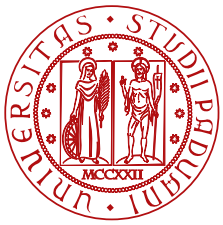


UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Realizzazione di un heterogeneous dataset per l'emulazione del plate reverb. Uno studio di caso al Computer Music and Neural Audio Communication Group, Audio Communication Group, Technische Universität Berlin

Relatore
Prof. Sergio Canazza Targon

Laureando
Pierpaolo Maria Pavarin

ANNO ACCADEMICO 2024-2025

Data di laurea 25/09/2025

Sommario

Nel contesto di un progetto di ricerca svolto presso il Computer Music and Neural Audio Communication Group e l'Audio Communication Group della Technische Universität Berlin, è stato creato un dataset eterogeneo per allenare un'intelligenza artificiale in grado di imitare il suono di un plate reverb analogico presente nei laboratori dell'ateneo. L'obiettivo era quello di digitalizzare questo effetto audio mantenendo il più possibile le sue caratteristiche originali. Per costruire il dataset sono stati raccolti diversi segnali audio di input, scelti per includere una buona varietà di timbri, dinamiche e contenuti. Successivamente, è stata eseguita l'analisi del materiale audio attraverso il calcolo delle principali feature scrivendo script Python utilizzando la libreria Librosa. Il dataset finale rappresenta una base solida per la fase di training del modello, permettendo all'IA di apprendere in modo efficace il comportamento del riverbero. Questo lavoro si inserisce nella più ampia ricerca sulla simulazione digitale di effetti sonori analogici tramite intelligenza artificiale. Oltre a questo dataset, è stato contemporaneamente sviluppato un dataset omogeneo per ottenere un training ancora più solido ed efficace.

Indice

Introduzione **1**

1 Contesto del Tirocinio **3**

1.1 Technische Universität Berlin 3

1.2 Audio Communication Group e Computer Music and Neural Audio Communication Group 4

1.3 Motivazione della scelta del tirocinio 5

2 Obiettivi **6**

2.1 Obiettivo generale: emulazione del plate reverb 6

2.2 Attività principali 10

2.3 Modalità di lavoro e strumenti utilizzati 11

2.4 Criticità incontrate 12

3 Costruzione del Dataset e Feature Extraction **18**

3.1 Il plate reverb: caratteristiche acustiche e rilevanza musicale 18

3.2 Raccolta e trattamento dei segnali audio 19

3.3 Feature extraction con Python e Librosa 24

3.4 Definizione delle features 26

3.5 Integrazione futura con il modello di IA 28

4 Apprendimenti e riflessioni **32**

4.1 Competenze tecniche acquisite 32

4.2 Riflessioni sull'applicazione dell'IA al suono 32

4.3 Crescita personale e professionale 34

Conclusioni **35**

Bibliografia **35**

Introduzione

Questa tesi è svolta nell'ambito del corso di laurea in Ingegneria Informatica presso Università degli Studi di Padova. Il lavoro è stato sviluppato in collaborazione con il gruppo di ricerca Audio Communication Group e si concentra sull'attività di tirocinio svolta presso tale gruppo, il quale fa parte della Technische Universität Berlin.

Contesto generale

Evoluzione della musica computazionale e dell'IA nel trattamento audio.

Obiettivo specifico

Imitare un plate reverb analogico con una IA.

Capitolo 1

Contesto del Tirocinio

1.1 Technische Universität Berlin

Il tirocinio è stato svolto presso la Technische Universität Berlin.

La storia dell'elettronica del suono alla Technische Universität Berlin comincia subito dopo la Seconda guerra mondiale, quando Hans Heinz Stuckenschmidt entrò in cattedra e si iniziò a pensare di inserire la tecnologia sonora (registratori, magnetofoni, filtri) nelle lezioni di musica.

Negli anni Cinquanta, Fritz Winckel fece fare al dipartimento un salto: non solo didattica, ma realizzazione concreta di produzioni, tra cui musica elettroacustica per spettacoli come burattini. Intorno al 1960 lo studio diventa sempre più organizzato, con spazi migliori, archivi sperimentali e la costruzione anche interna di apparecchi (p.e. un banco di mixaggio "universale").

Negli anni Sessanta-Settanta il centro si afferma più anche come punto d'incontro fra arte e tecnologia. Boris Blacher, Herbert Brün, ospiti internazionali, produzioni spaziali, concerti, processi sperimentali di suono: tutto ciò fa crescere la reputazione oltre la Germania.

Folkmar Hein, che dirige lo Studio Elettronico per molti anni, incarna questo equilibrio fra la ricerca tecnica e il fare artistico: non solo costruzione e uso di tecnologie audio, ma anche organizzazione di festival, scambi, documentazione, creazione del database EMDoku per la musica elettroacustica.

Oggi la TU Berlin mantiene viva questa eredità con programmi come Audio Communication & Sonology, centri di ricerca per l'audio spaziale, laboratori di ascolto, camere acustiche,

spazi per la produzione artistica e tecnologica. È un posto dove la tecnologia del suono non è solo strumento tecnico: diventa forma di espressione, oggetto di studio, ma anche esperienza concreta.

1.2 Audio Communication Group e Computer Music and Neural Audio Communication Group

L'Audio Communication Group della Technische Universität Berlin è un centro di ricerca interdisciplinare dedicato allo studio del suono e della comunicazione audio. Il gruppo integra approcci ingegneristici con prospettive umanistiche, psicologiche, culturali e sociali, affrontando il suono non solo come fenomeno fisico, ma anche come esperienza percettiva, sociale e artistica, progettando quindi anche opere di sound art e media art, nonché di sistemi interattivi basati sul suono.

Tra gli ambiti principali di ricerca rientrano lo sviluppo e l'applicazione di tecnologie per la realtà acustica virtuale, tra cui sintesi binaurale, ambisonics, spatial audio rendering e wave field synthesis, che non solo permettono la creazione di ambienti sonori immersivi, ma vengono anche impiegate per studiare la percezione musicale, la psicologia dei media, l'acustica architettonica e la cultura sonora contemporanea.

Il gruppo è guidato dal Prof. Dr. Stefan Weinzierl, esperto in acustica musicale e realtà virtuale acustica, e dispone di laboratori tra i più avanzati al mondo, che comprende studi audio multicanale, un media lab immersivo, camere anecoiche e riverberanti, nonché il più esteso sistema di wave field synthesis attualmente esistente (oltre 2700 altoparlanti, 832 canali). Questo ecosistema sperimentale consente lo sviluppo e la validazione di tecnologie audio neurali in ambienti controllati e altamente reattivi.

L'Audio Communication Group della TU Berlin rappresenta quindi un punto d'incontro tra ricerca scientifica, pratica artistica e riflessione critica sul suono, ponendosi all'avanguardia nello studio e nello sviluppo di nuove modalità per vivere, comprendere e progettare l'esperienza acustica.

Il Computer Music and Neural Audio Communication Group della Technische Universität Berlin è un'unità di ricerca specializzata nell'esplorazione di modelli computazionali applicati al suono, con particolare attenzione alle applicazioni in tempo reale nel contesto della musica elettronica, dell'audio spaziale e dell'interazione sonora, e svolge ricerca all'interno dell'Audio Communication Group della TU Berlin, in collaborazione con altri team come il Computational Musicology Team.. Il gruppo si occupa dello sviluppo di algoritmi per la sintesi, la modellazione

e la trasformazione del suono, integrando tecniche di elaborazione del segnale digitale (DSP) con metodologie di apprendimento automatico, in particolare reti neurali profonde.

Le attività di ricerca si concentrano su soluzioni teoriche e pratiche per l'interazione musicale uomo-macchina, mirando alla progettazione di sistemi sonori intelligenti e reattivi. In quest'ambito, particolare rilievo assume l'impiego di modelli neurali nella spazializzazione del suono (ambisonics, wave field synthesis), finalizzati alla generazione di ambienti sonori immersivi adattivi. Parallelamente, il gruppo indaga le potenzialità delle performance musicali distribuite in rete, sviluppando sistemi capaci di integrare input gestuali, spaziali e sonori in tempo reale.

Il suo lavoro si colloca all'incrocio tra ricerca scientifica e sperimentazione artistica, con l'obiettivo di offrire strumenti innovativi a musicisti, compositori e sound designer.

Esempi concreti dell'approccio adottato includono il progetto Scyclone, vincitore della Neural Audio Plug-in Competition 2023, e la presentazione Real□Time Neural Network Inference: A Practical Approach for DSP Engineers tenuta all'Audio Developer Conference 2023, testimonianze di un orientamento fortemente applicativo e sperimentale.

Sul piano formativo, il gruppo coordina il corso di laurea magistrale in Audio Communication and Technology e partecipa alla didattica in altri programmi interdisciplinari legati all'ingegneria del suono, all'acustica e al design computazionale.

Il tirocinio ha avuto inizio il 3 marzo 2025 e si è concluso il 3 giugno dello stesso anno.

1.3 Motivazione della scelta del tirocinio

L'intenzione di iniziare il tirocinio presso il Computer Music and Neural Audio Communication Group si è manifestata in seguito all'insorgere del bisogno personale di approfondire gli ambiti di ricerca di una disciplina che fosse un ibrido tra ingegneria e musica. Questa esigenza deriva da una forte passione personale per la musica, per i modi e le tecnologie con cui il suono si trasmette, per l'originalità della ricerca in questi campi e per il suono in generale, dal punto di vista sia fisico che della sua elaborazione. Le attività svolte durante il tirocinio sono perfettamente in linea con il percorso universitario antecedente e soprattutto futuro, dal momento che la feature engineering viene spesso combinata con l'ingegneria informatica, ad esempio per modellare dati che verranno usati da un modello, che è esattamente il progetto che fa capo alla materia trattata in tale tesi.

Capitolo 2

Obiettivi

2.1 Obiettivo generale: emulazione del plate reverb

Il plate reverb è stato introdotto negli anni Cinquanta e, per diversi decenni, ha rappresentato la soluzione di riferimento per ottenere riverbero artificiale di alta qualità, fino a quando la tecnologia digitale non ha iniziato a diffondersi. Il principio di funzionamento è relativamente semplice: un ampio pannello metallico viene messo in vibrazione da un trasduttore, ossia un dispositivo che converte un segnale elettrico in vibrazioni meccaniche. Queste vibrazioni si propagano lungo la piastra e vengono poi rilevate da un secondo trasduttore, in questo caso di tipo inverso, che converte le vibrazioni nuovamente in segnale elettrico.

