COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA MATEMÁTICA E INTELIGENCIA ARTIFICIAL

TRABAJO FIN DE GRADO

Leveraging Natural Language Processing Techniques for Music Metadata Enhancement

Autor: Emma Rey Sánchez

Directores: Jaime Pizarroso Gonzalo, Andrés Occhipinti Liberman

Madrid

June 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Leveraging Natural Processing Language Techniques for Music Metadata Enhancement

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2024/25 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Emma Rey Sánchez Fecha: 11/ 06/ 2025

Autorizada la entrega del proyecto

LOS DIRECTORES DEL PROYECTO

Fdo.: Jaime Pizarroso Gonzalo Fecha: 18/ 06/ 2025

Fdo.: Andrés Occhipinti Liberman Fecha: 18/ 06/ 2025

Agradecimientos.

Me gustaría en primer lugar agradecer a mi compañero de carrera y mejor amigo Carlos Mazuecos Reíllo por su apoyo incondicional durante todo el proceso de este proyecto.

También agradecer a Javier Rojo Llorens, y a su apoyo, que ha sido fundamental para culminar este trabajo.

Y finalmente agradecer a mis mentores Jaime Pizarroso Gonzalo y Andrés Occhipinti Liberman por su ayuda constante, guiándome con claridad en cada etapa del desarrollo.

LEVERAGING NATURAL PROCESSING LANGUAGE TECHNIQUES FOR MUSIC METADATA ENHANCEMENT

Autor: Rey Sánchez, Emma.
Directores: Pizarroso Gonzalo, Jaime; Occhippinti Liberman, Andrés.
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

Resumen

Este proyecto investiga si un régimen de entrenamiento en dos etapas puede mejorar la clasificación de géneros musicales. En lugar de entrenar modelos de deep learning directamente sobre un gran número de géneros detallados, se realiza una primera etapa en la que los géneros similares se agrupan en doce "macrogéneros". Los modelos se entrenan inicialmente con esta tarea simplificada y luego se afinan para clasificar los 103 géneros originales. Usando espectrogramas de Mel como entrada y evaluando arquitecturas CNN, ResNet-18 y Vision Transformer, el estudio concluye que el preentrenamiento jerárquico mejora la generalización en conjuntos de datos a gran escala y desbalanceados. Se utiliza el dataset Free Music Archive (FMA) como referencia principal.

1. Introducción

El metadato musical desempeña un papel esencial en la industria musical, permitiendo la clasificación, descubrimiento y distribución de canciones. Una de sus funciones clave es la clasificación por género, que facilita la navegación en catálogos, respalda sistemas de recomendación y aumenta la visibilidad de los artistas. Mientras que los métodos tradicionales dependían de características diseñadas manualmente y modelos simples, los avances recientes en deep learning permiten extraer patrones informativos automáticamente a partir del audio.

Sin embargo, los modelos de aprendizaje profundo tienen dificultades para escalar cuando el número de etiquetas de género crece y el conjunto de datos se desbalancea. Las fronteras entre géneros son difusas y muchas canciones comparten características entre categorías. Para afrontar estos retos, este proyecto analiza el impacto de un enfoque jerárquico de entrenamiento. Al entrenar primero con categorías amplias (macrogéneros) y luego afinar con etiquetas más detalladas, se busca facilitar el aprendizaje de representaciones significativas y mejorar la precisión final.

El dataset Free Music Archive (FMA) se emplea para validar este enfoque. Contiene más de 100,000 fragmentos de audio etiquetados en 161 géneros. Los experimentos se centran en el subconjunto "FMA Small" de 8,000 pistas, lo cual permite una experimentación viable manteniendo diversidad y complejidad.

2. Definición del proyecto

La pregunta central de este proyecto es: ¿puede el preentrenamiento jerárquico mejorar el rendimiento de los modelos de deep learning en la clasificación de géneros musicales a gran escala? Para responderla, se comparan dos regímenes de entrenamiento:

- Estándar: los modelos se entrenan desde cero para clasificar 103 géneros.

- Jerárquico: los modelos se entrenan primero para clasificar 12 macrogéneros y luego se afinan con los 103 géneros originales.

Los macrogéneros fueron definidos manualmente según similitudes acústicas y culturales percibidas, agrupando por ejemplo géneros como Indie Rock, Psych Rock y Noise Rock bajo "Rock". Esta simplificación ayuda a los modelos a aprender patrones generales antes de abordar distinciones que requieren detalles más finos y complejos.

3. Descripción del sistema

El pipeline comienza con el preprocesamiento del dataset FMA Small. Cada clip de 30 segundos se convierte en una imagen de espectrograma de Mel de 224x224 píxeles utilizando Librosa y Matplotlib. El conjunto se divide en entrenamiento, validación y prueba con una proporción 80/10/10.

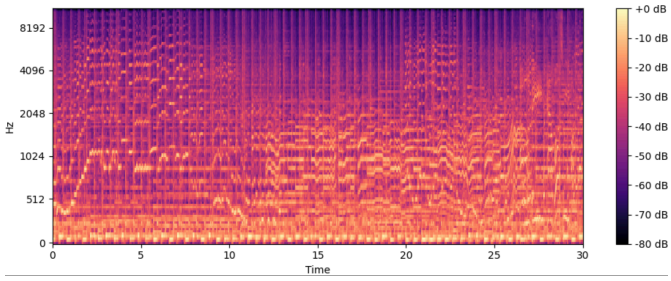

Figura 1 – Ejemplo de espectrograma Mel empleado en el entrenamiento

El entrenamiento de los modelos se realiza en dos fases:

- Preentrenamiento con macrogéneros: cada arquitectura se entrena para clasificar uno de los 12 grupos de géneros.
- Fine-tuning: se reemplaza la capa final por una con 103 clases y se continúa el entrenamiento.

