Classificazione suoni sottomarini attraverso l'uso di spettrogrammi e CNN

Autori: Daniela Amendola, Giovanni Arcangeli, Francesco Maddaloni
Tutor: Dott. Benedetto Simone

[1]Università degli studi di Salerno

Abstract

Questo progetto mira a sviluppare un classificatore per distinguere suoni sottomarini di origine animale e umana attraverso l'uso di spettrogrammi e reti neurali convoluzionali (CNN). Per raggiungere questo obiettivo, sono stati utilizzati diversi modelli di deep learning, tra cui GoogleNet, ResNet50 e AlexNet. Il lavoro ha previsto una prima fase di analisi dei dati audio in possesso con conseguente resampling, trimming e trasformazione dei dati in spettrogrammi. Successivamente, è stato effettuato un bilanciamento sia delle sottoclassi che delle due macro classi presenti nel dataset. Gli addestramenti hanno permesso di ottenere un ottimo classificatore binario con valori delle metriche di Precision, Recall ed F1-score intorno al 95-98%. Gli addestramenti multiclasse invece hanno dato risultati più soddisfacenti sul dataset non bilanicato, con metriche intorno 20-34%,mentre sul dataset bilanciato i valori si sono ridotti al 10%.

Key-words

Audio sottomarini, Classificazione, Spettrogrammi, Deep learning, Data augmentation, Trasformata di Fourier, GoogleNet, ResNet50, AlexNet.

1 Introduzione

L'ambiente marino è una risorsa importante e fondamentale per il ciclo della vita sulla Terra. Esso ospita una vasta gamma di organismi viventi, dai microrganismi planctonici agli enormi mammiferi marini, passando per pesci, coralli e alghe. Inoltre, gli oceani e i mari rappresentano una fonte inesauribile di risorse alimentari, materiali, energetiche e farmaceutiche per l'umanità. Tuttavia, le attività umane stanno compromettendo sempre più l'equilibrio di questi ecosistemi. L'inquinamento acustico marino, in particolare, rappresenta una minaccia crescente per la fauna marina, alterando i comportamenti naturali e causando stress agli animali.

Il monitoraggio e la classificazione dei suoni sottomarini sono strumenti cruciali per comprendere e mitigare questi impatti. I suoni sottomarini possono fornire informazioni preziose sulla biodiversità, sulla presenza di specie minacciate e sui cambiamenti nei comportamenti animali. In questo contesto, l'uso di tecnologie avanzate come i spettrogrammi e le reti neurali convoluzionali (CNN) si rivela particolarmente efficace. I spettrogrammi, che trasformano i segnali audio in rappresentazioni visive, consentono alle CNN di analizzare i dati con alta precisione.

Il primo passo è stato quello di effettuare uno studio sui lavori precedentemente svolti in questo campo andando ad approfondire le varie rappresentazioni possibili di dati audio [1] [2] [3] [4]. Dopo un attento studio si è deciso di proseguire sulla strada dell'analisi audio mediante spettrogrammi in quanto buona parte degli studi scientifici letti lavorano su altri tipi di rappresentazioni (Mel-Spectograms, Scalogrammi). L'idea era quella di tentare una strada altrimenti ignorata per motivi di performance legati alle CNN. Questo progetto mira a sviluppare un classificatore per distinguere suoni sottomarini di origine

animale e umana attraverso l'uso di spettrogrammi e CNN. Per raggiungere questo obiettivo, sono stati utilizzati diversi modelli di deep learning, tra cui GoogleNet, ResNet50 e AlexNet. Il lavoro ha previsto una prima fase di analisi dei dati audio in possesso con conseguente resampling, trimming e trasformazione dei dati in spettrogrammi. Successivamente, è stato effettuato un bilanciamento sia delle sottoclassi che delle due macro classi presenti nel dataset.

Il dataset utilizzato, fornito dal nostro Tutor, è stato ottenuto ricercando e selezionando file audio in formato .wav e .mp3 presenti in numerosi datasets disponibili online, come Watkins Marine Mammals Sound, Marine Mammals Bioacoustic (MMB) of Australia and Antarctica, Deepship e A Collection of Sounds from the Sea, NOAA. Questo dataset comprende un totale di 2663 campioni, suddivisi tra suoni di origine animale e antropogenica.

