MDPI

Project Report

Classificazione di Segnali Audio di Chitarra tramite Rappresentazioni Tempo-Frequenza e Reti Neurali Convoluzionali

Oleksandr Poddubnyy [1] e Davide Fadale [2]

[1] Addresses 1: ole.poddubnyy@stud.uniroma3.it
[2] Addresses 2: dav.fadale@stud.uniroma3.it

Abstract

Questo lavoro affronta il problema della classificazione automatica di segnali audio di chitarra appartenenti a 13 diverse categorie, utilizzando rappresentazioni tempo-frequenza e reti neurali convoluzionali. Il dataset, contenente campioni audio etichettati, è stato preprocessato per estrarre Mel-spectrogrammi, che rappresentano graficamente la distribuzione spettrale nel tempo, adattandoli a un formato compatibile con modelli deep learning. Successivamente, è stata progettata e addestrata una rete neurale convoluzionale (CNN) in ambiente TensorFlow, valutata tramite suddivisione train/test. I risultati hanno evidenziato un buon livello di accuratezza nella classificazione, con un'analisi approfondita delle confusion matrix per individuare le classi maggiormente confuse. Sono state inoltre esplorate estensioni come l'utilizzo di MFCC per confronto, il salvataggio del modello, e l'impiego del transfer learning tramite VGG16. L'intero flusso è stato supportato da visualizzazioni interattive e da un'analisi critica degli errori. I risultati dimostrano la validità dell'approccio e la replicabilità della pipeline proposta per la classificazione audio.

Keywords: classificazione audio; reti neurali convoluzionali; Mel-spectrogram; MFCC; deep learning; suoni di chitarra; transfer learning; elaborazione del segnale

1. Introduzione

Negli ultimi anni, la classificazione automatica dei segnali audio è diventata un ambito di ricerca molto attivo, grazie soprattutto agli avanzamenti nelle tecniche di rappresentazione dei segnali e all'applicazione di metodi di deep learning sempre più sofisticati [1,2]. In particolare, le reti neurali convoluzionali (CNN), originariamente sviluppate per l'analisi di immagini, hanno dimostrato una notevole efficacia anche nell'ambito dell'elaborazione di segnali audio trasformati in rappresentazioni visive come Mel-spectrogrammi [3].

Il riconoscimento e la classificazione automatica dei suoni degli strumenti musicali è una sfida interessante non solo dal punto di vista tecnologico, ma anche per le sue applicazioni pratiche nella produzione musicale, nella didattica musicale assistita dal computer e nella gestione di archivi digitali [4]. Tuttavia, questo tipo di classificazione può presentare numerose difficoltà, legate sia alla grande varietà timbrica e dinamica dei suoni, sia alla complessità intrinseca del segnale audio, caratterizzato da variazioni temporali e spettrali significative.

In questo contesto, il presente lavoro affronta il problema della classificazione automatica di segnali audio specificamente relativi alla chitarra, appartenenti a 13 diverse categorie, tra cui tecniche esecutive differenti come slap, slide e flageolet. Per raggiungere tale obiettivo, abbiamo esplorato e confrontato diverse rappresentazioni tempo-frequenza

Received:
Revised:
Accepted:
Published:

Citation: Poddubnyy O.; Fadale D. Classificazione di Suoni di Chitarra con CNN. *Computers* **2025**, *1*, 0.

come il Mel-spectrogram e le Mel-Frequency Cepstral Coefficients (MFCC), valutandone criticamente vantaggi e limiti. Inoltre, abbiamo impiegato tecniche avanzate come il transfer learning, sfruttando modelli pre-addestrati in ambito visivo adattati alla classificazione audio [1,5].

Questo lavoro presenta, quindi, una pipeline completa per la classificazione automatica di suoni di chitarra, integrando preprocessamento dei segnali, costruzione e addestramento di reti neurali convoluzionali, analisi delle prestazioni, visualizzazioni interattive e analisi approfondita degli errori.

2. Obiettivi del Progetto

L'obiettivo principale di questo lavoro è la classificazione automatica di segnali audio di chitarra, appartenenti a 13 differenti classi rappresentative di specifiche tecniche esecutive, come *slap*, *slide* e *flageolet*. Questo compito rientra nel più ampio contesto della classificazione audio, ambito particolarmente rilevante grazie alla crescente diffusione di strumenti digitali per l'analisi e la catalogazione automatica di contenuti multimediali [2,4].