CNN: Arquitectura con dos bloques de convolución, max pooling, dropout y capas densas.

ResNet-18: Inicializada con pesos de ImageNet, se adapta la capa final para la tarea de clasificación por géneros.

ViT: Aplica atención sobre parches 16x16 del espectrograma, también preentrenado en ImageNet.

Optimización: Se utilizan Adam y AdamW con early stopping y label smoothing. Se aplican aumentos de datos como recorte, volteo y alteración de color.

4. Resultados

La Tabla 1 resume las precisiones obtenidas en test para cada arquitectura en tres escenarios: entrenamiento con macrogéneros (12 clases), entrenamiento directo con 103 géneros y fine-tuning tras el preentrenamiento.

Modelo	*Macrogéneross(12)*	*Todos los géneros (103)*	*Después del Fine-tuning (103)*	*Mejora (%)*
CNN	50.00%	31.79%	33.42%	1,63%
Vision Transformer	51.86%	31.92%	35.65%	3,73%
Resnet-18	59.53%	40.06%	42.93%	2,87%

Tabla 1 – Precisiones obtenidas en test para cada arquitectura y régimen de entrenamiento. El porcentaje de mejora es la diferencia entre la precisión del modelo entrenándolo desde 0 en todos los géneros (3ª columna) y el mismo modelo entrenándolo en dos etapas (4ª columna)

Los resultados muestran una caída significativa en la precisión al pasar de la tarea simplificada de 12 clases a la clasificación completa de 103 géneros. No obstante, todos los modelos mejoran su rendimiento cuando se afinan tras el preentrenamiento jerárquico. El Vision Transformer es el que más se beneficia, con una mejora del 3.73%, seguido por ResNet-18 con 2.87% y CNN con 1.63%. ResNet-18 alcanza la mayor precisión global, lo que destaca la eficacia de las conexiones residuales y los pesos preentrenados en la generalización ante estilos musicales diversos.

5. Conclusiones

Los experimentos confirman que el preentrenamiento jerárquico mejora la clasificación de géneros, especialmente en modelos complejos como ResNet-18 y ViT. Al aprender primero a distinguir categorías amplias, los modelos desarrollan representaciones más transferibles que les ayudan a clasificar etiquetas más detalladas.

Este enfoque resulta especialmente útil en conjuntos de datos grandes y desbalanceados, donde el entrenamiento directo puede llevar al sobreajuste o una generalización deficiente. Incluso la CNN ligera mostró mejoras, lo que sugiere que el método es aplicable en una amplia gama de arquitecturas.

Entre las líneas futuras se encuentran: automatizar la agrupación de géneros, permitir clasificación multi-etiqueta o aplicar el enfoque a conjuntos más grandes como FMA Medium. También sería interesante analizar la interpretabilidad de los modelos para identificar qué regiones del espectrograma influyen más en las decisiones de clasificación.

LEVERAGING NATURAL PROCESSING LANGUAGE TECHNIQUES FOR MUSIC METADATA ENHANCEMENT

Author: Rey Sánchez, Emma.
Directors: Pizarroso Gonzalo, Jaime; Occhippinti Liberman, Andrés.
Collaborating Entity: ICAI – Universidad Pontificia Comillas

Summary

This project investigates whether a two-stage training regime can improve music genre classification. Instead of training deep learning models directly on a large number of fine-grained genres, a first stage groups similar genres into twelve "macrogenres." Models are initially trained on this simplified task and then fine-tuned to classify the original 103 genres. Using Mel-spectrograms as input and evaluating CNN, ResNet-18, and Vision Transformer architectures, the study finds that hierarchical pretraining improves generalization in large-scale, imbalanced datasets. The Free Music Archive (FMA) dataset is used as the main benchmark.

1. Introduction

Metadata plays a critical role in the music industry, enabling the classification, discovery, and distribution of tracks. Among its many facets, genre classification allows users to browse catalogs, supports recommendation systems, and increases exposure for artists. While traditional classification relied on manually engineered features and simple models, recent advances in deep learning allow automatic extraction of informative patterns from audio data.

However, deep learning models struggle with scalability when the number of genre labels grows and the dataset becomes imbalanced. Genre boundaries are often fuzzy, and musical tracks may share characteristics across multiple labels. To address these challenges, this project explores the impact of a hierarchical training approach. By training models first on broad genre categories (macrogenres) and then fine-tuning them on the detailed set, the goal is to facilitate the learning of meaningful representations and improve final classification accuracy.

The Free Music Archive (FMA) dataset is used to test this approach. It offers more than 100,000 audio clips labeled across 161 genres. The experiments focus on the "FMA Small" subset of 8,000 tracks, allowing for feasible experimentation while preserving diversity and complexity.

2. Project Definition

The central question this project addresses is: can hierarchical pretraining improve the performance of deep learning models for large-scale music genre classification? To answer it, two training regimes are compared:

- Standard: Models are trained from scratch to classify 103 genres.
- Hierarchical: Models are first trained to classify 12 macrogenres, then fine-tuned on the 103 original genres.

The macrogenres were manually defined based on perceived acoustic and cultural similarity, grouping genres like Indie Rock, Psych Rock, and Noise Rock under "Rock." This simplification helps the model learn coarse patterns before dealing with fine-grained distinctions.

Three architectures are tested: a lightweight CNN, ResNet-18, and a Vision Transformer (ViT). All models use Mel-spectrograms as input, which convert audio into 2D images by mapping frequencies to the Mel scale and measuring power over time. This image-like input enables the use of visual classification techniques.

3. System description

The pipeline begins with preprocessing the FMA Small dataset. Each 30-second clip is converted into a 224x224 Mel-spectrogram image using Librosa and Matplotlib. The dataset is split into training, validation, and test sets with an 80/10/10 ratio.