L'obiettivo finale del progetto è quello di ottenere prima un classificatore binario per poi raffinare tale risultato fino all'ottenimento di un classificatore multi classe. Gli addestramenti hanno permesso di ottenere un ottimo classificatore binario con valori delle metriche di Precision, Recall ed F1-score intorno al 95-98%. Gli addestramenti multiclasse invece hanno dato risultati più soddisfacenti sul dataset non bilanciato, con metriche intorno al 20-34%, mentre sul dataset bilanciato i valori si sono ridotti al 10%. Nei capitoli successivi verranno affrontate e approfondite le seguenti sezioni:

- **Background** in cui verrà fornita una base iniziale dei concetti fondamentali;
- **Stato dell'arte** con gli studi precedentemente condotti da cui abbiamo avuto modo di comprendere meglio il dominio del problema;
- **Metodologia** in cui vengono affrontati i problemi relativi al pre-processing e alla data augmentation;
- **Addestramento e validazione** con i relativi esperimenti condotti sugli addestramenti ed i risultati annessi;
- **Testing** in cui vengono riportati i risultati dei test effettuati sui modelli addestrati;
- **Conclusioni** per avere una rapida panoramica dei risultati ottenuti e delle varie considerazioni che ne conseguono

2 Background

2.1 Trasformata di Fourier

La ***trasformata di Fourier*** permette di scomporre una funzione non periodica (un' onda audio nel nostro caso nel dominio del tempo) in una combinazione lineare di funzioni con base $e^{j\omega t}$ dove $\omega \in \mathbb{R}$.

$$F(\omega) = \int_{-\infty}^{+\infty} f(t) e^{-j\omega t}\, dt \tag{1}$$

purché la funzione soddisfi la condizione di assoluta sommabilità:

$$\int_{-\infty}^{+\infty} |f(t)|\, dt < \infty \tag{2}$$

2.2 Teorema del campionamento di Nyquist-Shannon

Il ***Teorema del campionamento di Nyquist-Shannon*** ci dice che data una funzione la cui trasformata di Fourier sia nulla al di fuori di un certo intervallo di frequenze (ovvero un segnale a banda limitata), nella sua conversione analogico-digitale la **minima frequenza di campionamento necessaria** per **evitare** aliasing e **perdita di informazione** nella ricostruzione del segnale analogico originario (ovvero nella riconversione digitale-analogica) deve essere **maggiore del doppio della sua frequenza massima**. Dunque, se la frequenza massima di $f(x)$ è ω_{max}, la frequenza di campionamento deve essere maggiore di $2\omega_{max}$.

3 Stato dell'arte

La classificazione dei rumori sottomarini è un campo in rapida evoluzione, caratterizzato da diverse tecniche di astrazione delle caratteristiche e metodologie di ML. Yunqi Zhang e Qunfeng Zeng [5] si sono concentrati principalmente sull'analisi dei suoni delle navi utilizzando il dataset DeepShip. In questo primo studio è stata proposta una nuova metodologia chiamata Multi-scale short-time Fourier transform (MS-STFT) per migliorare l'informazione a bassa frequenza e mantenere i dettagli aumentando il numero di canali. Inoltre, è stato sviluppato un approccio di data augmentation e un'architettura Ladder-like Encode (LE) per incrementare la generalizzabilità del modello e l'accuratezza della classificazione. Infine, il metodo Frequency-CAM (FC) è stato introdotto per analizzare le bande di frequenza di interesse durante i compiti di classificazione. Questa integrazione, denominata MSLEFC, ha raggiunto un'accuratezza dell'82.94% e del 96.06%.

Un altro studio [6] ha utilizzato i dataset DeepShip e ShipsEar, sono stati utilizzati i Mel spettrogramma concatenati a sub-bande per amplificare i rumori a bassa frequenza emessi dalle navi, migliorando le caratteristiche attraverso la concatenazione di multispettri. Viene inoltre introdotto un meccanismo di attenzione multidominio per potenziare una semplice rete residuale, sviluppando così il modello leggero CFTANet. Questo sistema ha ottenuto accuratezze di riconoscimento del 90.60% e del 96.40%.

Nel terzo studio [7], sono stati analizzati i dati audio grezzi del Freesound Dataset (FSD) di Kaggle e sono stati convertiti in rappresentazioni di spettrogrammi, utilizzando poi reti neurali convoluzionali (CNN) per la classificazione. Sono stati testati due approcci: una propria architettura CNN e il trasferimento di apprendimento usando la rete pre-addestrata VGG19. Il miglior risultato ottenuto con l'architettura propria ha mostrato una precisione media ponderata delle etichette (LWLARP) di 0.813 e una top-5 accuracy del 88.9% su 80 classi sonore.