Rispetto alla riverberazione naturale prodotta da un ambiente fisico, il plate reverb ha caratteristiche sonore peculiari. Innanzitutto mancano riflessioni iniziali (early reflections), ovvero quei primi rimbalzi del suono sulle superfici che, in un ambiente reale, giungono all'ascoltatore immediatamente dopo il suono diretto e contribuiscono alla percezione dello spazio. Inoltre, la piastra presenta una distribuzione quasi uniforme delle frequenze di risonanza, cioè delle frequenze alle quali il materiale tende a vibrare più facilmente, producendo un decadimento sonoro omogeneo.

Nonostante queste differenze rispetto a un ambiente reale, il timbro ricco e omogeneo del plate reverb è diventato negli anni uno degli effetti più apprezzati e ricercati, tanto da essere ancora oggi riprodotto anche in formato digitale nelle produzioni musicali.

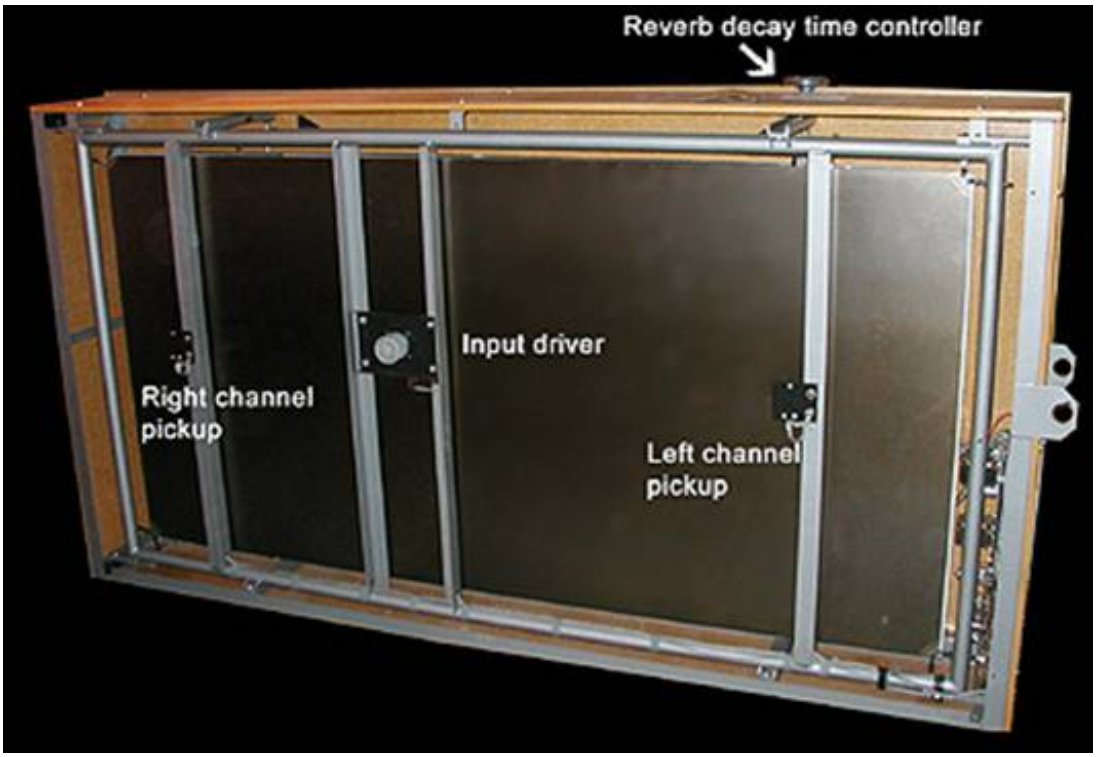


Figura 2.1: Funzioni del Plate Reverb.

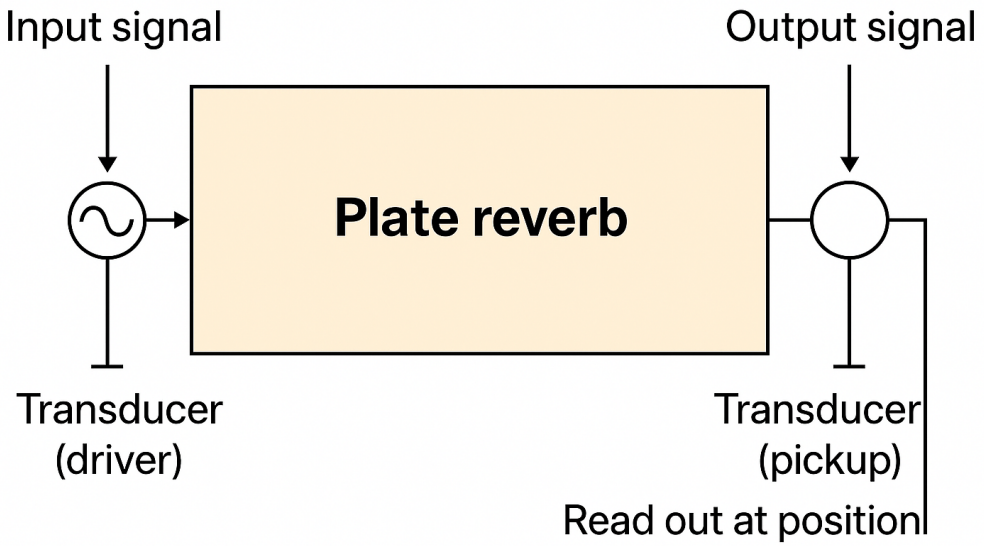


Figura 2.2: Schema di funzionamento del Plate Reverb.

Figura 2.3: Fotografie del Plate Reverb presso Elektronischen Studio, Technische Universität Berlin.

Figura 2.4: Progettazione del sistema di trasduzione per una coppia di unità di Plate Reverb sviluppate per l'installazione *Glaring Sounds* degli artisti Marta De Pascalis e Filippo Vogliazzo.

2.2 Attività principali

Ai fini dell'addestramento del modello di intelligenza artificiale finalizzato all'emulazione del plate reverb, si è ritenuto opportuno predisporre due insiemi di dati distinti: un dataset omogeneo e un dataset eterogeneo, ciascuno con una funzione complementare.

Il dataset omogeneo è stato costruito a partire da segnali acustici semplici e controllati, quali sweep di frequenze, toni sinusoidali e impulsi percussivi. La ridotta complessità di tali stimoli ha consentito di evidenziare con maggiore chiarezza le proprietà acustiche intrinseche del plate reverb, come il tempo di decadimento e la distribuzione delle risonanze, facilitando l'apprendimento stabile delle sue caratteristiche fondamentali.

Il dataset eterogeneo, invece, comprendeva segnali di natura più varia e realistica, tra cui voci, strumenti musicali e field recordings, quindi rumori complessi. L'impiego di tale materiale ha avuto lo scopo di migliorare la capacità di generalizzazione del modello, permettendogli di applicare correttamente le caratteristiche apprese del plate ad una gamma più ampia di suoni.

L'utilizzo congiunto dei due dataset ha quindi consentito di ottenere un equilibrio tra precisione nell'apprendimento delle caratteristiche distintive del riverbero e flessibilità nell'adattarle a contesti sonori eterogenei, ponendo così una base solida per il successivo processo di training del modello neurale.

Per caratterizzare i segnali raccolti e costruire un dataset adatto all'addestramento, sono state estratte diverse feature acustiche mediante la libreria Librosa in ambiente Python. Questi parametri descrivono aspetti differenti del contenuto sonoro e permettono al modello di IA di apprendere con maggiore efficacia il comportamento del plate reverb.

Le features selezionate sono:

RMS (Root Mean Square Energy): misura l'energia del segnale, utile a descrivere l'andamento dinamico del suono.

Spectral Bandwidth: misura la dispersione spettrale attorno al centroid, utile a valutare l'ampiezza in frequenza.

Spectral Centroid: rappresenta il "baricentro" dello spettro e descrive la brillantezza timbrica del suono.

Spectral Flatness: misura il rapporto tra media geometrica e media aritmetica dello spettro, utile a distinguere suoni tonali da suoni rumorosi.

Spectral Rolloff High: frequenza oltre la quale rimane solo una piccola percentuale di energia (es. 5%), utile per analizzare le alte frequenze.

Spectral Rolloff Low: frequenza sotto la quale si concentra una parte significativa dell'energia (es. 85–95%), indicativa delle basse frequenze.

Questi parametri costituiscono una prima rappresentazione acustica e computazionale del materiale raccolto. Una discussione più approfondita sul significato di ciascuna feature e sulla loro pertinenza rispetto all'emulazione del plate reverb sarà proposta nei capitoli successivi.

2.3 Modalità di lavoro e strumenti utilizzati

Per assemblare il dataset, come ad esempio selezionare quantità uniformi di file audio da ogni pool di datasets, non erano necessari particolari script, dal momento che il lavoro avrebbe potuto comunque essere svolto a mano, ma con un impiego di tempo considerevolmente maggiore. Per l'analisi di features invece non si poteva far altro che usare degli script (Python in particolare), dal momento che la features analysis è stata gestita con Librosa, una libreria disponibile soltanto con il linguaggio di programmazione Python. In seguito verrà elencato e spiegato ogni script, ognuno reperibile al relativo GitHub.

Team leader del team Computer Music and Neural Audio Communication Group nel periodo in cui è stato assemblato il dataset e fatta la features analysis, come solito, era Fares Schulz. Altri collaboratori e attuali membri del gruppo sono Christian Limberg e Leonardo Auri, mentre gli altri sono in continuo divenire.

Il progetto del dataset è stato sottoposto a supervisione ogni tre settimane nelle riunioni del gruppo di ricerca dove tutti i membri presentavano i progressi fatti nel tempo intercorso tra ogni riunione, mentre gli altri esponevano eventuali perplessità o punti critici a riguardo. Nel frattempo Fares Schulz è comunque stato consultato per dirigere il lavoro e fornire consigli per migliorare l'operato.

2.4 Criticità incontrate

Come prima cosa è stato fornito un account Cloudy per coordinare il lavoro di organizzazione con i membri del gruppo di ricerca quali riunioni periodiche, eventi importanti per tutta la comunità di acustica ed audio engieering della TU Berlin, impegni dei singoli membri che avrebbero inciso sulla comunità, ferie, ecc., e soprattutto per prendere appunti e dare direttive riguardo tasks da portare a termine, il che comprende coordinare il lavoro dei tirocinanti: Cloudy permette infatti di gestire un file system su cloud.
Ad ogni riunione venivano fornite nuove direttive sulla base dell'aggiornamento delle tasks che ciascun membro del CMNAS stava conducendo, e ci si scambiavano pareri riguardo i lavori degli altri membri ed eventuali note positive. Gli appunti presi durante l'incontro venivano poi salvati su Cloudy, all'interno delle cartelle dei rispettivi meeting e associati a ciascun membro, oppure annotati direttamente sul computer dell'interessato.