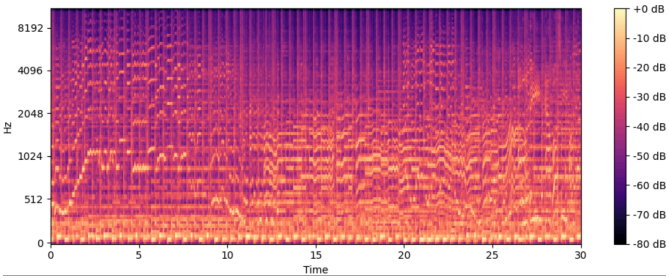

Figure 2 - Example Mel-spectrogram used for training

Models are trained in two phases:

- Macrogenre pretraining: Each architecture is trained to classify one of 12 genre groups.
- Fine-tuning: The classification head is updated for 103 classes, and the model is trained further.

CNN: A two-block architecture with max pooling, dropout, and fully connected layers.

ResNet-18: Initialized with ImageNet weights, the final layer is adapted for the genre task.

ViT: Uses self-attention over 16x16 spectrogram patches, pretrained on ImageNet.

Optimization: Adam and AdamW optimizers are used with early stopping and label smoothing. Augmentations like cropping, flipping, and color jitter are applied during training.

4. Results

Table 1 summarizes the test accuracies for each architecture across three settings: macrogenre training (12 classes), direct training on 103 genres, and fine-tuning after macrogenre pretraining.

Model	*Macrogenres(12)*	*All Genres (103)*	*After Fine-tuning (103)*	*Improvement (%)*
CNN	50.00%	31.79%	33.42%	1,63%
Vision Transformer	51.86%	31.92%	35.65%	3,73%
Resnet-18	59.53%	40.06%	42.93%	2,87%

Table 1 – Accuracies obtained in tests for each architecture and training regime. The percentage improvement is the difference between the model's accuracy when trained from zero across all genres (3rd column) and the same model when trained in two stages (4th column).

The results reveal a substantial decrease in accuracy when transitioning from the simplified 12-class macrogenre task to the full 103-genre classification. Nevertheless, all models demonstrate improved performance when fine-tuned after macrogenre pretraining. The Vision Transformer benefits the most from this hierarchical approach, achieving a 3.73% improvement in accuracy, followed by ResNet-18 with 2.87%, and CNN with 1.63%. Among all models, ResNet-18 delivers the highest overall accuracy, highlighting the effectiveness of residual connections and pretrained weights in generalizing across diverse musical styles.

5. Conclusions

The experiments confirm that hierarchical pretraining improves genre classification performance, particularly for complex models like ResNet-18 and ViT. By first learning to distinguish broad genre categories, the models develop more transferable representations, which enhances their ability to classify fine-grained labels.

This approach is especially beneficial in large, imbalanced datasets where direct training can lead to overfitting or poor generalization. Even the lightweight CNN showed improvements, suggesting that the method is broadly applicable.

Future work could involve automated genre clustering, multi-label classification, or applying the method to larger datasets like FMA Medium. Interpretable models could also help identify which audio features contribute most to genre recognition.

Table of contents

1. INTRODUCTION

This project explores the application of advanced Natural Language Processing (NLP) techniques in the music domain, aiming to explore diverse methodologies to automate the generation and enhancement of metadata for songs.

1.1. CONTEXT AND MOTIVATION

Music metadata refers to the embedded descriptive information in audio files that provides essential details about a song. This metadata serves to identify, label, and present audio content, encompassing elements such as the artist, producer, genre, composer, BPM, copyright details, release date, and album title. By enabling the systematic organization, storage, and utilization of files in various contexts and applications, metadata plays a crucial role in the music industry. Furthermore, it forms a critical bridge between technology and artistic creation, underpinning essential processes like song discovery, copyright management, and royalty payments.

This project is driven by the motivation to enrich listeners' musical experiences by making it easier to find, explore, and connect with tracks that match their tastes. Well-curated metadata is fundamental to achieving this objective, as it not only boosts a song's visibility in search results but also facilitates intuitive browsing by genre, mood, or specific instrumentation.

Beyond benefiting audiences, detailed metadata also increases exposure for artists. When tracks are simpler to locate, they gain a wider audience and become easier to share. Accurately credited works further pave the way for potential collaborations with other artists and industry professionals looking to partner on related projects. In particular, the classification of songs by genre and mood offers a clear pathway through extensive music libraries, enabling users to identify the style that aligns with their current mindset or personal preferences. By focusing on robust metadata, the project not only promotes an efficient musical search process but also lays the foundation for more personalized and meaningful interactions between listeners and musical content.

1.2. PROJECT OVERVIEW AND OBJECTIVES

This project investigates the effectiveness of a hierarchical pretraining strategy in large-scale music genre classification. Instead of directly training models to predict fine-grained genre labels, the original genre taxonomy is first grouped into a smaller set of acoustically and culturally coherent macrogenres. Models are initially trained on this simplified classification task and subsequently fine-tuned on the original, more detailed genre set. This two-stage approach is compared against a standard regime in which models are trained from scratch to predict the full set of genres.

Three deep learning architectures, Convolutional Neural Networks (CNN), Vision Transformers (ViT), and ResNet-18, are evaluated under both training regimes. All models operate on Mel-spectrograms, which encode audio signals into time–frequency representations suitable for image-based classification. Experiments are conducted on the Free Music Archive (FMA) dataset [1], which provides over 100,000 audio tracks across 161 genres, offering a realistic benchmark characterized by high class imbalance and genre detail. By keeping the preprocessing pipeline, input format, and evaluation protocol constant, the study isolates the impact of hierarchical supervision on model performance.

The following sections will delve into the technical details—from data preprocessing and macrogenre grouping to model architecture design and training protocols—culminating in a thorough performance comparison that highlights the advantages of hierarchical pre-training in music information retrieval.