Per raggiungere questo obiettivo generale, sono stati definiti una serie di sotto-obiettivi specifici:

- Effettuare il caricamento e il preprocessing dei file audio originali per renderli idonei alle successive fasi di elaborazione.
- Estrarre rappresentazioni tempo-frequenza efficaci per la classificazione audio. In particolare, si è focalizzata l'attenzione su tecniche quali il Mel-spectrogram e le Mel Frequency Cepstral Coefficients (MFCC), che permettono una visualizzazione efficace delle caratteristiche spettrali e temporali dei suoni [1,3].
- Realizzare, addestrare e valutare una rete neurale convoluzionale (CNN) in grado di apprendere pattern distintivi da queste rappresentazioni, allo scopo di classificare con accuratezza i vari tipi di suoni di chitarra.
- Valutare in modo critico e comparativo le prestazioni delle diverse rappresentazioni adottate, discutendo vantaggi e limiti di ciascuna.
- Esplorare tecniche avanzate come il transfer learning mediante l'impiego del modello pre-addestrato VGG16, adattati al dominio audio, per migliorare ulteriormente le prestazioni del sistema classificatorio anche in presenza di dataset limitati [5].
- Integrare visualizzazioni interattive, analisi approfondite degli errori e strumenti per il salvataggio e il riutilizzo dei modelli addestrati, con l'intento di facilitare l'interpretabilità dei risultati e migliorare l'usabilità del progetto.

Nel complesso, questo lavoro intende fornire una pipeline completa e replicabile per la classificazione audio automatica, che integri metodologie consolidate nella letteratura con nuove estensioni sperimentali volte a massimizzare accuratezza e interpretabilità dei risultati.

3. Materiali e Metodi

In questa sezione verranno descritti nel dettaglio i materiali utilizzati e le metodologie implementate nel corso del progetto. In particolare, sarà dapprima introdotto il dataset impiegato, costituito da registrazioni audio di diverse tecniche di esecuzione sulla chitarra, con le relative operazioni di preprocessing necessarie per renderlo compatibile con le fasi successive dell'analisi. Successivamente, saranno illustrate le tecniche adottate per la rappresentazione tempo-frequenza dei segnali audio, focalizzando l'attenzione sulle motivazioni della scelta del Mel-spectrogram e delle MFCC come strumenti principali per estrarre caratteristiche rilevanti ai fini della classificazione. Infine, verrà presentata la struttura della rete neurale convoluzionale (CNN) utilizzata per il processo di apprendimento

automatico, insieme ai dettagli operativi delle diverse fasi di addestramento e valutazione del modello.

3.1. Dataset e Preprocessing

Il dataset utilizzato nel presente lavoro è costituito da registrazioni audio di chitarra appartenenti a 13 diverse classi, ciascuna corrispondente ad una particolare tecnica esecutiva dello strumento (es. *slap*, *slide*, *flageolet*). I dati audio originali sono stati organizzati secondo una struttura gerarchica in cui ogni classe è rappresentata da una specifica cartella, contenente i relativi campioni audio. Questa organizzazione ha permesso una facile identificazione e gestione delle etichette associate ad ogni file audio.

Il preprocessing dei dati è stato svolto tramite la libreria `librosa` in Python [6], una delle più diffuse nell'ambito dell'elaborazione audio. Ogni file audio è stato innanzitutto caricato e standardizzato in termini di durata e frequenza di campionamento, per garantire uniformità tra i campioni e facilitare le operazioni successive. La frequenza di campionamento adottata è stata di 22050 Hz, comunemente usata per analisi audio musicale [3].

Dopo la fase iniziale di caricamento, i file audio sono stati convertiti in rappresentazioni tempo-frequenza, principalmente Mel-spectrogram, per consentirne l'analisi tramite reti neurali convoluzionali (CNN). Ogni spettrogramma risultante è stato adattato ad una dimensione standard (128×128 pixel), utilizzando tecniche di padding o cropping quando necessario, al fine di uniformare i dati e agevolare l'addestramento del modello neurale.

Terminata l'estrazione delle rappresentazioni, il dataset finale è stato composto da una matrice numerica `X`, contenente gli spettrogrammi, e da un vettore di etichette categoriche `y`. Queste ultime sono state convertite da stringhe testuali a codici numerici attraverso il modulo `LabelEncoder` della libreria `scikit-learn` [7], e infine codificate in formato *one-hot encoding* per essere compatibili con l'output softmax della rete neurale.

3.2. Rappresentazioni Tempo-Frequenza

Le rappresentazioni tempo-frequenza sono ampiamente utilizzate nell'elaborazione dei segnali audio, in quanto consentono di catturare simultaneamente informazioni temporali e spettrali del segnale analizzato [2]. In questo progetto, sono state impiegate principalmente due rappresentazioni: il *Mel-spectrogram* e le *Mel Frequency Cepstral Coefficients* (MFCC).

Il *Mel-spectrogram* è una rappresentazione bidimensionale del segnale audio, ottenuta applicando la Trasformata di Fourier a breve termine (Short-Time Fourier Transform, STFT) e successivamente proiettando le frequenze risultanti su una scala mel, che simula il modo in cui l'orecchio umano percepisce le frequenze sonore [8]. Il Mel-spectrogram risulta particolarmente efficace per applicazioni di classificazione audio basate su reti neurali convoluzionali, in quanto conserva le informazioni spettrali dettagliate del segnale originale, permettendo così al modello di apprendere pattern complessi e distintivi.