In una prima fase è stata raccolta una lista di pagine online contenenti elenchi di dataset, predisposta dall'Elektronischen Studio, con l'obiettivo di includere il maggior numero possibile di tipologie differenti, così da ottenere un insieme eterogeneo.

La selezione dei dataset si è basata su criteri specifici: disponibilità del relativo paper corredato da un riferimento bibliografico univoco (es. DOI o numero arXiv): presenza di licenze di tipo permissivo, escluse quelle che vietano la creazione di opere derivate o la loro eventuale commercializzazione, dal momento che il gruppo di ricerca ha espresso la possiblità di sviluppare e commercializzare un VST addestrato su tali dati (pertanto sono stati esclusi, ad esempio, tutti i dataset con licenza CC BY-NC 4.0 e derivate).

Per quanto riguarda le caratteristiche tecniche, sarebbe stato meglio se i dataset considerati avessero avuto direttamente un sample rate di 48kHz e una bit depth di 32 bit, in modo da non dover dedicare troppo tempo a fare il resample eventuale dei file audio non idonei (ci si riferisca al file CSV per i valori originali), dal momento che appunto alla fine avremmo dovuto avere un dataset eterogeneo il cui sample rate fosse 48kHz e con una bit depth di 32 bit.

Infine, durante la raccolta, eventuali dataset omogenei con soli suoni percussivi avrebbero dovuto essere stati messi da parte per il futuro addestramento del R.A.V.E. model, approfondito in seguito.

Durante la fase di selezione, i file individuati per l'inclusione nel dataset sono stati progressivamente caricati all'interno della cartella dedicata sul server remoto marckh della TU Berlin, il cui accesso diretto non è disponibile per ragioni di sicurezza. Nella stessa cartella

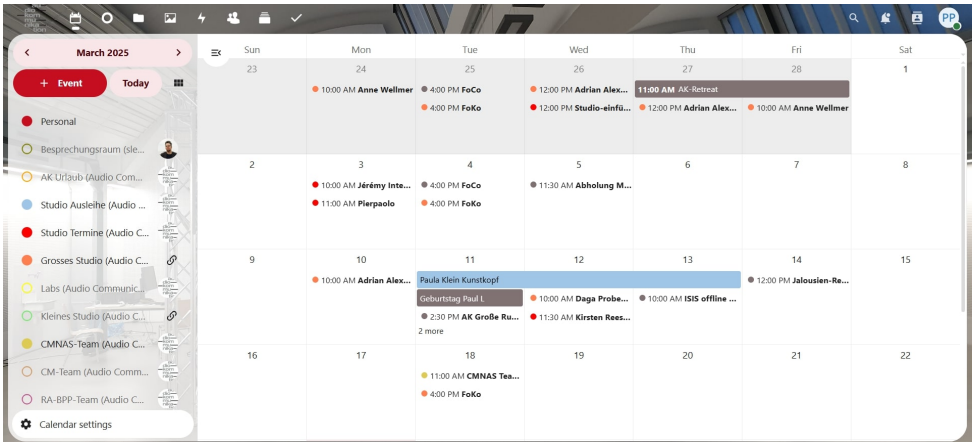


Figura 2.5: Screenshot del calendario Cloudy.

Figura 2.6: Screenshots di un file Cloudy per le direttive.

sono stati collocati gli script per l'elaborazione delle feature e, per ciascun dataset selezionato, i file README all'interno delle rispettive sottocartelle. La struttura complessiva è disponibile alla relativa pagina GitHub.

I siti elenchi di datasets sono i segeuenti:

- Magenta Datasets
- AI Audio Datasets (GitHub)
- ISMIR Datasets
- 40 Open Source Audio Datasets (Towards Data Science)
- Alexander Lerch – Audio Dataset List
- HuggingFace Audio Datasets
- Kaggle Audio Datasets

Il bilanciamento è risultato relativamente semplice, poiché l'obiettivo principale era garantire l'eterogeneità piuttosto che la presenza di categorie specifiche. L'approccio più efficace si è rivelato quello di selezionare casualmente i file. Tale operazione è stata realizzata tramite lo script `extract_and_filter.py` progettato per estrarre in maniera casuale ed equa i campioni audio da ciascun dataset.

In una fase preliminare, i link dei dataset selezionati da diverse fonti sono stati raccolti in un documento di riferimento (Google Doc), successivamente ordinati e catalogati in una tabella (Google Spreadsheet) per una migliore organizzazione.

Infine, tutti i dataset individuati sono stati acquisiti e da ciascuno sono stati selezionati alcuni file audio, con l'obiettivo appunto di costruire un unico dataset eterogeneo.

Le linee guida prevedevano la costruzione di un dataset della dimensione massima di 100 GB. Al momento della selezione dei materiali da ciascun dataset, si è scelto di mantenere, nei limiti del possibile, una quantità di dati uniforme per ogni sorgente. Un criterio proporzionale alla dimensione complessiva di ciascun dataset avrebbe infatti portato a risultati squilibrati: i dataset più grandi sarebbero stati sovrarappresentati, mentre quelli più piccoli avrebbero contribuito con una quantità trascurabile di campioni. Poiché l'obiettivo principale era garantire la massima eterogeneità, è stato ritenuto più opportuno attribuire lo stesso peso a ciascun dataset, indipendentemente dalla sua estensione originaria.

Poiché diversi dataset presentavano una dimensione inferiore al valore medio teorico di 3,7 GB (ottenuto dividendo i 100 GB complessivi per i 27 dataset totali), si è scelto di includere integralmente tutti quelli più piccoli, alcuni dei quali pari a pochi centinaia di megabyte o a circa 2 GB, e solo successivamente fare la divisione tra i dataset di dimensione maggiore, cioè quelli che rimanevano. La somma di tali dataset, comprensiva anche di annotazioni e metadata dal peso marginale, ammontava a circa 20 GB. Di conseguenza, la capacità residua è stata ripartita tra i dataset di dimensioni maggiori, ossia quelli eccedenti i 3,7 GB.

Il calcolo è il seguente:

$$Q = \frac{100\,\text{GB} - 20\,\text{GB}}{27 - 12} \approx 5,3\,\text{GB}$$

dove:
Q = quantità di dati da selezionare per ciascun dataset di grandi dimensioni;
27 = numero totale di dataset;
12 = numero di dataset con dimensione inferiore a 3,7 GB;
100 GB = dimensione massima complessiva del dataset eterogeneo;
20 GB = spazio occupato dai 12 dataset più piccoli.

Si è evidenziato quindi un limite di circa 5,3 GB da prelevare per ciascun dataset grande, garantendo così un equilibrio nella distribuzione complessiva dei dati.

In fase di verifica sono emerse discrepanze tra le dimensioni dichiarate nelle anteprime dei dataset e quelle effettivamente riscontrate dopo il download. In alcuni casi i valori indicati comprendevano anche file video o altri formati non pertinenti, determinando un aumento rispetto alle dimensioni dei soli file audio in formato wav. Ciò ha reso necessario ricalcolare la suddivisione dei dati, ridefinendo la quantità da prelevare da ciascun dataset. Di conseguenza, è stato necessario riscaricare parte dei dataset già utilizzati, al fine di integrare la differenza mancante tra i 5,3 GB inizialmente previsti e i nuovi valori risultanti dal ricalcolo.

L'acquisizione effettiva dei file ha evidenziato una dimensione complessiva di 11,774 GB, inferiore ai circa 20 GB inizialmente stimati. A tale valore sono stati aggiunti ulteriori dataset di dimensioni ridotte (1,87 GB, 711 MB e 1,34 GB), resi includibili grazie allo spazio disponibile superiore alle previsioni. La capacità residua, pari a 84,305 GB, è stata quindi suddivisa tra i dataset più grandi, ovvero quelli eccedenti i 5,3 GB di dimensione.

Poichè uno dei dataset è stato escluso a causa della discrepanza tra la licenza dichiarata e quella effettivamente applicata al materiale scaricato, il numero dei dataset di grandi dimensioni considerati si riduce da 12 a 11. La ripartizione risulta quindi:

$$Q = \frac{100\,\text{GB} - 11,774\,\text{GB} - 1,87\,\text{GB} - 0,711\,\text{GB} - 1,34\,\text{GB}}{27 - 11 - 3} \approx 6,485\,\text{GB}$$

dove:
Q = nuova quantità di dati da selezionare per ciascun dataset di grandi dimensioni;
27 = numero totale di dataset;
11 = numero di dataset di piccole dimensioni già inclusi;
3 = ulteriori dataset esclusi dal calcolo;
100 GB = dimensione massima complessiva del dataset eterogeneo;
11.774 GB = spazio complessivo effettivo degli 11 dataset più piccoli;
1.87 GB, **0.711 GB**, **1.34 GB** = dimensioni effettive di alcuni dataset piccoli già conteggiati.

ha portato a un valore di circa 6,485 GB per ciascun dataset, rendendo necessario integrare circa 1,1 GB aggiuntivi rispetto ai 5,3 GB precedentemente selezionati. Per raggiungere questo nuovo equilibrio è stato quindi necessario riscaricare i dataset con dimensioni pari o superiori a 6,3 GB, da cui è stato prelevato 1 GB aggiuntivo di materiale audio.

Questa scelta metodologica non era l'unica possibile: si sarebbero potuti mantenere i dataset già ridotti a 5,3 GB e distribuire l'eccedenza soltanto tra i rimanenti, portandoli a circa 8 GB anzichè 6,485 GB. Tuttavia, tale approccio avrebbe comportato una maggiore disparità nella distribuzione complessiva.

Poiché l'obiettivo primario era massimizzare l'eterogeneità, si è ritenuto preferibile uniformare la quantità di dati selezionati, anche a costo di un ricalcolo complessivo. Inoltre, pur non essendo possibile ottenere una perfetta parità (dato che alcuni dataset rimangono inferiori a 6,485 GB), ridurre al minimo le differenze è stato considerato un criterio metodologicamente più coerente con le finalità del progetto.