Mishachandar e Vairamuthu [8] hanno utilizzato un dataset personalizzato, combinando dati da varie fonti, tra cui: cetacei, pesci, invertebrati marini, suoni antropogenici, naturali e suoni oceanici non identificati dai registri acustici passivi. Il preprocessing ha incluso la pulizia degli audio e la conversione in mono-canale, mentre la data augmentation ha coinvolto tecniche come il time shifting e il pitch shifting. L'addestramento è stato condotto su spettrogrammi logaritmici, evidenziando l'importanza della rappresentazione visiva delle frequenze per l'analisi audio. Il metodo proposto ha ottenuto un'accuratezza del 96.1% nella classificazione, distinguendo efficacemente i sistemi acustici naturali da quelli artificiali.

Un altro studio [9] ha creato un proprio dataset non accessibile pubblicamente, acquisendo i dati con un sistema specifico e generando rumori bianchi gaussiani per la data augmentation. Sono stati utilizzati sia Support Vector Machine (SVM) che reti neurali convoluzionali (CNN) per la classificazione, verificando i risultati con dati originali e con dati a diversi rapporti segnale-rumore (SNR). I risultati hanno mostrato che le caratteristiche di livello di rumore (NL) e densità spettrale di potenza (PSD) hanno fornito le migliori performance, con accuratezze di classificazione del 98.95% e 97.65% rispettivamente quando il SNR era -10 dB.

Infine, nello studio condotto da Mishachandar e Vairamuthu [8], è stata proposta una tecnica che combina CNN e Mel-spectrogram per il riconoscimento automatico dei rumori delle navi. La frequenza di campionamento è stata uniformata e sono state applicate varie tecniche di data augmentation. L'addestramento ha sfruttato sia le rappresentazioni temporali che quelle frequenziali degli audio, migliorando le capacità del modello di riconoscere e classificare i suoni. Il sistema ha raggiunto un'accuratezza del 99% nel rilevare la presenza di navi e classificarle in diverse categorie

Nel nostro studio, adottiamo un approccio innovativo utilizzando spettrogrammi per l'analisi audio. A differenza degli studi precedenti, ci concentriamo sull'analisi sia dei suoni umani che di quelli animali. Questo permette una comprensione più ampia e una maggiore generalizzazione del modello. Gli spettrogrammi forniscono una rappresentazione visiva dettagliata delle variazioni di frequenza nel tempo, consentendo una più accurata identificazione delle caratteristiche distintive dei suoni analizzati.

4 Metodologia

In questo capitolo viene descritta la metodologia adottata per il presente lavoro, con le relative motivazioni delle scelte effettuate.

4.1 Data Analysis

Il primo step del progetto è stato quello di analizzare il Dataset per estrarre informazioni sugli audio presenti in vista delle successive operazioni di pre-processing da effettuare. In particolare, dopo aver analizzato ed eliminato degli audio duplicati si è proseguito studiando durata, bitdepth, frequenza di campionamento, frequenza massima di ogni audio e numero di canali audio.

Durante una prima analisi esplorativa del dataset, sono stati identificati 50 file duplicati all'interno di due classi diverse. Si è quindi deciso di riascoltare ed analizzare i singoli audio per evitare l'eliminazione di 50 campioni. Il risultato dell'analisi, riportato nelle tabelle qui di seguito, ha permesso di identificare l'ubicazione corretta di 23 dei 50 file duplicati.
Tutte le operazioni sono state effettuate su un file `.csv` sul quale erano stati estratti i path dei file audio. Ciò è stato fatto per non intaccare fisicamente il Dataset.

ID	**Descrizione**	**Preferenza**
72021005	L'audio è molto lungo e potrebbe essere stato inserito in 2 classi perchè sono presenti entrambi i suoni? I suoni gravi assomigliano a quelli presenti nella cartella Bearded Seal ed al verso generico di un foca	-
7202100T	È molto grave e prolungato rispetto ai suoni lievi di Bearded Seal	Bowhead whale
7202100V	Io sento due versi distinti. Si dovrebbe spezzare?	-
7202100Z	Mi sembra un verso da Balena, super "sottile"	Bowhead Whale
78018002	Molto lungo: Per i suoni molto acuti che si sentono lo metterei in Bearded Seal	Bearded Seal
78018003	Il suono sovrastante è un suono acuto che assimilerei ad una balena, ma in mezzo si sento un suono più duro assimilabile ad una foca. Credo contenga entrambi i suoni sovrapposti. Lo metterei in Bowhead whale essendo il suono sovrastante quello che assimilerei ad una balena.	-
7801800B	Non saprei, sembra molto grave per essere una foca ma è al limite. Sono molto indeciso su questo.	Bowhead Whales
7801800D	Assomiglia a 8800601U, lo assimilerei a BowHead Whales	Bowhead Whale
7801800H	-	Bowhead Whale
7801800j	molto dubbioso	-