Le *Mel Frequency Cepstral Coefficients* (MFCC), invece, sono coefficienti ottenuti applicando una trasformata discreta del coseno (Discrete Cosine Transform, DCT) al logaritmo delle energie delle bande mel del segnale audio. Le MFCC rappresentano una versione compressa e altamente discriminativa delle caratteristiche timbriche del segnale audio, rendendole adatte soprattutto per modelli semplici o quando si richiede un set limitato di caratteristiche [9]. Tuttavia, la compressione della rappresentazione comporta una perdita di dettagli nelle informazioni temporali e spettrali fini, che può limitare la performance nel caso di reti profonde come le CNN.

In questo studio, entrambe le rappresentazioni sono state utilizzate e confrontate, permettendo così di valutare criticamente i vantaggi e gli svantaggi relativi di ciascuna tecnica nel contesto della classificazione automatica dei segnali audio di chitarra.

3.3. Architettura della Rete Neurale

Per risolvere il problema della classificazione automatica dei segnali audio, è stata progettata e implementata una rete neurale convoluzionale (CNN), particolarmente adatta a gestire dati strutturati sotto forma di immagini o rappresentazioni bidimensionali, come nel caso dei Mel-spectrogrammi [1,3].

La struttura della CNN adottata è composta da diversi strati funzionali sequenziali. Il primo blocco convoluzionale è costituito da uno strato `Conv2D` con 32 filtri di dimensione 3×3, che permette di rilevare le caratteristiche locali fondamentali dei segnali audio (come attacchi, transizioni e armonici). Tale strato è seguito da uno strato di attivazione ReLU (Rectified Linear Unit), che consente una più rapida convergenza durante l'addestramento, e da uno strato di pooling di tipo `MaxPooling2D` con dimensione 2×2, che ha la funzione di ridurre la dimensionalità spaziale delle caratteristiche estratte, mantenendo solo le informazioni più rilevanti [10].

Il secondo blocco convoluzionale ripete una struttura analoga, aumentando la complessità del modello con 64 filtri di dimensione 3×3, seguiti nuovamente da un'attivazione ReLU e da un ulteriore strato di MaxPooling. Questa sequenza permette alla rete di apprendere caratteristiche audio più complesse e astratte.

In seguito ai blocchi convoluzionali, è stato introdotto uno strato di appiattimento (`Flatten`) che converte le caratteristiche multidimensionali in un vettore monodimensionale, necessario per la classificazione finale. Tale vettore viene passato ad uno o più strati densamente connessi (`Dense`), accompagnati da uno strato di dropout, una tecnica di regolarizzazione che previene l'overfitting eliminando casualmente alcune connessioni durante l'addestramento [11].

Infine, la classificazione vera e propria viene realizzata mediante uno strato finale `Dense` con un numero di neuroni pari alle classi da identificare (13), e una funzione di attivazione *softmax*, che restituisce probabilità normalizzate per ciascuna classe del dataset.

3.4. Fasi di Addestramento

L'addestramento della rete neurale convoluzionale è stato realizzato seguendo una procedura standardizzata che comprende diverse fasi principali: suddivisione del dataset, compilazione del modello, addestramento vero e proprio, e valutazione delle prestazioni.

In primo luogo, il dataset è stato suddiviso in due sottoinsiemi distinti (*train* e *test*) utilizzando una strategia di campionamento stratificato (`train_test_split` con opzione `stratify`), per garantire che la distribuzione delle classi rimanesse proporzionata in entrambi i set e per fornire una valutazione realistica della capacità di generalizzazione del modello [7]. È stata scelta una proporzione comune, ovvero l'80% per l'addestramento e il 20% per il test.

Successivamente, il modello CNN è stato compilato impostando la funzione di perdita *categorical cross-entropy*, particolarmente indicata per problemi di classificazione multiclasse con output softmax, e l'algoritmo di ottimizzazione *Adam*, che combina efficacemente tecniche adattative di gradiente con velocità e stabilità di convergenza [12]. L'accuratezza (*accuracy*) è stata adottata come metrica primaria di monitoraggio dell'addestramento.

Durante la fase di addestramento, il modello è stato allenato per un numero fisso di epoche, monitorando contemporaneamente la funzione di perdita e l'accuratezza sia sul set di addestramento che su un sottoinsieme dedicato di validazione, ricavato dal training set stesso tramite validazione incrociata (*validation split*). La storia delle metriche (*accuracy*

e *loss*) registrate durante l'addestramento è stata utilizzata per analizzare l'andamento del processo di apprendimento, identificando tempestivamente situazioni di possibile overfitting.