Capitolo 3

Costruzione del Dataset e Feature Extraction

3.1 Il plate reverb: caratteristiche acustiche e rilevanza musicale

Come scritto prima, il plate reverb è un dispositivo elettromeccanico progettato per generare riverberazione attraverso le vibrazioni di una piastra metallica, generalmente in acciaio, mantenuta in tensione all'interno di un telaio. Un trasduttore elettroacustico (driver) converte il segnale elettrico in vibrazioni meccaniche sulla piastra, mentre uno o più pickup, collocati in punti differenti della superficie, trasformano nuovamente tali vibrazioni in segnale elettrico, producendo una o più uscite. La posizione reciproca di driver e pickup risulta determinante, poiché zone diverse della piastra presentano combinazioni proprie di modi di vibrazione: da ciò deriva una parziale decorrelazione del segnale captato in punti distanti, aspetto particolarmente utile per l'elaborazione stereo.

Il sistema è generalmente dotato di un meccanismo di smorzamento regolabile (ad esempio pannelli o feltri avvicinati alla piastra), che consente di variare la durata del decadimento del riverbero. La tensione applicata alla piastra e le condizioni di vincolo ai bordi influenzano la densità dei modi e la distribuzione delle frequenze di risonanza. Rispetto a un ambiente fisico reale, la risposta della piastra si caratterizza per una maggiore uniformità spettrale alle medie–alte frequenze e per l'assenza di riflessioni precoci marcate. Il risultato è un decadimento regolare, con elevata densità di code e una percezione di brillantezza, dovuta alla minore dissipazione energetica alle alte frequenze (entro i limiti imposti dallo smorzamento).

Dal punto di vista operativo, i parametri di maggiore rilevanza sono:

tempo di decadimento (RT), controllabile tramite smorzamento;

colore timbrico, influenzato da tensione, materiale e spessore della piastra;

ampiezza stereo, ottenuta grazie a pickup multipli;

rapporto segnale/rumore, con la presenza di componenti meccaniche ed elettriche residue;

non-linearità marginali, dovute alla saturazione del driver a livelli elevati, percepite come parte integrante del carattere sonoro dell'unità.

L'interesse verso l'emulazione digitale del plate reverb è motivato da considerazioni di natura sia artistica sia pratica.

Dal punto di vista artistico, il plate è apprezzato per il suo timbro peculiare: un alone denso, continuo e brillante, che conferisce maggiore presenza e sustain a voci e strumenti (in particolare voce solista, rullante, chitarra elettrica, pianoforte e archi), senza evocare esplicitamente uno spazio architettonico reale. Tale carattere estetico si è consolidato come standard nella produzione discografica e rimane tuttora ricercato.

Dal punto di vista pratico, i plate analogici presentano alcune criticità: dimensioni ingombranti, costi elevati, necessità di manutenzione e taratura (in particolare tensione della piastra, allineamento dei trasduttori e regolazione dello smorzamento). L'emulazione digitale consente di superare tali limiti, offrendo i seguenti vantaggi:

conservazione e accessibilità del timbro caratteristico in contesti contemporanei (DAW, live processing, produzione mobile);

richiamo immediato delle impostazioni, stabilità e ripetibilità tra sessioni di lavoro;

disponibilità di controlli parametrici avanzati (decadimento, colore timbrico, ampiezza stereo, eventuale pre-delay);

riduzione dei costi e semplificazione logistica, mantenendo, quando l'emulazione è accurata, le qualità sonore considerate desiderabili.

Nel contesto della presente tesi, l'emulazione è sviluppata a partire dalla caratterizzazione acustica del comportamento della piastra, mediante un dataset dedicato e descrittori audio mirati. L'obiettivo è riprodurre digitalmente la "firma sonora" del plate analogico, coniugando fedeltà timbrica e funzionalità operativa in ambienti di produzione moderni.

3.2 Raccolta e trattamento dei segnali audio

Per la costruzione del dataset eterogeneo non ci si è limitati a raccogliere e ordinare materiali sonori, ma si è sviluppata un'infrastruttura metodologica che lo rendesse uno strumento operativo flessibile e riutilizzabile. La collezione è stata suddivisa in quattro macro-categorie: *parlato*,

registrazioni ambientali, *brani musicali completi* e *strumenti isolati*. L'insieme complessivo ammonta a circa 91 GB di materiale audio e costituisce la base per l'addestramento del modello di riverbero.

A supporto dell'organizzazione, sono stati prodotti diversi file riepilogativi (CSV ed Excel) contenenti metadati e informazioni tecniche, in modo da documentare la provenienza dei dati, monitorarne la struttura e fornire input ai successivi strumenti di analisi automatica.

Un passaggio importante ha riguardato la normalizzazione tecnica del materiale sonoro. I file originari provenivano da dataset differenti, con caratteristiche di campionamento e formati non uniformi. Per garantire coerenza, è stata creata una versione parallela del dataset convertita in maniera uniforme a 48 kHz e bit depth a 32 bit, che ha rappresentato lo standard di riferimento per le fasi di analisi e feature extraction. Contestualmente, sono state realizzate rappresentazioni statistiche (istogrammi e distribuzioni) utili a valutare la ripartizione delle principali feature acustiche, sia a livello di dataset singoli sia a livello di categorie.

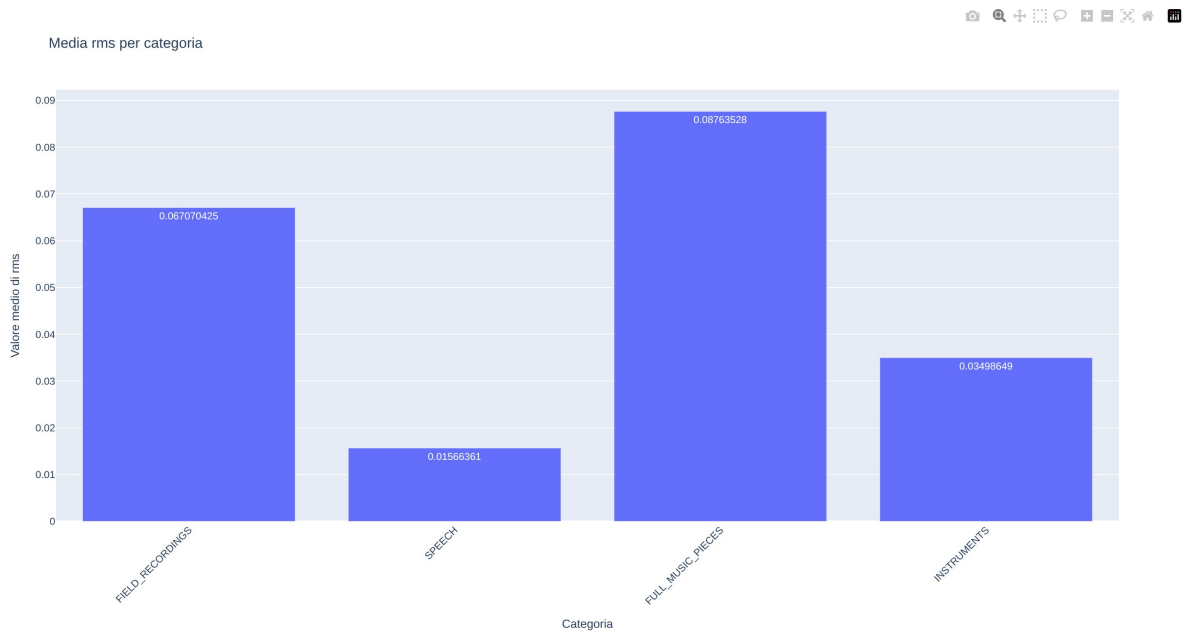
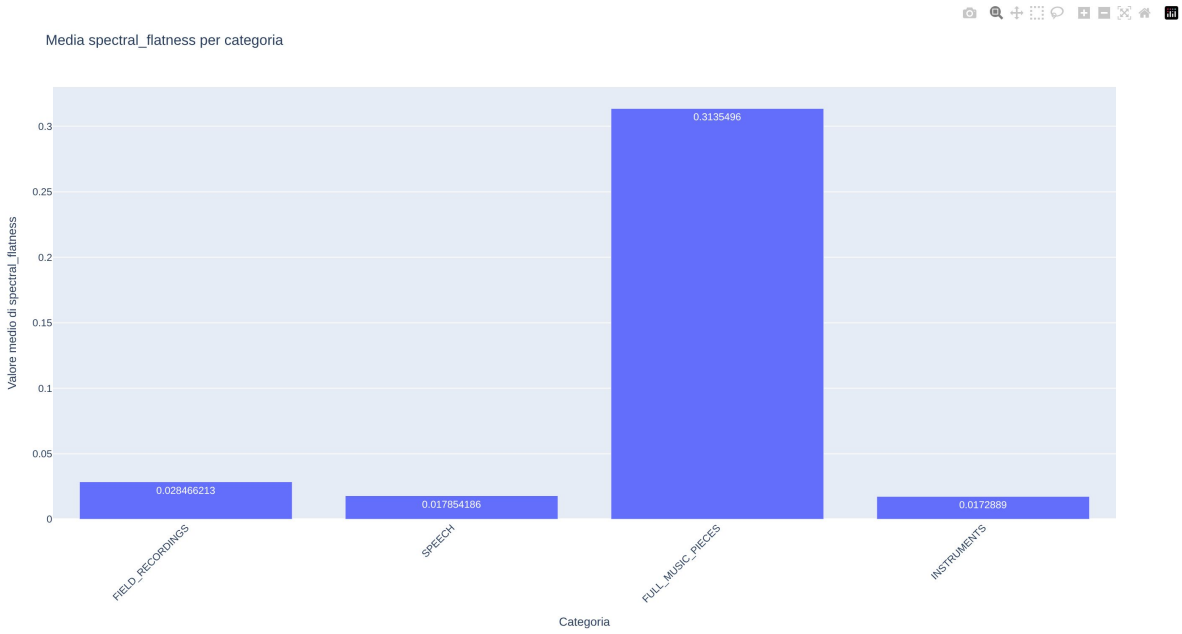
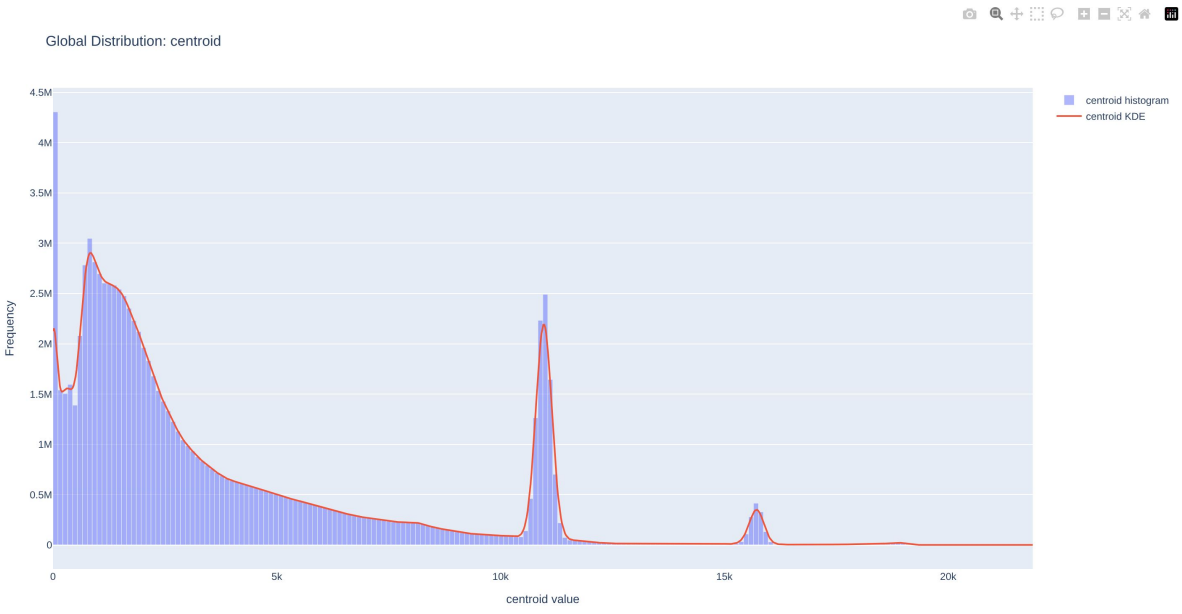