Table 1: Classificazione dei Suoni: Bearded Seal vs Bowhead Whale

ID	**Descrizione**	**Preferenza**
7702800U	Dagli audio presenti in Long Finned lo piazzerei lì, in Sperm whale ci sono un sacco di audio molto lunghi o con suoni completamente diversi e di "Toc toc toc". In termini di classificazione forse sarebbe più utile averlo in Long Finned Pilot Whale.	Long Finned Pilot Whale
7702800V	Uguale a sopra	Long Finned Pilot Whale
7702800X	-	Long Finned Pilot Whale
7702801F	-	Long Finned Pilot Whale
7702801M	Non si sente nulla a parte un piccolo tocco, lo metterei in Sperm Whale	Sperm Whale

Table 2: Classificazione dei Suoni: Sperm Whale vs Long Finned Pilot Whale

ID	Descrizione	Preferenza
84016002	Mi sembra molto simile agli altri audio in Sperm Whale	Sperm Whale
8401600b	21 minuti. In molti punti si sente i ticchettii tipici degli audio di Sperm Whale.	Sperm Whale

Table 3: Classificazione dei Suoni: Sperm Whale vs Short Finned (Pacific) Pilot Whale

ID	Descrizione	Preferenza
84021003	L'audio è molto disturbato ma ha quello sfregolio presente in altri audio in Pantropical. Non ha i ticchettii di Sperm Whale. Lo lascerei in Pantropical	Pantropical Spotted Dolphin

Table 4: Classificazione dei suoni: Pantropical Spotted Dolphin vs Sperm Whale

ID	Descrizione	Preferenza
9101200B.wav	Sono tutti indistinguibili però non avendo quei ticchettii decisi che stanno in Sperm whale li lascerei in Melon Headed	Melon Headed Whale
9101200K.wav	-	Melon Headed Whale
9101201E.wav	-	Melon Headed Whale
91012022.wav	-	Melon Headed Whale
91012048.wav	-	Melon Headed Whale
91012049.wav	-	Melon Headed Whale

Table 5: Classificazione dei suoni: Melon Headed Whale vs Sperm Whale

Un primo step è stato quello di ricavare informazioni sulla durate in secondi dei campioni per capire come effettuare tagli e spezzare quindi gli audio in frammenti di x secondi (`Trimming`). Attraverso la libreria librosa è stato possibile accedere a questo dato velocemente ottenendo il grafico riportato di seguito:

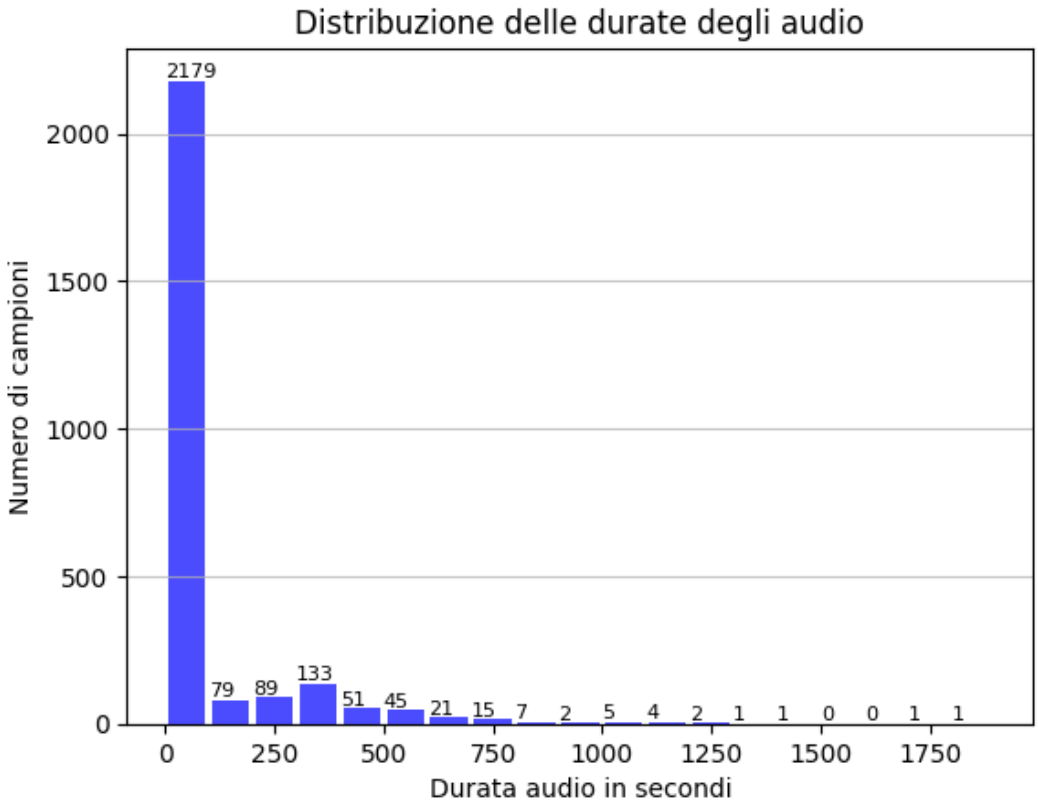

Figure 1: Plot durate audio

Si è deciso quindi di studiare nel dettaglio i risultati per decidere come effettuare successivamente le operazioni di trimming. In particolare gli audio presentano una durata media di 75 secondi, una durata massima di 1887 secondi ed una durata di 0.05 secondi.
Prendendo in considerazione il valore mediano di **3.48** si è deciso successivamente di effettuare trimming a 3 secondi.

Successivamente sono state estratte rispettivamente le frequenze di campionamento di tutti i file nonché la frequenza massima di questi ultimi. Tale operazione è stata effettuata per ottenere informazioni su come effettuare il resampling dei campioni presenti nel dataset.
Il resampling è necessario per uniformare le dimensioni di tutti i dati in modo che gli audio abbiano tutti la stessa base temporale e per agevolare la normalizzazione dei dati audio per il training.
Di seguito sono riportati i grafici relativi rispettivamente alle distribuzioni delle frequenze di campionamento e delle frequenze massime per campione.

