *jazz* alcanzan la puntuación perfecta (1,00, 1,00, 1,00), lo que indica que estas categorías quedan perfectamente discriminadas, esto se condice claramente con lo observado en el UMAP 2 de la introducción. Por otro lado, géneros como *metal*, *blues* y *disco* también muestran métricas muy elevadas ($F1 \geq 0,90$), lo cual es nuevo porque en el UMAP estas clases estaban muy mezcladas.

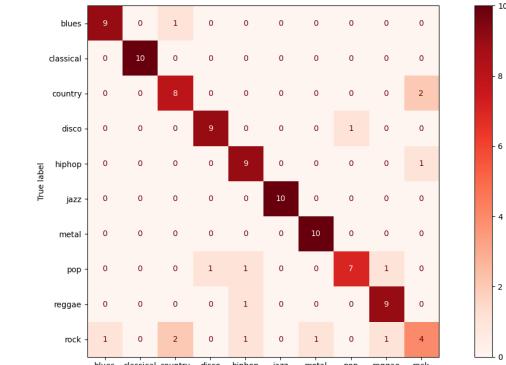


Figura 12: Matriz de confusión del mejor modelo

Por otro lado, *rock* es el más problemático, con un F1-score de apenas 0.47 y un recall de 0.40, lo que sugiere confusiones frecuentes con otros estilos. Mas específicamente, por la matriz de confusión 12, sólo 4 de las 10 muestras de *rock* se etiquetan correctamente; el resto se reparte entre *blues* (1), *country* (2), *disco* (1) y *pop* (1), lo que explica su bajo recall (0.40) y F1-score (0.47).

4. Conclusión

Los resultados muestran que los embeddings pre-entrenados, especialmente los de EnCodecMAE, aportan una riqueza de características que se traduce en un rendimiento notablemente superior al de las representaciones tradicionales.

El mejor desempeño se obtuvo con la GRU bidireccional sobre embeddings EnCodecMAE, alcanzando un 85 % de accuracy y un F1-score macro de 0.84, con clasificación perfecta de géneros como *classical* y *jazz*. Sin embargo, géneros como *rock* presentan errores significativas, se sugiere la necesidad de nuevas estrategias específicas (por ej. aumentar la cantidad de datos, ponderación de clases o arquitecturas mas complejas) para mejorar la discriminación en categorías más complejas. Por supuesto, implicaría un upgrade en el hardware actual.

Referencias

- [1] Leonardo Pepino, Pablo Riera, Luciana Ferrer. *EnCodecMAE: Leveraging neural codecs for universal audio representation learning* 2023
- [2] Schoerkhuber, Christian, and Anssi Klapuri. *Constant-Q transform toolbox for music processing*. 7th Sound and Music Computing Conference, Barcelona, Spain. 2010.
- [3] Lena Sophie Brüder. *Music classification using Constant-Q based Features* Master's Thesis, Ruhr-University Bochum, in cooperation with Research In Motion Deutschland GmbH. 2013