2. STATE OF THE ART

Automatic music genre classification has evolved through successive paradigms, from hand-crafted features and classical classifiers to end-to-end deep learning architectures that ingest spectrogram images. Early approaches relied on features such as Mel-Frequency Cepstral Coefficients (MFCCs), chroma and timbre descriptors, coupled with support vector machines or Gaussian mixture models to assign genre labels. Although these systems achieved moderate performance, their dependence on manual feature engineering limited their ability to generalize across diverse musical styles [7].

Convolutional neural networks (CNNs) reframed the problem by treating log-Mel spectrograms as images, learning hierarchies of time–frequency patterns directly. Initial three-layer CNNs demonstrated clear gains over MFCC-based pipelines on GTZAN, later supplanted by deeper backbones such as VGG, ResNet, DenseNet, and EfficientNet that achieved over 80% accuracy on GTZAN and FMA-small at the expense of tens of millions of parameters [3, 4]. Despite their strength in extracting local features, CNNs remain constrained by finite receptive fields and uniform treatment of all regions, limiting their ability to capture long-range temporal dependencies intrinsic to musical form.

To incorporate sequential context, hybrid convolutional–recurrent architectures (CRNNs) appended LSTM or GRU layers to CNN-extracted feature maps. Such models improved upon pure CNNs for FMA-small, reaching around 65% accuracy while preserving moderate parameter counts [7]. However, recurrent stages introduce sequential bottlenecks that impede parallel training efficiency on modern accelerators.

Attention-based models inspired by the Vision Transformer (ViT) have recently been applied to spectrograms by dividing inputs into non-overlapping patches and employing self-attention to model global relationships. Pretrained ViTs fine-tuned on spectrogram images can rival deep CNNs on GTZAN [8], but they demand substantial computational resources, careful optimization—often involving AdamW or other advanced optimizers [5, 6]—and aggressive data augmentation to mitigate overfitting.

Hybrid CNN–Transformer designs combine localized convolutional feature extraction with global self-attention. The CNN-TE architecture, for example, applies a lightweight convolutional front end followed by Transformer encoder layers, achieving state-of-the-art results on both GTZAN and FMA with fewer parameters and faster inference than monolithic CNNs [1]. Such hybrids leverage convolutional inductive bias for low-level pattern detection while reserving attention mechanisms for modeling high-level, time-spanning dependencies.

Nevertheless, most studies focus on small to medium-scale datasets—such as GTZAN and the smaller FMA variants—leaving the behavior of modern architectures on large, imbalanced taxonomies (e.g., FMA “large” with 161 genres) largely unexplored [2, 4]. Furthermore, although hierarchical label structures are a natural fit for genre classification, few works have systematically investigated whether introducing coarse-to-fine label progression—such as grouping genres into higher-level clusters before fine-grained classification—can improve model performance [3]. This gap highlights the need for more empirical studies that evaluate hierarchical training setups under controlled conditions and at scale.

In summary, while CNNs, CRNNs, Transformers, and their hybrids have each advanced music-genre classification, significant gaps remain in scalability, reproducibility, and the exploitation of hierarchical genre structures. A structured comparison of these architectures, using consistent preprocessing and evaluation setups, can help clarify how model design and training choices affect performance in large-scale music genre classification.

3. METHODOLOGY

To implement and evaluate the proposed hierarchical and single-stage training regimes for music genre classification, a comprehensive pipeline encompassing data acquisition, preprocessing, model design, and rigorous training and validation protocols has been devised. This section describes in detail each step of that pipeline, highlighting how raw audio files were transformed into model-ready inputs, how three distinct deep-learning architectures were constructed and adapted to this domain, and how all models were trained and evaluated under both the standard and hierarchical schemes.

3.1. TECHNICAL DESCRIPTION

Accurate music genre classification begins with transforming raw audio data into representations that deep learning models can effectively interpret. In this project, Mel-spectrograms were selected as the input format, as they allow framing the problem as an image classification task, enabling the use of convolutional and attention-based architectures originally developed for visual pattern recognition. A detailed explanation of how Mel-spectrograms are computed and why they are suitable for this task is provided in Section 3.2.

Each audio clip from the Free Music Archive (FMA) dataset is converted into a fixed-size spectrogram image. The dataset is split into training, validation, and test sets following an 80/10/10 ratio, ensuring that each class remains balanced across splits.

To address the challenges posed by large-scale classification and improve model training efficiency, the original 103 FMA genres were manually grouped by the author into twelve broader macrogenres. This grouping was based on perceived acoustic similarity and musical proximity, informed by an exploratory review of genre characteristics and listening comparisons. Subgenres such as *Indie-Rock*, *Psych-Rock*, and *Noise-Rock* were, for example, grouped under the *Rock* macrogenre due to their shared instrumentation and production style. The idea is that coarse distinctions are learned first, and fine-grained categorization is tackled in later stages.

The full mapping between macrogenres and subgenres, along with the number of tracks per subgenre, is provided in the Annex. This manual reduction from 103 to 12 classes was a key design choice to support structured learning and facilitate genre generalization.

The macrogenre grouping significantly simplifies the initial learning task. By first learning to distinguish between broad categories (e.g., Rock vs. Pop), the model can develop robust low-level filters before fine-tuning on the detailed 103-genre task.