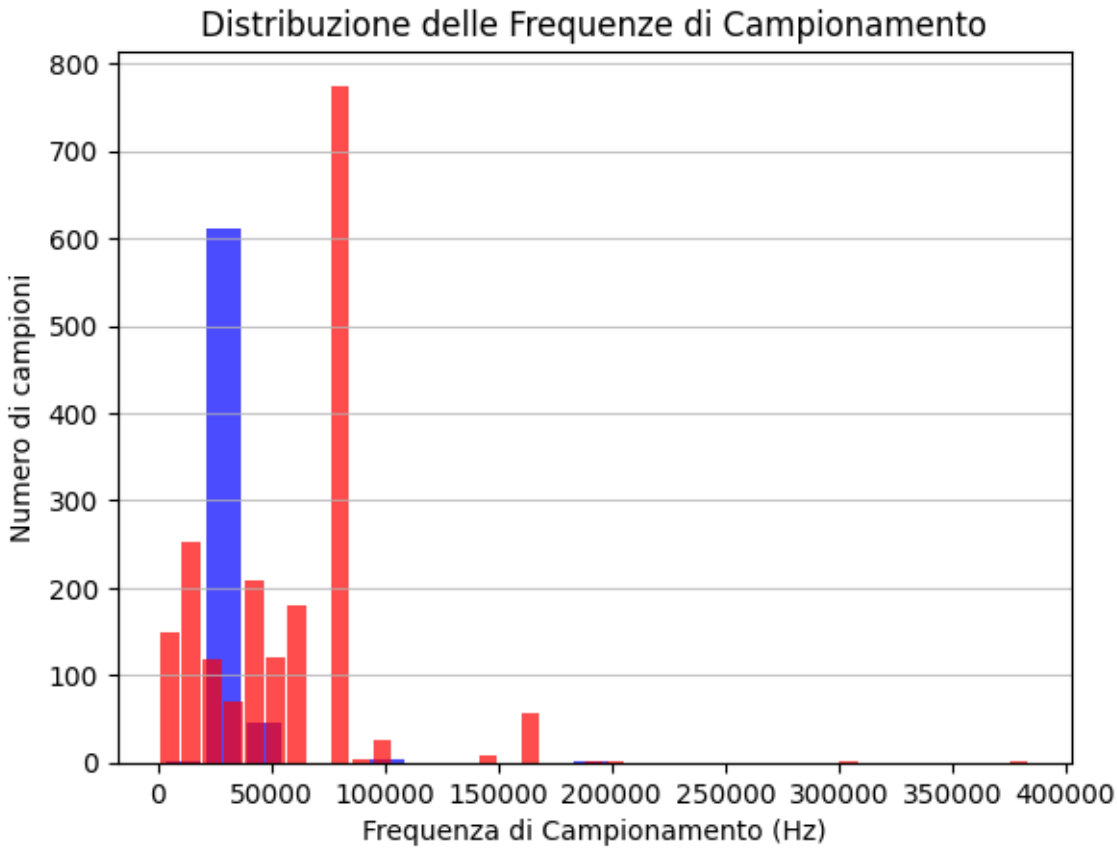

Figure 2: Distribuzione frequenze di campionamento: Target=Blu, Non-Target = Rosso

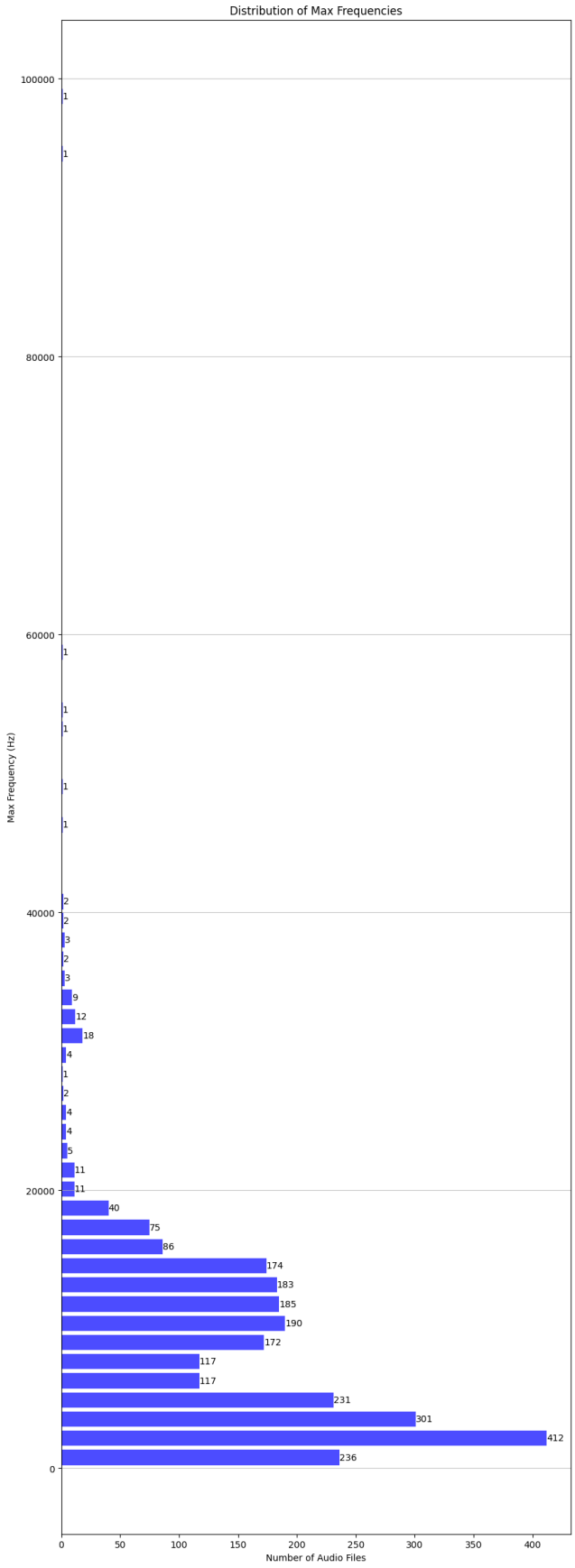

Figure 3: Distribuzione frequenze massime

Dal quest'ultimo studio è stato possibile estrarre la frequenza massima raggiunta dagli audio per poter determinare al nuova frequenza per il ricampionamento. In particolare gli audio raggiungono un frequenza massima di circa 99Khz.
Utilizzando il Teorema del campionamento di Nyquist-Shannon è stato deciso di prendere in esame la frequenza massima raggiunta dai campioni. Di conseguenza il valore ottenuto per il resampling sarebbe di `198.834Hz`, cioè `198Mhz`. Si è deciso però di campionare gli audio a `192Mhz` essendo questo un valore di campionamento tipico per i file audio.