Terminato l'addestramento, il modello è stato valutato sul test set, composto da dati mai presentati al modello durante la fase di apprendimento. I risultati finali, misurati in termini di accuratezza complessiva e confusion matrix, hanno permesso di quantificare la capacità del modello di generalizzare correttamente le caratteristiche apprese dal dataset di addestramento.

4. Estensioni Sperimentali

Al fine di ampliare ulteriormente l'analisi e valutare criticamente la robustezza e l'efficacia dell'approccio adottato, il progetto è stato arricchito da diverse estensioni sperimentali. In particolare, sono state implementate strategie avanzate di gestione e analisi del modello, tra cui il salvataggio e riutilizzo del modello addestrato per facilitare la replicabilità degli esperimenti e la comparazione delle prestazioni.

È stato inoltre condotto un confronto dettagliato tra due importanti rappresentazioni tempo-frequenza (Mel-Spectrogram e MFCC), al fine di esaminare in modo approfondito i vantaggi e i limiti di ciascuna tecnica nel contesto specifico della classificazione audio tramite CNN. In aggiunta, si è esplorato l'utilizzo del transfer learning mediante modello pre-addestrato VGG16, normalmente utilizzato per classificazione di immagini, adattandolo per la classificazione di rappresentazioni audio-visive, al fine di verificare la possibilità di migliorare ulteriormente le performance del sistema con risorse computazionali e dati limitati.

Per rendere il progetto più interattivo e fruibile, sono state integrate visualizzazioni grafiche e strumenti per l'ascolto diretto dei dati audio, offrendo così una modalità aggiuntiva di interpretazione e validazione qualitativa dei risultati. Tra le visualizzazioni utilizzate, la *waveform* ha avuto un ruolo centrale: essa mostra l'andamento dell'ampiezza del segnale audio nel tempo, permettendo di osservare visivamente la struttura temporale dei suoni, come attacchi, pause e variazioni dinamiche, che possono essere correlati con la percezione uditiva e con i pattern rilevati dalla rete neurale. Infine, è stata svolta un'accurata analisi degli errori commessi dal modello, con l'obiettivo di individuare possibili criticità, approfondire la comprensione del comportamento della rete neurale e suggerire possibili direzioni per miglioramenti futuri.

4.1. Salvataggio e Riutilizzo del Modello

Una fase fondamentale nello sviluppo di sistemi di classificazione basati su reti neurali è rappresentata dal salvataggio del modello addestrato, che consente di preservare i risultati dell'apprendimento e di riutilizzare la rete neurale in successive analisi senza dover ripetere il processo di addestramento, spesso computazionalmente costoso [13].

Nel presente progetto, il modello addestrato è stato salvato utilizzando la funzione `model.save()` fornita dalla libreria TensorFlow/Keras, generando un file in formato `.h5` contenente tutte le informazioni essenziali: la struttura completa della rete neurale, i pesi appresi durante il training, nonché le configurazioni ottimali dell'addestramento (funzione di perdita, ottimizzatore e metriche di valutazione).

Il salvataggio strutturato del modello permette non solo di riutilizzarlo facilmente per effettuare inferenza su nuovi dati audio, ma anche di riprendere e approfondire sperimentazioni precedenti, effettuare confronti diretti tra differenti versioni del modello e condividere modelli pre-addestrati con altri ricercatori o applicazioni pratiche.

Per il riutilizzo del modello, è sufficiente caricare il file salvato tramite la funzione `load_model()` di Keras. Questa operazione è particolarmente vantaggiosa quando si lavora

in ambienti differenti, ad esempio se si desidera utilizzare il modello precedentemente addestrato su un'altra piattaforma o in un contesto di produzione. Tale procedura, dunque, rappresenta un elemento chiave nella pipeline di machine learning moderna, facilitando la replicabilità degli esperimenti e il deployment efficace delle soluzioni sviluppate.

4.2. Confronto tra Mel-Spectrogram e MFCC

Nel corso del progetto, è stato condotto un confronto sperimentale tra due tra le più diffuse rappresentazioni tempo-frequenza del segnale audio: il Mel-spectrogram e i Mel Frequency Cepstral Coefficients (MFCC). L'obiettivo è stato valutare l'impatto di ciascuna rappresentazione sulle prestazioni della rete neurale convoluzionale (CNN), analizzandone punti di forza e limiti in relazione alla classificazione dei suoni di chitarra.

Il *Mel-spectrogram* fornisce una mappa bidimensionale che rappresenta l'intensità delle frequenze nel tempo, proiettata su scala mel. Questa rappresentazione mantiene una ricca quantità di informazioni sia spettrali che temporali, risultando particolarmente adatta all'impiego con CNN, le quali sono in grado di apprendere pattern spaziali complessi da tali rappresentazioni visive [1,3]. Inoltre, i Mel-spectrogrammi possono essere interpretati come vere e proprie immagini, facilitando l'applicazione di tecniche di visione artificiale al dominio audio.