This approach of hierarchical pretraining, starting with macrogenres and progressing to fine-grained genres, has been shown to facilitate scalability and improve classification performance in complex settings. [4]

3.2. DATA REPRESENTATION

To enable the use of image-based neural architectures for audio classification, each audio file is transformed into a 2D visual representation capturing its time-frequency characteristics. Specifically, the raw waveform is first converted into a Mel-spectrogram, a perceptually meaningful representation that reflects the human ear's sensitivity to different frequencies. This is done by applying the Short-Time Fourier Transform (STFT) to split the signal into overlapping time windows, computing the power spectrum for each, and mapping the resulting values onto the Mel scale. The logarithm of the magnitude (in decibels) is then taken to emphasize salient features. These spectrograms are stored as RGB images and resized to 224×224 pixels to match the input requirements of the visual backbone models. The final classification task is therefore formulated as an image classification problem, where each spectrogram corresponds to one of twelve genre labels.

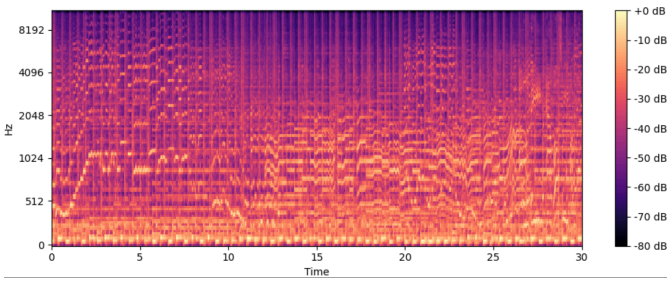

Figure 1 - Mel-spectrogram of a 30-second country music snippet. The vertical axis represents frequency bins mapped to the perceptual Mel scale, while the horizontal axis corresponds to time (in seconds). The color intensity encodes power in decibels (dB), with brighter regions indicating stronger spectral energy at specific time–frequency locations.

3.3. MODEL DESIGN

To explore how different neural architectures handle spectrogram-based audio classification, three models were implemented: a custom lightweight Convolutional Neural Network (CNN), a ResNet-18 with residual blocks, and a Vision Transformer (ViT). Each model was chosen to represent a distinct learning paradigm—local pattern recognition via convolutions, hierarchical feature refinement via residual learning, and global context modeling via self-attention. This diversity allows a meaningful comparison across different architectural biases in both standard and hierarchical training regimes.

CNN

The CNN architecture processes spectrograms resized to 128×128 pixels. It consists of two convolutional blocks:

- The first block uses a 5×5 convolution (stride 2, 16 channels), followed by ReLU, 2×2 max pooling, and spatial downsampling.
- The second block uses a 3×3 convolution (stride 2, 32 channels), followed by ReLU, a 4×4 max pooling, and batch normalization.

After these layers, the output tensor of shape [32, 6, 6] is flattened and passed through two fully connected layers with 1000 and 500 units, respectively, each followed by dropout (p = 0.2). The final classification head maps to either 12 or 103 outputs, depending on the task. Dropout (p = 0.6) is applied after the convolutional layers to prevent overfitting. The architecture is trained from scratch using standard cross-entropy loss, optimized by Adam for pretraining and AdamW during fine-tuning.

ResNet-18

ResNet-18, a widely used deep residual network, was selected for its proven capacity to learn robust image features. Spectrograms are resized to 224×224 and normalized to ImageNet statistics. The model is initialized with ImageNet-pretrained weights, and its final fully connected layer is replaced by a dropout (p = 0.5) and a linear projection to either 12 (macrogenres) or 103 (all genres) classes.

During macrogenre training, moderate augmentations are applied (random resized cropping, horizontal flips, and color jitter), and training uses cross-entropy loss with label smoothing (ε = 0.1) to encourage generalization across acoustically similar genres. In the fine-tuning stage, the model is partially reinitialized: backbone weights are restored from the 12-class checkpoint where shapes match, and the classification head is reset. Fine-tuning proceeds with reduced learning rates and a learning-rate scheduler for robustness.

ViT (Vision Transformer)

The Vision Transformer introduces a fundamentally different approach by treating each 224×224 spectrogram as a sequence of 16×16 image patches. Using the *vit_base_patch16_224* architecture from TIMM, pretrained on ImageNet, the model is augmented with a custom classification head tailored to the genre task.

During pretraining, augmentations include random resized cropping, flipping, rotation (±15°), and color jitter. Initially, only the head is trained while the transformer layers remain frozen; later, full fine-

tuning is enabled with AdamW and early stopping. The attention-based architecture enables ViT to model long-range dependencies within spectrograms, potentially capturing global temporal–spectral patterns overlooked by purely convolutional approaches.

3.4. TRAINING AND VALIDATION

Training proceeds in two stages—macrogenre pretraining followed by fine-tuning on all 103 genres—using consistent validation monitoring and early-stopping rules to ensure generalization. All networks optimize the cross-entropy loss [10] over C classes.

Two optimizers are employed: **Adam** [5] and **AdamW** [6]. Adam updates parameters θ by computing first and second moment estimates of the gradient. AdamW modifies this update by decoupling weight decay λ from the gradient calculations. This decoupling allows better control of regularization, particularly beneficial during fine-tuning. Both optimizers are used throughout the training pipeline, with Adam typically applied in initial stages and AdamW during fine-tuning to stabilize convergence and prevent overfitting.

Common Setup

To ensure reproducibility, random seeds are fixed across libraries. Data is loaded using PyTorch DataLoader with a batch size of 32. Validation and test splits remain fixed, while training data is shuffled each epoch. The loss function is standard cross-entropy, with label smoothing ($\varepsilon = 0.1$) applied in ResNet training to reduce overconfidence. All models are evaluated using top-1 accuracy on the validation and test sets.

CNN Training Protocol

During macrogenre training, the CNN is initialized with 12 output units. The optimizer is **Adam** with learning rate $1\times10{-}41\times10{-}4$. After ten epochs, the checkpoint with the highest validation accuracy is saved. For fine-tuning, the classification head is replaced with a 103-unit layer, and the remaining weights are loaded from the 12-class checkpoint. Training uses AdamW with weight decay $1\times10{-}21\times10{-}2$, and early stopping is applied based on validation loss.