Un'altra importante analisi è stata quella relativa alla distribuzione del numero di canali audio, cioè verificare quanti campioni presentassero uno o due canali. Di seguito è riportato il grafico estratto:

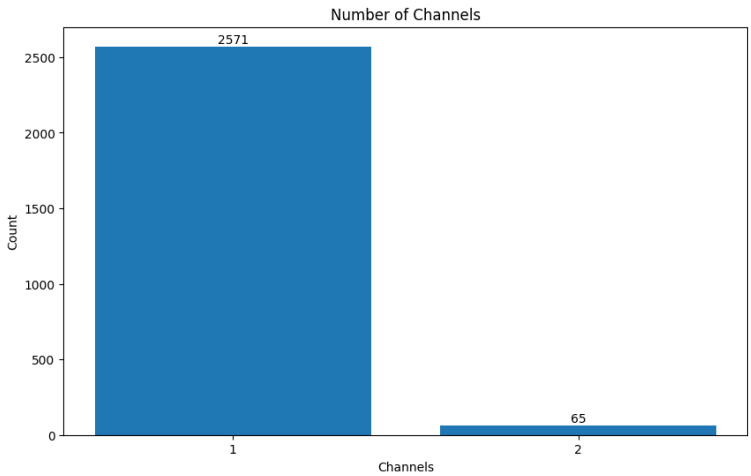

Figure 4: Distribuzione canali audio

Essendo il numero di audio in due canali decisamente inferiori rispetto agli audio in mono canale e per semplicità di analisi nelle fasi successive, si è deciso di convertire tutti gli audio bi-canale in mono-canale.

Penultima analisi effettuata in questa fase è stata rivolta sulla distribuzione dei Bit Depth. Tale valore indica la quantità di bit di informazioni presenti in ogni campione e ne descrive, quindi, la risoluzione sia essa a 8, 16, 24 e 32 bit. Di seguito è riportato il grafico ottenuto in fase di analisi: Parte dei campioni a causa del formato non è stato riportato nel grafico.

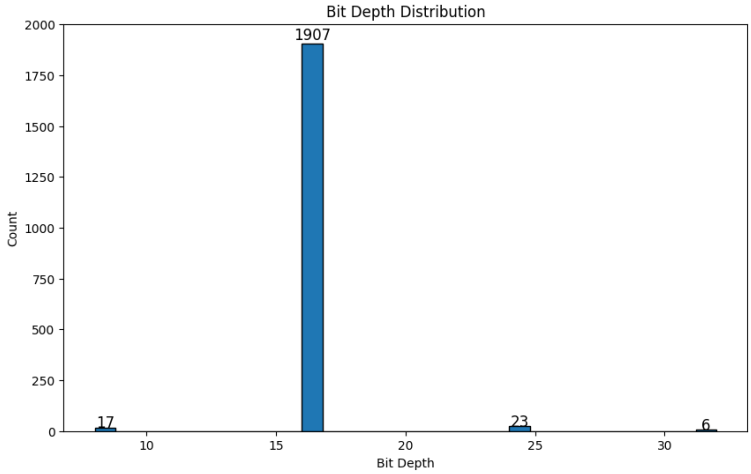

Figure 5: Distribuzione dei Bit Depth

Come è possibile notare dal grafico la grande maggioranza dei campioni presenta un Bit Depth di 16 bit. Si è così deciso in fase successiva di convertire i restanti campioni a 16 bit.

Infine è stata riportata in un grafico la distribuzione di tutte le classi presenti nel dataset per avere una visione chiara del numero di campioni a disposizione per ogni in classe. Ciò sarà molto utile in fase di Data Augmentation.

