MusicGen Audio Restoration with DSP: Reducing Noise, Preserving Musicality

August 29, 2025

Giorgia Cecchi (2069895)

Abstract

In questo progetto, affronto il problema del fruscio ad alta frequenza presente nelle tracce generate da *MusicGen*, applicando tre tecniche di *Digital Signal Processing* (DSP) per ridurre il rumore senza alterare la qualità musicale. La valutazione è condotta con metriche specifiche e ascolto soggettivo,e mostra, che i DSP diminuiscono efficacemente il fruscio mantenendo naturalezza e fedeltà timbrica.

1. Introduction

Negli ultimi anni, i modelli generativi di musica basati su reti neurali hanno raggiunto traguardi significativi. Tra i più noti c'è *MusicGen*,sviluppato da Gabriel Défossez e dal team di Meta AI, un modello *text-to-music* capace di generare brani musicali a partire da semplici prompt testuali.

Sebbene risulti efficace, le tracce prodotte spesso presentano fruscio ad alta frequenza (hiss) e artefatti di sintesi che ne riducono la qualità percepita all'ascolto. L'obiettivo del mio progetto, è stato quello di applicare tecniche di *audio restoration* per migliorare l'ascolto.

In un primo momento ho provato modelli neurali preesistenti, come Demucs e Facebook Denoiser, che però pur attenuando il rumore snaturavano il timbro, rendendo l'audio poco naturale e lontano dall'originale. Di conseguenza, mi sono concentrata su tre modelli di *Digital Signal Processing* (DSP), più leggeri e trasparenti.

2. Methodology

Ho seguito un workflow preciso: configurazione dell'ambiente e installazione delle librerie necessarie; caricamento di *MusicGen* (facebook/musicgen-small) da Hugging Face e generazione di cinque clip musicali; salvataggio e organizzazione dei file audio in formato `.wav` (16 kHz).

In seguito ho applicato i tre modelli DSP e li ho confrontati con lo stesso protocollo di valutazione, basato su metriche oggettive e ascolto soggettivo. I risultati sono stati riassunti in tabelle, includendo il valore di Δ, che evidenzia la differenza tra traccia originale e ricostruita per ciascuna metrica.

2.1. Evaluation metrics

Per valutare i modelli ho adottato tre metriche complementari:

- **HF-band energy (dB)**: misura l'energia al di sopra dei 6 kHz, ovvero nella banda in cui tende a concentrarsi il fruscio. Un valore più basso indica una riduzione del rumore;
- **HPSS ratio**: valuta il rapporto armonico percussivo, utile per capire se la musicalità viene preservata;
- **Log-Mel distance**: confronta lo spettrogramma log-mel della traccia originale con quello elaborato; valori bassi indicano maggiore fedeltà timbrica.

2.2. Modello 1: NoiseReduce (DSP)

NoiseReduce è un algoritmo DSP basato su *gating* spettrale: analizza l'audio a finestre e attenua le componenti vicine al rumore stimato, preservando le altre.

Strategy Per ridurre il fruscio di *MusicGen* (soprattutto sopra i 6 kHz) ho isolato questa banda con un filtro passa–alto, vi ho applicato *NoiseReduce* con parametri leggeri e ho poi ricombinato la banda pulita con il resto del brano originale.

Evaluation results I risultati sono riportati in forma di differenza (Δ) rispetto alla traccia originale.

Il modello soddisfa tutti i criteri: l'energia in banda alta diminuisce ($\Delta < 0$), il rapporto HPSS resta stabile o leggermente positivo, e la *Log-Mel distance* rimane sotto 0.20 (media ≈ 0.17).

Email: Giorgia Cecchi <cecchi.2069895@studenti.uniroma1.it>.

Machine Learning 2025, Sapienza University of Rome, 2nd semester a.y. 2024/2025.

Table 1. Risultati del Modello 1 (NoiseReduce).

Traccia	Δ HF-band [dB]	Δ HPSS	Log-Mel	Criteri
track_1	-0.812	+0.003	0.168	3/3
track_2	-3.441	+0.001	0.169	3/3
track_3	-6.752	+0.100	0.177	3/3
track_4	-6.681	+0.114	0.179	3/3
track_5	-2.281	+0.000	0.174	3/3

All'ascolto i miglioramenti sono evidenti nelle tracce più rumorose (3 e 4), mentre nelle più pulite le differenze sono minime o impercettibili.Il metodo riduce il fruscio solo dove serve, senza alterare la naturalezza.

2.3. Modello 2: High-Band Spectral Subtraction (HB-SS)

Questo modello applica sottrazione spettrale alle alte frequenze (≥ 6 kHz), dove il fruscio è più presente.