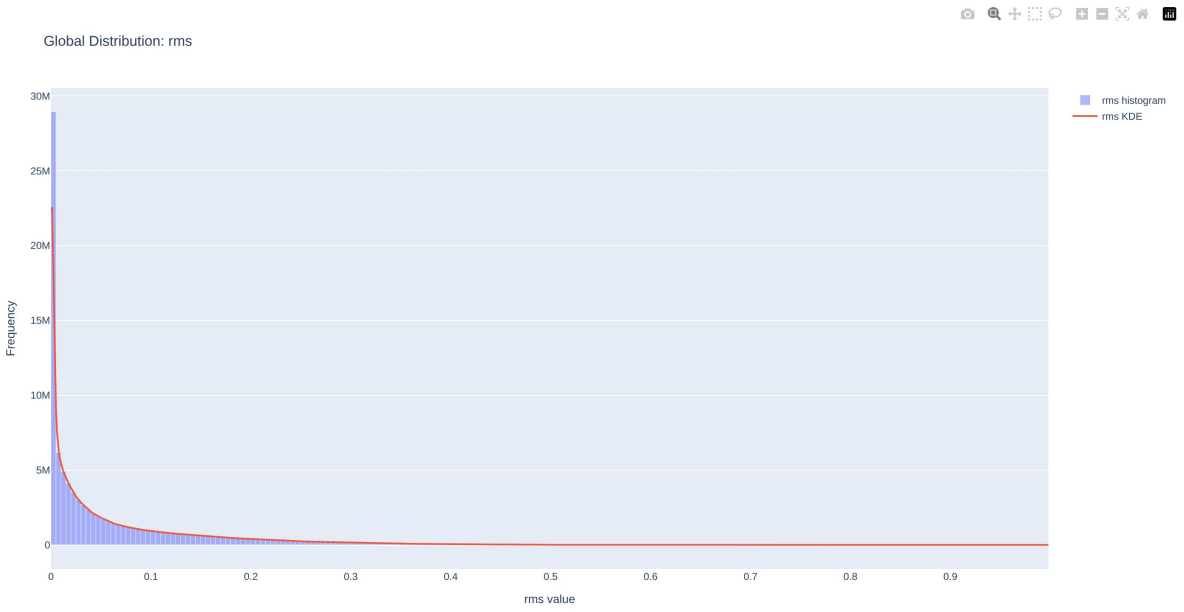
Figura 3.1: Media RMS per categoria.



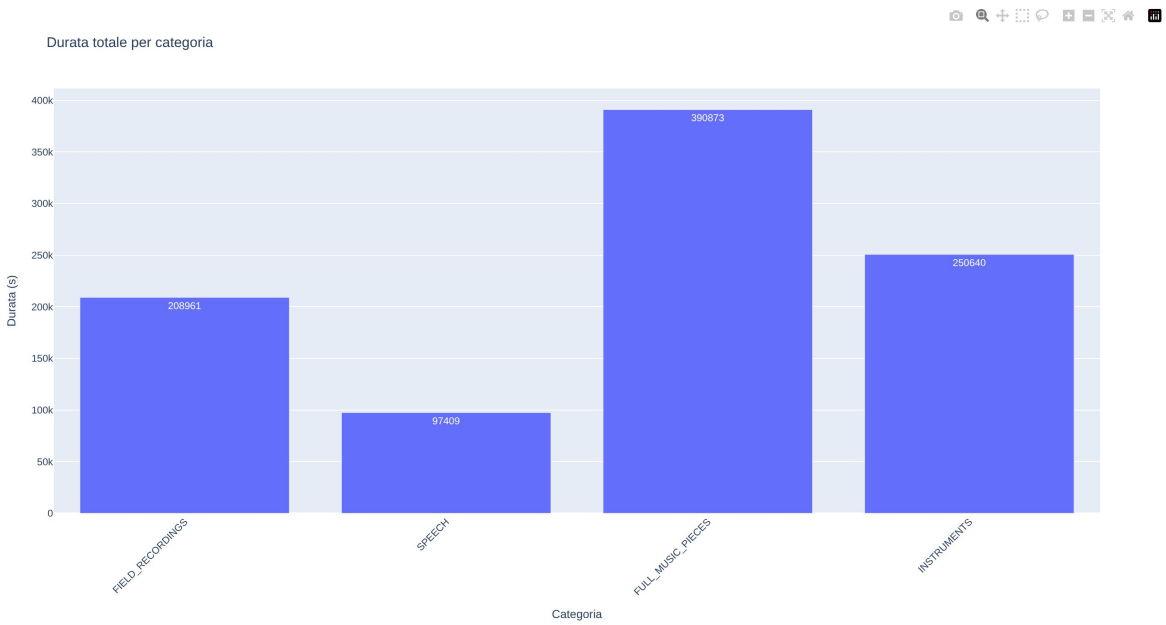
(a) Media spectral flatness per categoria.



(b) Global centroid distribution.



(a) Global Distribution: RMS.



(b) Durata totale per categoria.

Per rendere più agevole la gestione dell'archivio è stata predisposta una suite di script Python, organizzata in più moduli funzionali:

dataset_dimension_tools

[label=·]`dataset_reduction_by_dimension.py` = seleziona casualmente un sottoinsieme di file da una cartella sorgente, mantenendo un bilanciamento tra le sottocartelle, fino a raggiungere una dimensione totale specificata dall'utente (in GB), e poi li copia in una cartella di destinazione mantenendo la stessa struttura di directory; `dataset_reduction_by_length.py` = seleziona casualmente un sottoinsieme di file da una cartella sorgente, mantenendo un bilanciamento tra le sottocartelle, fino a raggiungere una durata totale specificata dall'utente (in ore), e poi li copia in una cartella di destinazione mantenendo la stessa struttura di directory.

features_tools

[label=·]`extract_and_filter.py` = estrae file audio mantenendo la struttura di directory originale, sulla base degli input forniti dall'utente, che sono: percorso dei file ZIP, cartella di destinazione e limite massimo per i file audio (in MB); `extract_files_by_size.py` = seleziona e copia un sottoinsieme di file da una directory sorgente a una directory di destinazione, preservando la struttura originale delle cartelle. I file vengono scelti in modo equilibrato tra le sottocartelle fino al raggiungimento di una dimensione totale specificata (in GB). Viene inoltre generato un log dettagliato che elenca i file copiati e la dimensione complessiva copiata.

features_tools

[label=·]`features_distributions_category.py` = analizza il dataset di file audio, le caratteristiche acustiche e l'interazione grafica delle loro statistiche di distribuzione. In particolare, le caratteristiche considerate sono: RMS, spectral centroid, spectral bandwidth, spectral flatness, spectral rolloff high, spectral rolloff low; mentre le distribuzioni sono: distribuzione di ogni caratteristica per ciascun dataset all'interno di una categoria (normalizzata), distribuzione di ogni caratteristica per ciascuna categoria (normalizzata), distribuzione per l'intero dataset (normalizzata); `features_histograms_category.py` = analizza il dataset di file audio, le caratteristiche acustiche e l'interazione grafica delle loro statistiche. In particolare, le caratteristiche considerate sono: RMS, spectral centroid, spectral bandwidth, spectral flatness, spectral rolloff high, spectral rolloff low; mentre gli istogrammi riguardano ciascuna categoria in termini di durata totale e la media di ogni caratteristica all'interno di una categoria. Lo script richiede un file CSV su cui effettuare

dei calcoli e salverà gli istogrammi in una cartella chiamata `histograms_per_category`. `features_histograms_dataset.py` = analizza il dataset di file audio, le caratteristiche acustiche e l'interazione grafica delle loro statistiche. In particolare, le caratteristiche considerate sono: RMS, spectral centroid, spectral bandwidth, spectral flatness, spectral rolloff high, spectral rolloff low; mentre gli istogrammi riguardano ciascun dataset in termini di durata e la media di ogni caratteristica all'interno di un dataset. Lo script richiede un file CSV su cui effettuare dei calcoli e salverà gli istogrammi in una cartella chiamata `histograms_per_dataset`.

generic_tools

[label=·]`clear_audio.py` = pulisce una cartella (e tutte le sue sottocartelle) eliminando tutti i file che non hanno estensioni specifiche (estensioni permesse = `*.wav`, `.flac`); `remove_annotations.py` = elimina ricorsivamente tutte le cartelle chiamate `annotation` all'interno di una struttura di directory; `resampling_flac.py` = converte ricorsivamente tutti i file `.flac` in una cartella (e sottocartelle) in un nuovo formato `.flac` a 48 kHz e 32-bit PCM, utilizzando ffmpeg; `resampling_no_flac.py` = converte ricorsivamente tutti i file (eccetto i `.flac`) in una cartella (e sottocartelle) in un nuovo formato a 48 kHz e 32-bit PCM, utilizzando ffmpeg; `search_readme_zip.py` = cerca ed estrae eventuali file `README` da un archivio `.zip` specificato dall'utente; `categories_csv.py` = analizza le categorie di file audio (`WAV` e `FLAC`) organizzati in sottocartelle e produce un file `.csv` di riepilogo con statistiche tecniche e audio-spettrali per ciascuna categoria; `datasets_csv.py` = analizza un dataset di file audio (`WAV` e `FLAC`) organizzati in sottocartelle e produce un file `.csv` di riepilogo con statistiche tecniche e audio-spettrali per ciascuna categoria.