ViT Training Protocol

The ViT uses **AdamW** (lr $= 1\times10{-}41\times10{-}4$, weight decay $= 5\times10{-}35\times10{-}3$) throughout. Macrogenre pretraining includes strong augmentations, while fine-tuning uses only resizing and normalization. During fine-tuning, a new classification head is trained from scratch with pretrained backbone weights selectively restored where tensor shapes match. Early stopping halts training after three epochs without validation loss improvement.

ResNet-18 Training Protocol

ResNet training mirrors ViT's regime but adds label smoothing and uses a learning rate scheduler (ReduceLROnPlateau) in the fine-tuning phase. During macrogenre training, ResNet runs for 20 epochs, saving the best checkpoint based on validation accuracy. In fine-tuning, the classification head is reinitialized, backbone weights are reused, and a reduced learning rate (5×10^{-5}) is applied. Validation loss is monitored each epoch, and the best model is selected using early stopping.

Evaluation

Validation is always conducted in evaluation mode to disable dropout and update batch norm statistics. Each epoch's validation loop computes loss and accuracy. The final test accuracy is reported using the best model checkpoint (based on validation loss for fine-tuning, or validation accuracy for pretraining).

4. EXPERIMENTS

This section describes all the elements necessary to reproduce the experiments conducted in this project. As mentioned in Section 1.2, the main objective of this project is to evaluate how the choice of training regime influences the performance of music genre classification models. It seeks to determine whether introducing a hierarchical pre-training, where models are first trained to distinguish broad genre categories before fine-tuning on specific subgenres, can improve generalization and accuracy compared to training directly on the full set of fine-grained genres.

The following subsections detail the datasets used, the preprocessing steps applied, and the configuration of both the hardware/software environment and the model training protocols.

4.1. DATASET

Initial prototyping on the GTZAN dataset [9] provided a lightweight environment for validating the spectrogram-generation pipeline and tuning core model parameters; the main evaluation then leverages the Free Music Archive (FMA) [1] dataset, a well-known collection of Creative Commons-licensed music, widely used for tasks involving music information retrieval. The FMA dataset comprises over 100,000 tracks across 161 annotated genres, ranging from popular styles like Rock, Jazz, and Electronic to niche genres like Chiptune, Glitch, or Avant-Garde.

Each track in the FMA is accompanied by rich metadata (title, genre, license, duration, artist), and the dataset includes curated splits for training, validation, and testing. All tracks are sampled at 44.1 kHz and provided in 30-second MP3 clips. The genre annotations are multi-level, allowing for coarse- and fine-grained categorizations, which made it particularly suitable for hierarchical classification strategies.

Initially, experiments began with the FMA Medium subset (approximately 25,000 tracks), which balances data volume and genre diversity. However, a literature review revealed that many prior works achieved strong results using smaller subsets such as FMA Small, containing only 8,000 tracks. Consequently, the dataset was scaled down to FMA Small to ensure comparability, reduce computational costs, and maintain consistency with standard benchmarks. This subset offers an ideal tradeoff between complexity and feasibility for model training on consumer-grade hardware.

Despite the lower volume, FMA Small still reflects essential challenges in genre classification: temporal ambiguity, overlapping genre characteristics, and limited intra-class consistency. These aspects make it suitable for evaluating how architectural biases and hierarchical learning regimes affect classification.

Data Preparation

Two variants of the dataset were prepared:

- **12-Class Macrogenre Dataset**: A custom mapping collapsed over 100 original genres into twelve high-level "macrogenres," including Rock, Pop, Jazz, Folk, Metal, Classical, and more. These were selected based on both acoustic similarity and cultural proximity, using genre taxonomy principles and musicological intuition.

- **103-Class Full Genre Dataset**: This included all sufficiently represented genres in FMA Small, leading to a high-dimensional classification task. Rare genres with fewer than a handful of examples (e.g., "Western Swing" with only 4 samples) were excluded to prevent instability during training and validation.

All tracks were converted into Mel-spectrograms, which transform the raw waveform into a perceptually meaningful time–frequency representation. These spectrograms were stored as PNG images, with one per track, and organized into folders labeled by genre. Preprocessing was conducted using Librosa and Matplotlib, and visual normalization ensured consistency across samples.

Model	*Macrogenres(12)*	*All Genres (103)*	*After Fine-tuning (103)*	*Improvement (%)*
CNN	50.00%	31.79%	33.42%	1,63%
Vision Transformer	51.86%	31.92%	35.65%	3,73%
Resnet-18	59.53%	40.06%	42.93%	2,87%

Table 1 - Accuracies obtained in tests for each architecture and training regime. The percentage improvement is the difference between the model's accuracy when trained from zero across all genres (3rd column) and the same model when trained in two stages (4th column).

Each dataset was split into training (80%), validation (10%), and test (10%) sets, with stratification to maintain class balance.

Overall, the FMA Small dataset—restructured both for hierarchical and flat classification—provided a robust and reproducible basis for comparing architectural behavior across training strategies.

4.2. CONFIGURATION

All experiments were conducted on an Apple Silicon MacBook Pro M1 leveraging the MPS backend for GPU acceleration; fallback to CPU or CUDA-enabled GPU on Linux servers was supported via simple configuration flags. The software environment included Python 3.10, PyTorch 2.1, torchvision 0.15, timm 0.8.12, librosa 0.9, scikit-learn 1.2, and matplotlib 3.7. Random number generators in Python, NumPy, and PyTorch were seeded with the value 42 to ensure reproducibility of data splits, model weight initialization, and data augmentation permutations. Training hyperparameters common to all models included a batch size of 32, the use of the Adam optimizer for initial CNN training and AdamW (learning rate = 1×10^{-4}, weight decay = 1×10^{-2}) for all Transformer and ResNet-18 runs, and early stopping with a patience of three epochs based on validation loss. Models were pretrained on the 12-class macrogenre task for ten epochs and fine-tuned on the 103-class task for up to twenty epochs, with checkpoints saved at the best validation performance.