Strategy Dopo aver calcolato la STFT, ho stimato il profilo medio di rumore $N(k)$ dai frame più silenziosi e attenuato l'ampiezza spettrale solo nell'alta banda alta secondo:

$$|Y(k,t)| = \max\big(|X(k,t)| - \alpha\, N(k),\ \beta\,|X(k,t)|\big),$$

con $\alpha = 1.2$ (oversubtraction, rimozione più aggressiva) e $\beta = 0.04$ (spectral floor, per evitare *musical noise*). L'audio è stato ricostruito usando la fase originale.

Evaluation results Il Δ è sempre negativo (riduzione sistematica del fruscio), il Δ dell'HPSS ratio resta stabile o leggermente positivo, e la *Log-Mel distance* rimane nell'intervallo 0.15–0.23 (media ≈ 0.17). Solo 2 tracce su 5 rispettano tutti i criteri.

Table 2. Risultati del Modello 2 (HB-SS).

Traccia	Δ HF-band [dB]	Δ HPSS	Log-Mel	Criteri
track_1	-0.122	+0.001	0.229	2/3
track_2	-2.028	+0.001	0.159	3/3
track_3	-6.729	+0.080	0.228	2/3
track_4	-4.553	+0.096	0.215	2/3
track_5	-2.389	+0.000	0.182	3/3

La riduzione è percepibile nelle tracce rumorose (3 e 4), quasi nulla nelle più pulite (1 e 5). Il metodo è trasparente e meno aggressivo del Modello 1.

2.4. Modello 3: MCRA (Minimum Controlled Recursive Averaging) + filtro HF "safe"

Il terzo modello utilizza *MCRA* per stimare in tempo reale lo spettro del rumore e applica un filtro di Wiener a tutto lo spettro(*full band*), con guadagno minimo per preservare la naturalezza.

Strategy Ho stimato lo spettro di potenza con la STFT, ricavato il rumore con MCRA e applicato un filtro di Wiener, imponendo un guadagno minimo (g_{floor}) per evitare annullamenti totali di alcune bande.

Evaluation results Il Δ HF-band è sempre negativo, il Δ HPSS sempre positivo, ma la *Log-Mel distance* è alta: quattro tracce tra 0.43–0.49 e una (track 1) a 0.79. Nessuna traccia soddisfa tutti i criteri.

Table 3. Risultati del Modello 3 (MCRA + filtro HF safe).

Traccia	Δ HF-band [dB]	Δ HPSS	Log-Mel	Criteri
track_1	-3.544	+1.420	0.796	2/3
track_2	-6.174	+9.305	0.438	2/3
track_3	-6.709	+1.015	0.449	2/3
track_4	-4.802	+1.019	0.428	2/3
track_5	-5.336	+4.961	0.489	2/3

All'ascolto, anche nella traccia con Log-Mel distance elevata (track 1) non emergono peggioramenti evidenti: il suono rimane naturale e privo di artefatti. Nelle tracce più rumorose il fruscio è ben controllato, mentre nei brani già puliti la differenza è minima. Nel complesso, il modello è efficace e robusto, adatto quando serve un intervento *full band* deciso, accettando però un livello di trasparenza inferiore rispetto agli altri due modelli.

3. Conclusions

L'obiettivo era ridurre il fruscio tipico delle tracce generate da *MusicGen* senza compromettere la naturalezza del suono. Tutti e tre i modelli hanno ridotto l'energia in banda alta, ma con differenze nella trasparenza e nella fedeltà timbrica.

Il Modello 1 (*NoiseReduce*) si è dimostrato il più equilibrato: ha soddisfatto sempre tutti i criteri, riducendo il fruscio in modo costante senza alterare il timbro.Il Modello 2 (*HBSS*) è utile nei casi più rumorosi ma meno stabile, mentre il Modello 3 (*MCRA*) è efficace sul rumore full band ma introduce variazioni timbriche più marcate. Nel complesso, considero i tre approcci complementari, ma il Modello 1 emerge come la soluzione più bilanciata ed efficace. Ciò conferma che è possibile ridurre il fruscio di *MusicGen* senza perdere musicalità, grazie a semplici tecniche DSP.

Il codice e i file del progetto sono disponibili nel repository GitHub: `https://github.com/giorgiacecchi/musicgen-restoration/tree/main`.

Bibliography. Ho utilizzato *MusicGen* (Défossez et al., 2023), insieme a diversi approcci DSP classici come la sottrazione spettrale (Boll, 1979), MCRA (Cohen & Berdugo, 2001) e la libreria Python *NoiseReduce* (Sainburg, 2020).

References

Boll, S. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(2):113–120, 1979.

Cohen, I. and Berdugo, B. Noise estimation by minima controlled recursive averaging for robust speech enhancement. *IEEE Signal Processing Letters*, 9(1):12–15, 2001.

Défossez, G., Copet, J., Synnaeve, G., and Adi, Y. Musicgen: Simple and controllable music generation. *arXiv preprint arXiv:2306.05284*, 2023.

Sainburg, T. noisereduce: Noise reduction in python using spectral gating. https://github.com/timsainb/noisereduce, 2020. Accessed: 2025-08-29.