L'insieme di questi strumenti rende il dataset non una semplice raccolta di file, ma un vero e proprio ecosistema operativo. Esso può essere selezionato, ridotto, analizzato e rappresentato visivamente tramite procedure standardizzate, trasformandosi così in una piattaforma di sperimentazione metodologica oltre che in una base per l'addestramento del modello.

3.3 Feature extraction con Python e Librosa

Nella fase di analisi è stato necessario individuare un insieme di descrittori acustici in grado di rappresentare in maniera efficace il comportamento del plate reverb. Le feature selezionate coprono aspetti energetici, spettrali e timbrici del segnale.

Le features considerate sono le seguenti:

- **RMS (Root Mean Square Energy)**: rappresenta una misura statistica dell'energia di un segnale nel tempo, calcolata come la radice quadrata della media dei valori al quadrato dell'ampiezza. È strettamente legata alla percezione umana della loudness, dato che l'orecchio non risponde all'ampiezza istantanea ma a una forma di energia integrata. La RMS è uno dei descrittori fondamentali per la caratterizzazione timbrica e dinamica di un suono. In contesti musicali, un valore RMS elevato corrisponde a segnali più intensi, mentre valori bassi indicano regioni di silenzio o decadimento sonoro.

- **Spectral Centroid**: spesso definita come il "baricentro" dello spettro, è calcolata come media pesata delle frequenze in base alla loro ampiezza. È stata ampiamente correlata alla percezione di brillantezza timbrica: suoni con una centroid alta sono percepiti come chiari e brillanti, mentre centroid basse corrispondono a timbri cupi. Tzanetakis e Cook (2002) la utilizzano come feature primaria nella classificazione automatica di generi musicali, dimostrando la sua importanza nel distinguere strumenti e stili.

- **Spectral Bandwidth**: misura la dispersione delle frequenze attorno al centroide. È un indice della "pienezza" o complessità timbrica: uno spettro con banda stretta indica un suono tonale puro, mentre una banda larga suggerisce suoni ricchi di armoniche o rumorosi. Il bandwidth svolge un ruolo cruciale nella discriminazione tra strumenti musicali con contenuti armonici differenti.

- **Spectral Flatness**: definita come il rapporto tra media geometrica e media aritmetica dello spettro, quantifica quanto lo spettro si avvicini a quello di un rumore bianco. Valori bassi (vicini a 0) indicano uno spettro dominato da poche componenti armoniche (suono tonale), mentre valori alti (vicini a 1) descrivono suoni rumorosi. La flatness è strettamente collegata alla percezione di tonalità vs rumore e si rivela una feature cruciale nella sintesi e classificazione sonora.

- **Spectral Rolloff (High e Low)**: indica la frequenza sotto o sopra la quale è contenuta una certa percentuale dell'energia spettrale. In genere si usa un valore tra l'85% e il 95% (*low rolloff*) per descrivere l'estensione verso le basse frequenze, e un valore intorno al 5% (*high rolloff*) per descrivere la coda verso le alte frequenze. Questa misura fornisce un'indicazione robusta della distribuzione energetica dello spettro. L'uso congiunto dei due rolloff consente di definire la "cornice" energetica dello spettro e di distinguere, ad esempio, la brillantezza percepita di strumenti a corda rispetto a percussioni o fiati, cioè la clssificazione di timbri.

Queste feature non sono semplici valori numerici, ma descrittori psicoacustici connessi direttamente a percezioni umane come loudness, brillantezza, ricchezza timbrica, rumorosità e spettro armonico.

- **Loudness (sonorità percepita)**: è la percezione dell'intensità di un suono. Non dipende soltanto dall'ampiezza fisica del segnale, ma anche dalla frequenza (curva di Fletcher-Munson) e dalla durata. Due suoni con la stessa energia fisica possono essere percepiti con loudness diversa.

- **Brillantezza**: rappresenta la sensazione di chiarezza o luminosità timbrica, legata alla presenza di componenti spettrali alte. È fortemente correlata alla spectral centroid.

- **Ricchezza timbrica**: indica la complessità e la varietà delle componenti spettrali. Suoni con molte armoniche ben distribuite (come archi o pianoforte) vengono percepiti come ricchi, mentre un suono sinusoidale puro appare timbricamente povero.

- **Rumorosità**: è la percezione di quanto un suono assomigli a un rumore piuttosto che a un tono definito. È quantificabile tramite la spectral flatness: più lo spettro è piatto, più il suono viene percepito come rumoroso.

- **Spettro armonico**: è l'insieme delle componenti sinusoidali che costituiscono un suono musicale. Nei suoni armonici le frequenze sono multipli interi della fondamentale, mentre nei suoni inarmonici (percussioni, rumori metallici) questa relazione non viene rispettata.

- **Ampiezza**: è la grandezza fisica del segnale (ad esempio spostamento d'aria o voltaggio), che determina l'intensità del suono. È direttamente collegata all'energia (RMS) e contribuisce alla percezione di loudness, anche se non in maniera lineare.

3.4 Definizione delle features

1) Root mean square (rms), che riassume la distribuzione di energia in ciascun frame

$$\text{rms} = \sqrt{\frac{\sum_{n=1}^{N} F_n^2}{N}}. \tag{3.1}$$

2) Spectral centroid, che misura la frequenza media pesata dalla somma delle ampiezze spettrali all'interno di un frame

$$\text{Centroid} = \frac{\sum_{k=1}^{K} P(f_k)\, f_k}{\sum_{k=1}^{K} P(f_k)}. \tag{3.2}$$

dove f_k è la frequenza nel k-esimo canale, $K = 40$ è il numero di canali, e $P(f_k)$ è l'ampiezza spettrale nel k-esimo canale.

3) Spectral bandwidth, che mostra l'intervallo di frequenze di un segnale pesato dal suo spettro

$$\text{Bandwidth} = \frac{\sum_{k=1}^{K} |\text{Centroid} - f_k|\ P(f_k)}{\sum_{k=1}^{K} P(f_k)}. \tag{3.3}$$

4) Spectral flatness, che misura la "piattezza" dello spettro, definita come il rapporto tra la media geometrica e la media aritmetica dello spettro in un frame

$$\text{Flatness} = \frac{\left(\prod_{k=1}^{K} P(f_k)\right)^{\frac{1}{K}}}{\frac{1}{K}\sum_{k=1}^{K} P(f_k)}. \tag{3.4}$$

5) Spectral roll-off high, che rappresenta la frequenza al di sotto della quale si trova una frazione α_h (tipicamente $0.85 \div 0.95$) dell'energia spettrale totale

$$\sum_{k=1}^{m_h} P(f_k) = \alpha_h \sum_{k=1}^{K} P(f_k), \tag{3.5}$$

dove f_{m_h} è la frequenza di high roll-off.

6) Spectral roll-off low, analogo al precedente, ma considerando una frazione α_l (tipicamente $0.05 \div 0.15$) dell'energia spettrale totale

$$\sum_{k=1}^{m_l} P(f_k) = \alpha_l \sum_{k=1}^{K} P(f_k), \tag{3.6}$$

dove f_{m_l} è la frequenza di roll-off basso.

Legenda dei simboli.

N numero di campioni nel frame.

F_n valore/ampiezza del segnale al n-esimo campione del frame.

K numero di canali/spettro (qui $K = 40$).

f_k frequenza del k-esimo canale.

$P(f_k)$ ampiezza spettrale nel k-esimo canale.

Centroid il centroide spettrale definito in (3).

$\alpha_h,\ \alpha_l$ frazioni di energia totale usate per i rolloff alto/basso (ad es. $\alpha_h \in [0.85, 0.95]$, $\alpha_l \in [0.05, 0.15]$).

$m_h,\ m_l$ indici dei canali tali che f_{m_h} e f_{m_l} soddisfano rispettivamente (6) e (7).

3.5 Integrazione futura con il modello di IA

Il dataset costruito durante questo lavoro rappresenta la base per la fase successiva: l'addestramento di un modello neurale che imiti il comportamento del plate reverb. Al momento, come riportato dal supervisore Fares Schulz, il gruppo non ha ancora avviato un trsaining vero e proprio, ma sta sperimentando diversea rchitetture per cpaire quale sia la più adatta. Per questa fase ci si appoggia ad un recente studio sullo spring reverb, che confronta alcuni dei modelli più promettenti.

Questi modelli si differenziano per struttura, capacità di apprendere le caratteristiche del segnale e costi computazionali.

La prima architettura è WaveNet, originariamente sviluppata per la sintesi vocale, ma ormai diventata un riferimento anche in ambito musicale. Questa rete usa convoluzioni dilatate stratificate, che permettono di osservare il segnale su scale temporaeli molto ampie. Questa caratteristica la rende particolarmente adatta a riprodurre le code lunghe e complesse del riverbero, catturandone le sfumature più sottili.

Il limite principale è il peso computazionale: l'accuratezza si paga con una forte richiesta di risorse, che può complicare l'uso in contesti real-time.

La seconda architettura è la Gated Recurrent Units (GRU). Si tratta di reti ricorrenti progettate per essere più leggere e veloci rispetto alle tradizionali LSTM (citate in seguito). Le GRU gestiscono in maniera semplificata i meccanismi di memoria, riuscendo comunque a tenere traccia delle informazioni passate.

La loro forza è la rapidità di addestramento e l'efficienza, ma a volte tendono a perdere precisione quando si tratta di sequenze molto lunghe, come ad esempio quelle di un riverbero particolarmente esteso.

La terza architettura è la Temporal Convolutional Network (TCN), e anche in questo caso si lavora con convoluzioni, ma la differenza con le precedenti è che sono organizzate in modo progressivo: gli strati iniziali osservano dettagli molto ravvicinati, mentrei successivi allargano la finestra temporale.

Tale approccio è bilanciato, spesso garantisce stabilità nell'addestramento e un contenuto consumo di risorse. Pur non raggiungendo i picchi di accuratezza di WaveNet, rappresenta una soluzione pratica in molti scenari.

La quarta architettura è la Long Short-Term Memory (LSTM), costituita da reti ricorrenti concepite per superare i limiti delle reti tradizionali, introducendo celle di memoria che permettono di mantenere informazioni anche a lungo termine. Questa caratteristica le rende molto potenti quando si tratta di modellare sequenze temporali articolate, come appunto la coda di riverbero.