Al contrario, le *MFCC* rappresentano una forma più compatta di caratterizzazione del segnale: estraggono i coefficienti cepstrali applicando una trasformata discreta del coseno (DCT) sulle bande della scala mel. Questa tecnica è particolarmente efficace nel catturare le caratteristiche timbriche principali del segnale audio, ma comporta una compressione dell'informazione, con conseguente perdita di dettagli temporali e spettrali fini [9]. Tale perdita può limitare le capacità della rete di individuare pattern complessi necessari per distinguere alcune classi più sottilmente differenziate.

Dai test condotti, è emerso che i modelli addestrati su Mel-spectrogrammi hanno generalmente ottenuto risultati migliori in termini di accuratezza rispetto a quelli basati su MFCC. Tuttavia, le MFCC hanno mostrato buone prestazioni su classi più semplici, con tempi di elaborazione e dimensione dei dati ridotti, rendendoli adatti a sistemi con risorse computazionali limitate.

Il confronto ha quindi evidenziato che la scelta della rappresentazione deve essere guidata dal bilanciamento tra accuratezza desiderata, complessità del problema e vincoli computazionali.

4.3. Transfer Learning con VGG16

Una delle estensioni sperimentali più rilevanti del progetto ha riguardato l'impiego del *transfer learning*, una tecnica che consente di riutilizzare modelli neurali pre-addestrati su grandi dataset per nuovi compiti specifici, riducendo tempi di addestramento e migliorando la generalizzazione, soprattutto in presenza di dataset limitati [14].

Nel presente lavoro, è stato sperimentato l'utilizzo del modello VGG16, originariamente sviluppato per la classificazione di immagini nel dataset ImageNet [15]. L'idea chiave è che, trattando i Mel-spectrogrammi come immagini bidimensionali, sia possibile sfruttare le capacità di estrazione di caratteristiche visive apprese dal modello per analizzare anche segnali audio rappresentati visivamente.

La procedura ha previsto il caricamento del modello VGG16 con i pesi pre-addestrati, escludendo la parte finale del classificatore originario. I livelli convoluzionali pre-addestrati sono stati mantenuti congelati per preservare le feature generali apprese, mentre sono stati aggiunti nuovi strati densamente connessi (fully connected) in coda al modello per adattarlo alla classificazione delle 13 classi di suoni di chitarra.

L'addestramento ha interessato solo i livelli finali, mentre la base convoluzionale è stata usata come estrattore di caratteristiche. Questo approccio ha permesso di ottenere risultati competitivi, con un tempo di addestramento significativamente ridotto rispetto ai modelli CNN addestrati da zero, evidenziando la validità del transfer learning anche nel dominio audio, quando opportunamente rappresentato.

L'esperimento dimostra che tecniche originariamente concepite per la visione artificiale possono essere efficacemente adattate a contesti di classificazione audio tramite rappresentazioni tempo-frequenza, offrendo nuove prospettive per la riusabilità di architetture complesse e pre-addestrate in scenari di apprendimento supervisionato.

4.4. Visualizzazione e Ascolto dei Dati

Per migliorare l'interpretabilità e la verifica qualitativa della pipeline sviluppata, sono state aggiunte funzionalità di visualizzazione interattiva e ascolto diretto dei campioni audio.

In particolare:

- I Mel-spectrogram sono stati visualizzati tramite librerie Python come `librosa.display` e `matplotlib`, consentendo una rappresentazione in tempo reale delle caratteristiche spettrali e temporali del segnale [6,16].
- Nei notebook Jupyter è stato integrato il modulo `IPython.display.Audio`, che permette di ascoltare ogni singolo campione audio, sincronizzato con la visualizzazione del relativo spettrogramma. Questo approccio favorisce un'inedita modalità di analisi esplorativa, in cui l'immagine e il suono si supportano reciprocamente.

L'integrazione di queste funzionalità ha reso possibile verifiche rapide di coerenza tra le caratteristiche visive dei campioni e il loro contenuto sonoro, facilitando l'individuazione di eventuali artefatti o anomalie. Questo approccio multimodale migliora sia l'esperienza dell'utente sia la qualità della validazione finale.

4.5. Analisi degli Errori

L'analisi degli errori rappresenta una fase essenziale nel processo di valutazione di un modello di classificazione, in quanto permette di comprendere non solo dove il modello fallisce, ma anche perché. Tale analisi fornisce indicazioni preziose per miglioramenti futuri, sia in termini di struttura del modello che di qualità dei dati o tecniche di rappresentazione [17].

Nel contesto del presente progetto, l'analisi è stata condotta esaminando:

- Gli esempi di test classificati erroneamente dal modello, visualizzando i rispettivi Mel-spectrogrammi e confrontandoli con quelli correttamente classificati.
- La *confusion matrix*, utile per individuare le coppie di classi maggiormente confuse tra loro, come ad esempio *slide* e *slap*, che condividono alcune componenti spettrali simili.
- Le probabilità di classificazione assegnate dal modello, al fine di individuare eventuali casi di alta incertezza predittiva (bassa confidenza anche nella classe assegnata).