5. RESULTS

This chapter presents and interprets the classification performance achieved by each architecture under the two-stage hierarchical regime (macrogenre pretraining followed by fine-tuning) and the single-stage regime on both the 12-class "FMA Small" subset and the full 103-class problem. Rather than detailing implementation, the focus here is on comparative outcomes, the efficacy of hierarchical pretraining, and the relative inductive biases of CNN, ViT, and ResNet-18.

5.1. SUMMARY OF TEST ACCURACIES

Error! Reference source not found. aggregates the final test accuracies for each model on the FMA Small subset and the full-genre task, both when trained from scratch and when fine-tuned following macrogenre pretraining.

As shown in Table 1, all models exhibit a marked degradation when scaling from 12 to 103 target classes. The CNN's accuracy declines by 18.21 percentage points, the ViT by 19.94 pp, and ResNet-18 by 19.47 pp. This drop highlights the intrinsic difficulty of discriminating among 103 fine-grained genres and underlines the importance of strong feature representations and regularization in high-dimensional classification tasks.

5.2. BENEFITS OF HIERARCHICAL PRETRAINING

Fine-tuning from macrogenre pretraining yields consistent gains across all architectures. Relative improvements on the 103-class task are +1.63% for the CNN, +3.73% for the ViT, and +2.87% for ResNet-18. The larger benefit for the ViT suggests that self-attention layers, when first conditioned on coarse distinctions, can more effectively adapt to subtle timbral patterns than when trained again. ResNet-18, already the strongest baseline, retains its lead and further consolidates its generalization when leveraging hierarchical transfer.

5.3. COMPARISON OF INDUCTIVE BIASES

ResNet-18's superior performance in both regimes indicates that deep residual connections and pretrained visual filters transfer most effectively to spectrogram classification. The CNN, with only two convolutional blocks, captures gross frequency–time features but lacks the depth to disentangle highly overlapping genre boundaries. The ViT, while competitive on the 12-class task, suffers from overfitting when trained from scratch on 103 genres; its reliance on large-scale attention requires staged learning to avoid memorizing misleading patterns.

The results confirm that hierarchical staging enhances multi-class genre classification, particularly for architectures with weaker built-in inductive biases. Although ResNet-18 achieves the highest absolute accuracies, the ViT's responsiveness to coarse-to-fine transfer suggests promising avenues for further curriculum design and regularization strategies. The modest improvements observed for the CNN indicate diminishing returns for very shallow models in large-scale label spaces.

6. CONCLUSION AND FUTURE WORK

This project set out to explore the impact of training regimes on music genre classification by comparing two contrasting approaches: a standard strategy, in which models are trained directly to predict fine-grained genres, and a hierarchical curriculum that first groups genres into broader macrogenres before fine-tuning on the detailed taxonomy. The central hypothesis was that guiding models through a simplified intermediate task could lead to more effective feature learning and improved final performance.

To investigate this, three deep learning architectures—CNN, ResNet-18, and Vision Transformer (ViT)—were implemented and trained using a unified input representation: Mel-spectrograms. These time–frequency representations allowed us to transform the audio classification problem into an image classification task, leveraging powerful computer vision models and standard training pipelines.

The Free Music Archive (FMA) served as the primary dataset, specifically the FMA Small subset. A careful mapping from over 100 original genre labels into twelve acoustically coherent macrogenres was carried out based on both musicological intuition and data availability. This transformation enabled a meaningful hierarchical learning setup. Initial prototyping with the GTZAN dataset ensured the robustness of the preprocessing pipeline and model configurations, allowing for a smooth transition to the more challenging and realistic FMA evaluation.

6.1. SUMMARY OF FINDINGS

The experimental results suggest that hierarchical training provides noticeable improvements across architectures, with stronger effects observed in models with higher capacity, such as ResNet-18 and ViT. This indicates that a structured label progression may help certain models learn more robust internal representations, particularly in tasks involving many classes.

While the CNN showed modest improvements under the hierarchical regime, deeper architectures were better able to leverage the macrogenre pretraining to enhance their fine-grained

classification performance. This suggests that the benefits of hierarchical learning are more pronounced in architectures that can effectively capture long-range dependencies or hierarchical patterns.

By training all models under the same conditions, with identical spectrograms, augmentation policies, and hyperparameter tuning, the comparison isolated the effects of architecture and training strategy. The use of Mel-spectrograms allowed the problem to be reframed as an image classification task, enabling the application of visual pattern recognition models in a consistent and fair way.

The hierarchical curriculum also proved to be a computationally feasible enhancement. Since the macrogenre task involved fewer classes and was easier to solve, pretraining converged quickly and served as an efficient warm-up for the more complex fine-grained classification. This structure could be especially valuable in low-resource settings, where fully training large models from scratch is not possible.

6.2. LIMITATIONS

Several limitations should be acknowledged. First, the macrogenre mapping, while intuitive and grounded in genre similarity, is ultimately subjective and was constructed manually. Different mappings may yield different results, and automated clustering methods might offer a more principled alternative.

Second, although the FMA Small dataset provides a rich set of genres and a realistic classification challenge, its scale still limits the generalization capacity of deep models, especially Transformers. Larger-scale experiments on FMA Medium or the full dataset could further validate the observed trends.

Third, the models trained in this work were evaluated exclusively on classification accuracy. However, music genre classification is inherently ambiguous and multi-label in nature. A more nuanced evaluation using confusion matrices, genre co-occurrence, or even perceptual studies could provide deeper insight into model behavior.