Il problema è la loro complessità: hanno un numero elevato di parametri e risultano lente da addestrare, con un costo notevole di risorse computazionali.

La quinta e ultima architettura è la Gated Convolutional Network (GCN), che combina i vantaggi delle architetture convoluzionli con un meccanismo di "gating", cioè una sorta di filtro che seleziona dinamicamente le informazioni più rilevanti del segnale. A questo si aggiunge il condizionamento FiLM, che permette alla rete di adattarsi in base al contesto o ai parametri impostati.

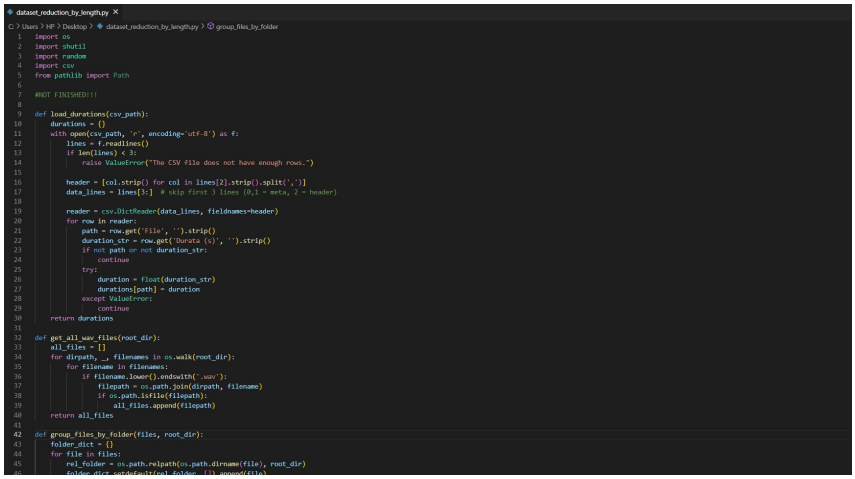
Nei test sullo spring reverb, il GCN ha fornito risultati particolarmente interessanti, soprattutto a frequenze di campionamento elevate come 48kHz, dove ha mostrato un buon compromesso tra qualità e velocità di elaborazione, rendendolo adatto anchea possibili applicazioni real-time.

I dati raccolti nello studio di riferimento suggeriscono quindi che, mentre WaveNet spicca per accuratezza a 16kHz, la GCN rappresenta la scelta più promettente a 48kHz.

Menzione speciale per il R.A.V.E. model (Realtime Audio Variational autoEncoder), cioè un sistema pensato specificamente per la sintesi audio neurale a bassa latenza, che potrebbe rivelarsi molto utile in applicazioni dove la rapidità di risposta è essenziale, ad esempio nelle performance dal vivo o nei contesti di produzione interattiva, e che rappresenta quindi un approccio diverso rispetto alle reti presentate in precedenza: tale modello non nasce per imitare un singolo effetto, ma per la sintesi neurale in tempo reale. Il R.A.V.E. model è così reattivo dal momento che il segnale audio viene compresso in una rappresentazione latente compatta e poi ricostruito in tempo reale con un decodificatore ottimizzato. Questo riduce drasticamente la quantità di dati da elaborare a ogni istante, mantenendo alta la qualità sonora e abbattendo la latenza.

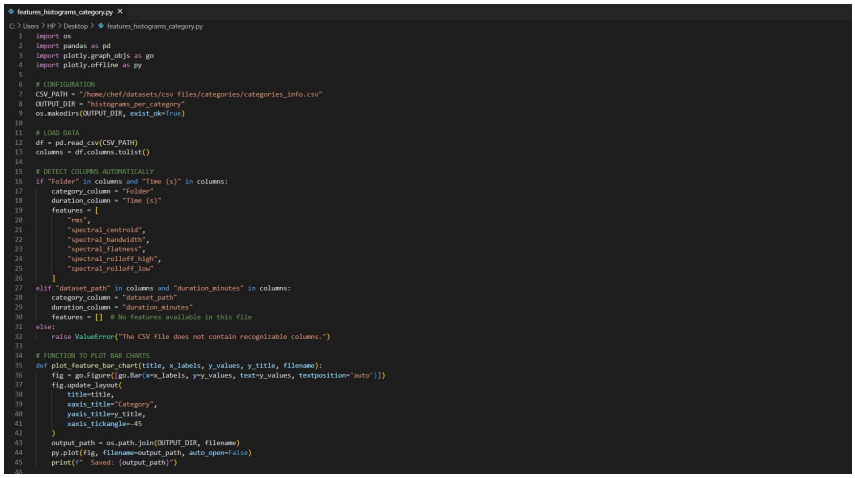
R.A.V.E. quindi entra in gioco come possibile alternativa o complemento alle reti precedenti, perchè non solo può emulare, ma può anche estendere l'effetto, garantendo reattività immediata nell'uso dal vivo.

In sintesi, l'inegrazione del dataset con queste architetture neurali rappresenta il passo naturale successivo. Un modello ben addestrato non si limita ad emulare il plate reverb originale, ma può offrire funzioni aggiuntive, controlli più flessibili e una fruizione più accessibile all'interno dei moderni ambienti di produzione musicale.



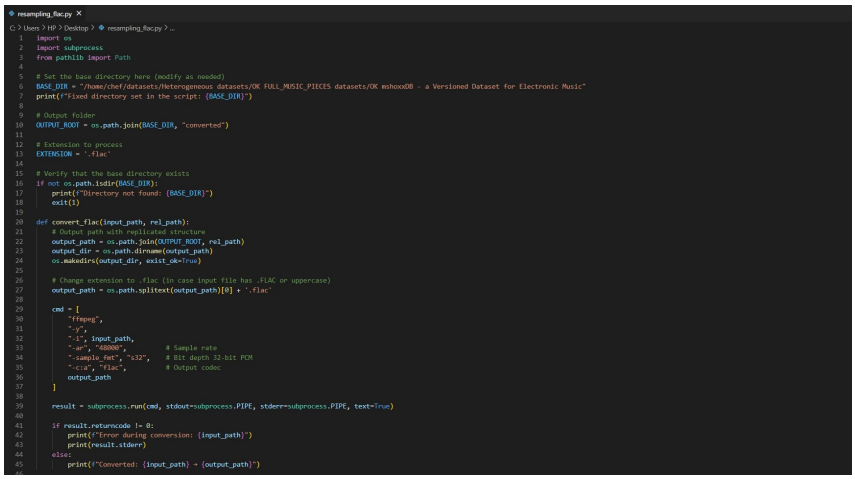
```python
import os
import shutil
import random
import csv
from pathlib import Path

#NOT FINISHED!!!

def load_durations(csv_path):
    durations = {}
    with open(csv_path, 'r', encoding='utf-8') as f:
        lines = f.readlines()
        if len(lines) < 3:
            raise ValueError("The CSV file does not have enough rows.")

        header = [col.strip() for col in lines[2].strip().split(',')]
        data_lines = lines[3:]  # skip first 3 lines (0,1 = meta, 2 = header)

        reader = csv.DictReader(data_lines, fieldnames=header)
        for row in reader:
            path = row.get('File', '').strip()
            duration_str = row.get('Durata (s)', '').strip()
            if not path or not duration_str:
                continue
            try:
                duration = float(duration_str)
                durations[path] = duration
            except ValueError:
                continue
    return durations

def get_all_wav_files(root_dir):
    all_files = []
    for dirpath, _, filenames in os.walk(root_dir):
        for filename in filenames:
            if filename.lower().endswith('.wav'):
                filepath = os.path.join(dirpath, filename)
                if os.path.isfile(filepath):
                    all_files.append(filepath)
    return all_files

def group_files_by_folder(files, root_dir):
    folder_dict = {}
    for file in files:
        rel_folder = os.path.relpath(os.path.dirname(file), root_dir)
```

Figura 3.4: dataset_reduction_by_length.py



```python
import os
import pandas as pd
import plotly.graph_objs as go
import plotly.offline as py

# CONFIGURATION
CSV_PATH = "/home/chef/datasets/csv files/categories/categories_info.csv"
OUTPUT_DIR = "histograms_per_category"
os.makedirs(OUTPUT_DIR, exist_ok=True)

# LOAD DATA
df = pd.read_csv(CSV_PATH)
columns = df.columns.tolist()

# DETECT COLUMNS AUTOMATICALLY
if "Folder" in columns and "Time (s)" in columns:
    category_column = "Folder"
    duration_column = "Time (s)"
    features = [
        "rms",
        "spectral_centroid",
        "spectral_bandwidth",
        "spectral_flatness",
        "spectral_rolloff_high",
        "spectral_rolloff_low"
    ]
elif "dataset_path" in columns and "duration_minutes" in columns:
    category_column = "dataset_path"
    duration_column = "duration_minutes"
    features = []  # No features available in this file
else:
    raise ValueError("The CSV file does not contain recognizable columns.")

# FUNCTION TO PLOT BAR CHARTS
def plot_feature_bar_chart(title, x_labels, y_values, y_title, filename):
    fig = go.Figure([go.Bar(x=x_labels, y=y_values, text=y_values, textposition='auto')])
    fig.update_layout(
        title=title,
        xaxis_title="Category",
        yaxis_title=y_title,
        xaxis_tickangle=-45
    )
    output_path = os.path.join(OUTPUT_DIR, filename)
    py.plot(fig, filename=output_path, auto_open=False)
    print(f"  Saved: {output_path}")
```

Figura 3.5: features_histograms_category.py



```python
import os
import subprocess
from pathlib import Path

# Set the base directory here (modify as needed)
BASE_DIR = "/home/chef/datasets/Heterogeneous datasets/OK FULL_MUSIC_PIECES datasets/OK mshoxx08 - a Versioned Dataset for Electronic Music"
print(f"Fixed directory set in the script: {BASE_DIR}")

# Output folder
OUTPUT_ROOT = os.path.join(BASE_DIR, "converted")

# Extension to process
EXTENSION = '.flac'

# Verify that the base directory exists
if not os.path.isdir(BASE_DIR):
    print(f"Directory not found: {BASE_DIR}")
    exit(1)

def convert_flac(input_path, rel_path):
    # Output path with replicated structure
    output_path = os.path.join(OUTPUT_ROOT, rel_path)
    output_dir = os.path.dirname(output_path)
    os.makedirs(output_dir, exist_ok=True)

    # Change extension to .flac (in case input file has .FLAC or uppercase)
    output_path = os.path.splitext(output_path)[0] + '.flac'

    cmd = [
        "ffmpeg",
        "-y",
        "-i", input_path,
        "-ar", "48000",          # Sample rate
        "-sample_fmt", "s32",    # Bit depth 32-bit PCM
        "-c:a", "flac",          # Output codec
        output_path
    ]

    result = subprocess.run(cmd, stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True)

    if result.returncode != 0:
        print(f"Error during conversion: {input_path}")
        print(result.stderr)
    else:
        print(f"Converted: {input_path} → {output_path}")
```

Figura 3.6: resampling_flac.py

Capitolo 4

Apprendimenti e riflessioni

4.1 Competenze tecniche acquisite

Il lavoro svolto ha permesso di sviluppare competenze che vanno oltre la semplice programmzione o analisi del segnale, configurandosi come un vero e proprio esercizio di gestione di un progetto complesso.