Per ogni esempio errato è stata effettuata una revisione qualitativa, osservando sia la struttura visiva dello spettrogramma sia ascoltando l'audio originale. In molti casi, si è riscontrata una reale ambiguità acustica tra classi simili, evidenziando i limiti della rappresentazione corrente o la necessità di aumentare la diversità del dataset.

L'analisi ha inoltre mostrato che alcune classi con campioni meno bilanciati risultavano più difficili da distinguere, suggerendo l'opportunità di adottare tecniche di *data augmentation* o riequilibrio dei dati in future versioni del modello.

Nel complesso, questa fase ha confermato l'importanza di affiancare alle metriche quantitative un'analisi qualitativa e interpretativa dei risultati, fondamentale per garantire la solidità e l'affidabilità del sistema sviluppato.

5. Risultati

In questa sezione vengono presentati i risultati sperimentali ottenuti a seguito dell'addestramento e della valutazione del modello di classificazione. I risultati sono stati analizzati sia in termini quantitativi, attraverso metriche standard di valutazione delle performance, sia qualitativi, mediante visualizzazioni interpretative e strumenti diagnostici.

La valutazione è stata condotta sul set di test, costituito da dati non visti durante l'addestramento. Le metriche principali considerate includono l'accuratezza complessiva, la funzione di perdita (*categorical cross-entropy*) e la matrice di confusione, che consente di analizzare il comportamento del modello sulle singole classi. A complemento dell'analisi, sono stati prodotti grafici di apprendimento che mostrano l'andamento della perdita e dell'accuratezza durante le epoche di training, sia sul set di addestramento che su quello di validazione.

Le sottosezioni seguenti dettagliano ciascuno di questi aspetti, con particolare attenzione alla valutazione dei modelli su diverse rappresentazioni tempo-frequenza e all'impatto delle strategie di ottimizzazione e regolarizzazione adottate.

5.1. Accuratezza e Metriche di Valutazione

La valutazione delle prestazioni del modello di classificazione è stata effettuata principalmente tramite l'accuratezza (*accuracy*), ovvero la proporzione di predizioni corrette rispetto al numero totale di esempi nel set di test. Oltre all'accuratezza complessiva, sono state calcolate anche metriche più dettagliate come precisione, richiamo e F1-score per ciascuna classe, al fine di ottenere una visione più articolata del comportamento del modello [18].

L'accuratezza finale ottenuta dal modello basato su Mel-spectrogram si è attestata su valori superiori al 90% sul set di test, con leggere variazioni a seconda dell'inizializzazione dei pesi e del numero di epoche. Il modello addestrato su MFCC, pur restituendo una performance inferiore, ha mantenuto un livello accettabile (circa 80–85%), confermando la validità di entrambe le rappresentazioni, seppur con un diverso grado di efficacia.

Queste metriche sono state calcolate con la funzione `classification_report()` della libreria `scikit-learn`, che fornisce anche il supporto (numero di esempi per classe) per interpretare correttamente le performance per ciascuna categoria.

I risultati ottenuti indicano che il modello CNN è in grado di apprendere in modo efficace i pattern caratteristici delle varie tecniche chitarristiche, con prestazioni elevate nelle classi più frequenti e lievi difficoltà nei casi in cui le caratteristiche spettrali risultano parzialmente sovrapposte.

5.2. Confusion Matrix

La matrice di confusione è uno strumento essenziale per analizzare in modo dettagliato il comportamento di un classificatore multiclasse, in quanto consente di visualizzare in forma tabellare il numero di predizioni corrette e errate suddivise per classe. Ciascuna cella (i, j) della matrice rappresenta il numero di esempi appartenenti alla classe reale i che sono stati predetti come classe j dal modello [19].

Nel nostro caso, la matrice di confusione è stata ottenuta tramite il modulo `confusion_matrix()` della libreria `scikit-learn`. Questa rappresentazione si è rivelata particolarmente utile per individuare le classi che il modello tende a confondere più frequentemente e per analizzare le affinità acustico-spettrali tra diverse tecniche chitarristiche. In particolare, ha permesso di evidenziare quali coppie di suoni presentano pattern simili nella rappresentazione tempo-frequenza, rendendo più complessa la distinzione automatica da parte della rete neurale.

Dall'analisi sono emersi i seguenti pattern ricorrenti:

- Elevata accuratezza diagonale per la maggior parte delle classi, indicativa di un buon apprendimento delle distinzioni principali.
- Alcune confusioni sistematiche tra classi acusticamente simili, come *slide* e *slap*, che condividono componenti spettrali sovrapposte nella gamma media.
- Una lieve riduzione delle prestazioni su classi meno rappresentate nel dataset, suggerendo una possibile influenza dello sbilanciamento delle classi.