6.3. FUTURE WORK

Building on the findings of this project, several promising directions could be pursued to extend and refine the approach. One immediate opportunity lies in replacing the manually defined macrogenre mapping with automated discovery methods, such as unsupervised clustering based on audio embeddings or topic modeling techniques like Latent Dirichlet Allocation (LDA), to derive genre hierarchies directly from the data. This would enhance reproducibility and potentially reveal more meaningful structure within the genre space. Another natural extension involves adopting a multi-task learning setup in which models simultaneously predict both macrogenres and fine-grained genres, allowing them to benefit from hierarchical supervision without requiring separate training stages. Moreover, expanding the classification framework to support multi-label predictions would better capture the complexity of musical genre boundaries and reflect the real-world scenario in which songs often span multiple styles. Investigating model interpretability, particularly through the analysis of attention weights in Transformer-based architectures, could offer deeper insights into which spectro-temporal regions drive genre decisions and how these are influenced by hierarchical pretraining. Finally, testing the generalization of trained models to unseen genres or few-shot scenarios could help assess the robustness of the learned representations, while incorporating user feedback or perceptual evaluations may bring these systems closer to practical applications in music recommendation and discovery.

7. BIBLIOGRAPHY

[1] Chen, J., Ma, X., Li, S., Ma, S., Zhang, Z., & Ma, X. (2024). A Hybrid Parallel Computing Architecture Based on CNN and Transformer for Music Genre Classification. Electronics, 13(16), 3313. https://doi.org/10.3390/electronics13163313

[2] Defferrard, M., Benzi, K., Vandergheynst, P., & Bresson, X. (2017b, September 5). FMA: A dataset for Music Analysis. arXiv.org. https://arxiv.org/abs/1612.01840

[3] DeMarco, D., Martz, E., & Ta, R. T. H. (n.d.). From Beethoven to Beyoncé: A Deep Learning Approach to Music Genre Classification. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1244/final-projects/DominicJosephDeMarcoEricMartzReginaTHTa.pdf

[4] Goemans, L. (2023, June 19). LIACS thesis repository. Scalability of Music Genre Classification Algorithms. https://theses.liacs.nl/

[5] Kingma, D. P., & Ba, J. (2017, January 30). Adam: A method for stochastic optimization. arXiv.org. https://arxiv.org/abs/1412.6980

[6] Loshchilov, I., & Hutter, F. (2019, January 4). Decoupled weight decay regularization. arXiv.org. https://arxiv.org/abs/1711.05101

[7] Lucas Rodríguez, M. (2021). Detección automática de géneros musicales, Trabajo de fin de grado, Universidad Politécnica de Madrid. E-Archivo UPM. https://oa.upm.es/view/institution/ETSI=5FInformatica/

[8] Vos, J., Rood, T., & Ionescu, A. (2023, June 16). Comparative anlysis of the novel Audio Spectrogram Transformer (AST) for FMA genre classification. HackMD. https://hackmd.io/@jimvos/rJltGnVvh

[9] Sturm, B. L. (2013). The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. arXiv.org. https://arxiv.org/abs/1306.1461

[10] Mao, A., Mohri, M., & Zhong, Y. (2023, June 20). Cross-entropy loss functions: Theoretical analysis and applications. arXiv.org. https://arxiv.org/abs/2304.07288

ANNEX

Macro-genrse	**Sub-genre**
Rock	Rock (8406)
	Indie-Rock (5756)
	Psych-Rock (2642)
	Noise-Rock (1893)
	Post-Rock (1512)
	Krautrock (734)
	Space-Rock (497)
Pop	Pop (8570)
	Experimental Pop (7330)
	Power-Pop (1032)
Metal	Metal (973)
	Death-Metal (197)
	Black-Metal (152)
Folk	Folk/Acoustic (7320)
	Psych-Folk (2300)
	Freak-Folk (1315)
	Free-Folk (755)
	British Folk (164)
Jazz	Jazz (2526)
	Free-Jazz (1542)
	Nu-Jazz (120)
	Modern Jazz (107)
	Jazz: Out (299)
	Jazz: Vocal (99)
Punk	Punk (5546)
	Post-Punk (1930)
	Electro-Punk (568)
Hip-Hop/Rap	Hip-Hop/Rap (6612)
	Hip-Hop Beats (1220)
	Alternative Hip-Hop (742)
	Abstract Hip-Hop (202)
Electronic	Electronic (35701)
	Electroacoustic (6133)
	Lo-Fi (6075)
	Ambient Electronic (5747)
	IDM (3484)
	Glitch (2825)
	Downtempo (2097)
	Minimal Electronic (1024)
	Breakbeat (735)
	Breakcore – Hard (511)
	Drum & Bass (500)
	Bigbeat (191)
R&B / Soul	R&B/Soul (2521)
	Soul-RnB (555)
Classical Music	Classical (3393)
	20th Century Classical (297)
	Chamber Music (170)
	Opera (162)
	Choral Music (216)
	Composed Music (635)
	Minimalism (1402)
Country / Americana	Country (1065)
	Americana (1068)
	Country & Western (75)
	Western Swing (4)
World Music	International (1855)
	World/International (313)
	Latin America (510)

	Brazilian (238)
	African (232)
	Balkan (616)
	Middle East (157)
	Indian (198)
	N. Indian Traditional (4)
	South Indian Traditional (17)
	Turkish (65)
	Romany (Gypsy) (112)
	Klezmer (57)
	North African (40)
	Pacific (23)
	Asia-Far East (122)
	Spanish (117)
	Flamenco (47)
	Fado (26)
	Tango (30)
	Cumbia (68)
	Salsa (12)

While this grouping captures a substantial portion of the FMA genres, some specific genres were excluded from the macrogenre grouping, including unclassifiable genres or those with minimal data representation. Genres like *Indian Traditional* (4) or *Western Swing* (4) have very few tracks, making it difficult to train robust models on them.