Un primo ambito riguarda la gestione dei dati su larga scala. La raccolta, l'organizzazione e l'elaborazione di quasi 100 GB di file audio hanno permesso di affrontare problematiche tipiche dei sistemi basati su big data; ottimizzazione delle risorse disponibili, uniformità degli standard e tracciabilità delle versioni. Questi aspetti sono risultati fondamentali per garantire coerenza interna al dataset e affidabilità delle procedure.

Un secondo ambito concerne la documentazione e tracciabilità delle procedure.

La produzione di file CSV contenenti metadati e l'aggiornamento di repositsory GitHub hanno reso il processo chiaro e replicabile. Particolare attenzione è stata posta alla creazione di una documentazione coerente, corretta e priva di ambiguità, in grado di facilitare la comprensione eil riutilizzo del lavoro da parte di altri membri del gruppo di ricerca. Questo approccio, vicino alle logiche di project management agile, si è rivelato cruciale per garantire continuità e interoperabilità.

4.2 Riflessioni sull'applicazione dell'IA al suono

L'emulazione del plate reverb mediante intelligenza artificiale si inserisce in un contesto più ampio, in cui l'IA rappresenta una leva trasformativa per l'industria musicale e creativa insieme.

La produzione musicale contemporanea è caratterizzata da una crescentee digitalizzazione, in cui gli strumenti software hanno sostituito dispositivi hardware costosi e complessi. In tale scenario, l'IA non costituisce soltanto un supporto, ma un elemento in grado di ridisegnare le modalità stesse di produzione: consente di ridurre i costi, una maggiore rapidità nei processi e possibilità di esperienze personalizzate per gli utenti.

L'emulazione del plate reverb costituisce un caso esemplare di questo approccio. La digitalizzazione di un dispositivo storicamente ingombrante e dispendioso permette di democratizzare l'accesso a tecnologie sonore di qualità, rendendo orizzontale il loro uso perchè disponibili ad un pubblico più ampio.

Parallelamente, l'IA non si limita a riprodurre fedelmente il comportamento fisico di un dispositivo, ma offre margini di estensione creativa che aprono la strada a soluzioni sonore inedite.

Un riferimento utile in questo senso è il testo Guerra Sonora di Steve Goodman, che analizza come il suono e le tecnologie di produzione sonora abbiano una dimensione politica, percettiva ed emotiva. Traslato nel contesto dell'IA, significa riconoscere che emulare un plate reverb non sia solo un esercizio ingegneristico, ma un'operazione che agisce su un immaginario sonoro consolidato da decenni di produzione musicale.

In altre parole, la tecnologia diventa un mezzo per preservare e al tempo stesso ridefinire la memoria sonora collettiva.

Un aspetto centrale della riflessione di Goodman è la nozione di affetto, in termini di quanto un suono produca effetti diretti sui corpi, ad esempio avvolgendoli o modificando stati d'animo. L'affettività sonora del plate reverb quindi si manifesta nella capacità del suo timbro di generare un campo emotivo, in questo caso un alone sonoro che amplifica la presenza di una voce o di uno strumento, diverso da uno spazio fisico realistico.

L'emulazione tramite IA, quindi, non è soltanto la riproduzione di un effetto tecnico, ma anche la ricostruzione di un'esperienza affettiva. Riprodurre digitalmente il plate significa garantire l'accesso ad una modalità specifica di modulare le emozioni, di dare forma a un paessaggio sonoro che incida direttamente sulla percezione e sul vissuto dell'ascoltatore.

La tecnologia, in questo caso, diventa uno strumento per continuare a trasmettere e trasformare intensità affettive consolidate da decenni di pratica musicale.

La prospettiva di Goodman permette quindi di comprendere come le tecnologie sonore non siano mai neutrali: esse modellano non solo l'estetica musicale, ma anche la dimensione affet-

tiva dell'ascolto. L'intelligenza artificiale, applicata al plate reverb, contribuisce a mantenere viva questa eredità, e soprattutto apre possibilità inedite di manipolare e reinventare l'affettività sonora.

4.3 Crescita personale e professionale

Le attività di creazione del dataset, elaborazione delle features e stesura della documentazione hanno rappresentato un contesto formativo non soltanto tecnico, ma anche metodologico e professionale.

Un aspetto rilevante è stato lo sviluppo della capacità di gestire responsabilità individuali all'interno di un progetto collettivo. Le scelte compiute in fase di selezione dei file audio e strutturazione del dataset hanno avuto ripercussioni dirette sugli sviluppi successivi del modello di intelligenza artificiale, richiedendo quindi un'attenzione particolare alla qualità e alla coerenza metodologica.

Altro elemento cruciale è stato l'inserimento in un contesto internazionale e interdisciplinare, in cui figure provenienti da ambiti diversi (ingegneria informatica, musicologia, fisica, acustica, psicoacustsica, informatica e sound design) hanno condiviso prospettive complementari. Questa dimensione collaborativa ha messo in luce come l'innovazione più significativa derivi spesso dall'incontro tra competenze eterogenee.

Infine, l'integrazione di competenze organizzative, intedisciplinarità e sensibilità culturale si configura come una delle principali eredità di questo percorso, con ricadute dirette sia sul piano della ricerca che su quello professionale.

Conclusioni

L'intento del lavoro svolto non è stato soltanto quello di raccogliere materiale audio, ma di costruire un insieme coerente, ben documentato e riutilizzabile, in grado di costruire una base concreta per le fasi successive di ricerca e sviluppo.

Il dataset ottenuto comprende una varietà di file audio, tra cui voci, strumenti musicali, field recordings, mix di brani e brani completi. Questa scelta è stata dettata dalla necessità di garantire un'ampia copertura di timbri, dinamiche e contenuti, così da consentire ad un modello neurale di generalizzare il comportamento del plate reverb in scenari realistici. La struttura e l'organizzazione dei dati sono state pensate per favorire non solo l'uso immediato, ma anche l'eventuale estensione futura con nuove categorie e materiali.

Per questo è stata inserita una fase di features extraction realizzata tramite scripts Python con l'utilizzo della libreria Librosa, che forniscono una serie di strumenti operativi per il bilanciamento, la riduzione, il resampling e appunto l'analisi dei dati, ed è stata redatta una documentazione precisa che descrive i contenuti sonori e i tools per gestirli.

Questa infrastruttura consente di mantenere flessibile il materiale, di adattarlo ad esigenze differenti e di semplificarne l'integrazione in pipeline di training neurale.

Le prospettive future riguardano principalmente tre direzioni: l'addestramento vero e proprio di modelli neurali, con la possibilità di confrontare architetture diverse (WaveNet, GRU, TCN, LSTM, GCN e R.A.V.E.), l'ampliamento del dataset attraverso nuove tipologie di sorgenti sonore o tecniche di data augmentation e l'estensione della stessa metodologia ad altri effetti analogici come chamber reverb.

Per concludere, è stato realizzato un dataset eterogeneo con relativi descrittori acustici e strumenti di gestione, che rappresenta una base solida per lo sviluppo di un modello o più di intelligenza artificiale in grado di emulare fedelmente e accuratamente il plate reverb e utilizzabile in contesti reali.

Bibliografia

[1] M. D. Pascalis e F. Vogliazzo. «Plate Reverb.» Epacoustics Studio. (n.d.), indirizzo: https://epacoustics.studio/plate-reverb/.

[2] S. Bilbao. «Plate Reverb.» The University of Edinburgh. (n.d.), indirizzo: https://www2.ph.ed.ac.uk/~sbilbao/platerevpage.htm.

[3] S. Goodman, *Sonic Warfare: Sound, Affect, and the Ecology of Fear*. MIT Press, 2010.

[4] F. Papaleo, X. Lizarraga-Seijas e F. Font, «Evaluating Neural Networks Architectures for Spring Reverb Modelling,» *arXiv preprint*, 2024. arXiv: 2409.04953 [cs.SD]. indirizzo: https://arxiv.org/abs/2409.04953.

[5] G. Peeters, «A large set of audio features for sound description (similarity and classification) in the CUIDADO project,» IRCAM, rapp. tecn., 2003. indirizzo: http://recherche.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf.

[6] X. Serra, «Musical Sound Modeling with Sinusoids plus Noise,» Universitat Pompeu Fabra, rapp. tecn., 1997. indirizzo: https://mtg.upf.edu/system/files/publications/MSM-1997-Xserra_0.pdf.

[7] G. Tzanetakis e P. Cook, «Musical genre classification of audio signals,» *IEEE Transactions on Speech and Audio Processing*, vol. 10, n. 5, pp. 293–302, 2002. doi: 10.1109/TSA.2002.800560.

[8] J. D. Deng, S. Cranefield e C. Simmermacher. «A Study on Feature Analysis for Musical Instrument Classification.» (n.d.), indirizzo: https://www.researchgate.net/publication/5503410_A_Study_on_Feature_Analysis_for_Musical_Instrument_Classification.

[9] Sonology.org. «Audio Communication & Sonology – Joint Master Degree.» (n.d.), indirizzo: https://sonology.org/audio-communication-sonology/.

[10] T. U. Berlin. «Electronic Studio at the Technical University (TU) in Berlin – Introduction.» (n.d.), indirizzo: https://fhein.users.ak.tu-berlin.de/Alias/Geschichte/Intro_eng.html.

[11] T. U. Berlin. «Historical Devices Owned by the Audio Communication Group.» (n.d.), indirizzo: https://soundandscience.net/collections/historical-devices-owned-by-the-audio-communication-group-technische-universitat-berlin/.

[12] J. Palmer, «Conversation with Folkmar Hein,» *eContact!*, vol. 10, n. 2, 2007. indirizzo: https://econtact.ca/10_2/HeinFo_Palmer.html.