La confusion matrix ha quindi permesso di validare il buon comportamento generale del modello, ma ha anche evidenziato aree specifiche di debolezza che potrebbero essere affrontate in futuri miglioramenti, ad esempio tramite tecniche di data augmentation mirata o metodi di riequilibrio dei dati.

Nel complesso, l'uso della confusion matrix si è rivelato uno strumento fondamentale non solo per valutare la performance, ma anche per diagnosticare problemi e orientare il miglioramento del sistema di classificazione.

5.3. Grafici di Apprendimento

Durante l'addestramento del modello, sono stati tracciati i grafici dell'andamento della funzione di perdita (*loss*) e dell'accuratezza (*accuracy*). Questi grafici, comunemente noti come *learning curves*, forniscono un supporto visivo fondamentale per comprendere il comportamento del modello nel tempo e identificare possibili problemi di *underfitting*, *overfitting* o instabilità durante la fase di ottimizzazione [20].

Le curve sono state ottenute salvando i valori di accuratezza e perdita ad ogni epoca attraverso l'oggetto `history` restituito dalla funzione `model.fit()` di Keras. I dati sono stati poi rappresentati graficamente tramite la libreria `matplotlib`.

Dall'analisi dei grafici di apprendimento sono state osservate le seguenti evidenze:

- Una progressiva riduzione della perdita sul training set accompagnata da un aumento dell'accuratezza, segno che il modello apprende efficacemente dalle osservazioni.
- Un'accuratezza di validazione stabile e prossima a quella di addestramento, indicativa di una buona capacità di generalizzazione.
- In alcune configurazioni con un numero elevato di epoche, si è notata una lieve divergenza tra le curve di training e validazione, suggerendo un iniziale inizio di overfitting. In questi casi, l'introduzione di tecniche di regolarizzazione (come il dropout) ha contribuito a stabilizzare l'addestramento.

L'analisi delle learning curves ha dunque svolto un ruolo cruciale nella fase di tuning e ottimizzazione del modello, permettendo di scegliere un numero di epoche appropriato e verificare la correttezza delle scelte di iperparametri adottate.

6. Discussione

I risultati sperimentali ottenuti nel presente studio confermano l'efficacia dell'approccio basato su rappresentazioni tempo-frequenza e reti neurali convoluzionali (CNN) per la classificazione automatica di segnali audio di chitarra. L'accuratezza complessiva raggiunta, associata a metriche dettagliate come precision, recall e F1-score, evidenzia un apprendimento solido da parte del modello, specialmente nel caso dei Mel-spectrogrammi, che si sono rivelati particolarmente efficaci come input per modelli CNN.

Il confronto con le MFCC ha messo in luce differenze significative: sebbene entrambe le rappresentazioni siano valide, i Mel-spectrogrammi offrono una maggiore ricchezza informativa, che permette alla rete di cogliere sfumature più fini nei pattern audio. Le MFCC, d'altra parte, si dimostrano più leggere in termini computazionali e sufficienti per compiti di classificazione più semplici o in ambienti a risorse limitate.

L'utilizzo del transfer learning con VGG16 ha portato a risultati promettenti, soprattutto in scenari in cui il dataset disponibile è di dimensioni contenute. Il riuso di architetture

pre-addestrate ha permesso di ridurre i tempi di addestramento e ha offerto un miglioramento iniziale delle performance, anche se in alcuni casi si è resa necessaria una fase di fine-tuning per adattare meglio le caratteristiche del dominio audio.

Dal punto di vista dell'analisi degli errori, si è osservato che le classi con caratteristiche acustiche simili tendono a generare ambiguità nel processo di classificazione. Questo suggerisce l'opportunità di esplorare in futuro approcci basati su modelli più sofisticati (es. modelli ibridi CNN–RNN), o tecniche di data augmentation per ampliare la varietà dei campioni disponibili.

Infine, le estensioni sperimentali relative alla visualizzazione e all'ascolto dei dati hanno conferito un valore aggiunto al lavoro, facilitando l'analisi qualitativa dei risultati e aprendo possibilità per applicazioni didattiche e interattive in ambito musicale e multimediale.

Nel complesso, il sistema sviluppato si è dimostrato robusto, interpretabile e potenzialmente estendibile a domini sonori più complessi, ponendo le basi per sviluppi futuri nel campo dell'analisi automatica di segnali audio musicali.

7. Conclusioni

In questo lavoro è stata affrontata la problematica della classificazione automatica di segnali audio di chitarra mediante tecniche di deep learning, con particolare enfasi sull'impiego di rappresentazioni tempo-frequenza e reti neurali convoluzionali. A partire da un dataset eterogeneo contenente suoni appartenenti a 13 differenti tecniche chitarristiche, è stata sviluppata una pipeline completa che integra il preprocessing, l'estrazione delle caratteristiche, la costruzione del modello, l'addestramento e la valutazione.

I risultati ottenuti hanno confermato che i Mel-spectrogrammi, per la loro capacità di rappresentare visivamente la densità spettrale nel tempo, costituiscono una scelta particolarmente efficace per l'alimentazione di modelli CNN. Le MFCC, pur essendo meno ricche informativamente, si sono dimostrate una valida alternativa in contesti a bassa complessità computazionale.

L'introduzione di estensioni sperimentali come il transfer learning con VGG16 e la possibilità di visualizzare e ascoltare i dati ha arricchito ulteriormente il progetto, rendendolo più robusto, interpretabile e potenzialmente riutilizzabile in ambiti applicativi diversi. L'analisi degli errori e la valutazione delle metriche hanno permesso di identificare sia i punti di forza del sistema che le sue aree di miglioramento.

In sintesi, il progetto ha dimostrato la validità dell'approccio adottato per il task di classificazione audio, offrendo una base solida per estensioni future, come l'applicazione a dataset più ampi, l'uso di architetture neurali avanzate o l'integrazione in sistemi interattivi per il riconoscimento sonoro in tempo reale.

Author Contributions: O.P. e D.F. hanno contribuito in egual misura a tutte le fasi del lavoro, inclusa la concettualizzazione, la metodologia, lo sviluppo software, l'analisi, la scrittura e la revisione del manoscritto. Entrambi gli autori hanno letto e approvato la versione finale dell'articolo.

Funding: Questa ricerca non ha ricevuto alcun finanziamento esterno.

Abbreviazioni

Le seguenti abbreviazioni sono utilizzate nel presente manoscritto:

CNN	Convolutional Neural Network (Rete Neurale Convoluzionale)
MFCC	Mel-Frequency Cepstral Coefficients (Coefficienti Cepstrali in Scala Mel)
STFT	Short-Time Fourier Transform (Trasformata di Fourier a Breve Termine)
DCT	Discrete Cosine Transform (Trasformata Discreta del Coseno)
AI	Artificial Intelligence (Intelligenza Artificiale)
RMS	Root Mean Square (Valore Quadratico Medio)
VGG16	Visual Geometry Group 16-layer network

References

1. Hershey, S.; Chaudhuri, S.; Ellis, D.P.; Gemmeke, J.F.; Jansen, A.; Moore, R.C.; Plakal, M.; Platt, D.; Saurous, R.A. CNN architectures for large-scale audio classification. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* **2017**, pp. 131–135.
2. Purwins, H.; Li, B.; Virtanen, T.; Schlüter, J.; Chang, S.Y.; Sainath, T. Deep Learning for Audio Signal Processing. *IEEE Journal of Selected Topics in Signal Processing* **2019**, *13*, 206–219.
3. Choi, K.; Fazekas, G.; Sandler, M.; Cho, K. Convolutional recurrent neural networks for music classification. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* **2017**, pp. 2392–2396.
4. Fu, Z.; Lu, G.; Ting, K.M.; Zhang, D. A survey on audio-based music classification and annotation. *IEEE Transactions on Multimedia* **2011**, *13*, 303–319.
5. Pons, J.; Serra, X. Musicnn: Pre-trained convolutional neural networks for music audio tagging. *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* **2019**, pp. 186–190.
6. McFee, B.; Raffel, C.; Liang, D.; Ellis, D.P.; McVicar, M.; Battenberg, E.; Nieto, O. librosa: Audio and music signal analysis in Python. In Proceedings of the Proceedings of the 14th Python in Science Conference, 2015, Vol. 8, pp. 18–25.
7. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
8. Stevens, S.S.; Volkmann, J.; Newman, E.B. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America* **1937**, *8*, 185–190.
9. Logan, B. Mel frequency cepstral coefficients for music modeling. In Proceedings of the Proceedings of the International Symposium on Music Information Retrieval (ISMIR), 2000.
10. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
11. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* **2014**, *15*, 1929–1958.
12. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**.
13. Chollet, F. *Deep Learning with Python*; Manning Publications Co., 2017.
14. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2014, pp. 3320–3328.
15. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations (ICLR), 2015.
16. AI, F. Working with Audio Signals in Python. *Fritz AI blog* **2023**.
17. Amershi, S.; Weld, D.; Vorvoreanu, M.; Fourney, A.; Nushi, B.; Collisson, P.; Suh, J.; Cakmak, M.; Kamar, E.; Horvitz, E. Guidelines for Human-AI Interaction. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* **2019**, pp. 1–13.
18. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies* **2011**, *2*, 37–63.
19. Fawcett, T. An introduction to ROC analysis. *Pattern Recognition Letters* **2006**, *27*, 861–874.
20. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016.