Di seguito è riportato tale grafico:

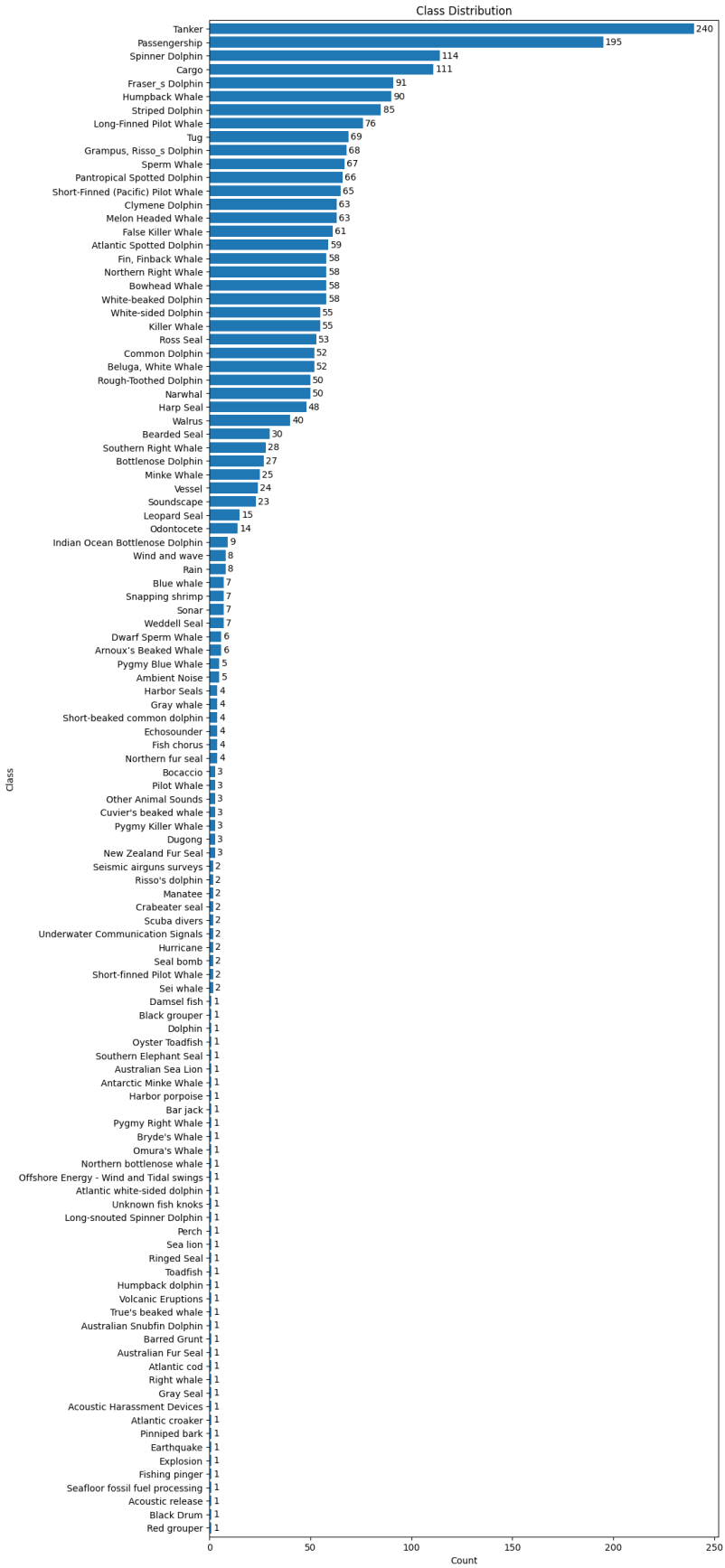

Figure 6: Distribuzione delle classi

Unendo le osservazioni fatte nella sezione precedente i dati sono stati rielaborati effettuando le seguenti operazioni:

- Resampling file audio a 192Mhz;
- Trimming audio a 3 secondi;
- Conversione di tutti gli audio a mono canale;
- Equalizzazione audio a 16 bit di bit depth.

La ***Trasformata di Fourier***, quindi, nel nostro caso consente di passare dalla rappresentazione di un segnale nel dominio del tempo (tempo/ ampiezza) alla rappresentazione nel dominio della frequenza (frequenza/ ampiezza), cioè al suo spettro.

A livello di implementazione è stata utilizzata la `stft` (Short Time Fourier Trasform), una versione velocizzata ed ottimizzata della trasformata.
Il risultato di questa operazione è stato un nuovo Dataset con la stessa struttura dell'originale con all'interno i sotto-spettrogrammi di tutti gli audio.
Di seguito un esempio di un sotto-spettrogramma estratto da un suoni di un capodoglio recuperato dal dataset Watkins Marine Mammals Sound:

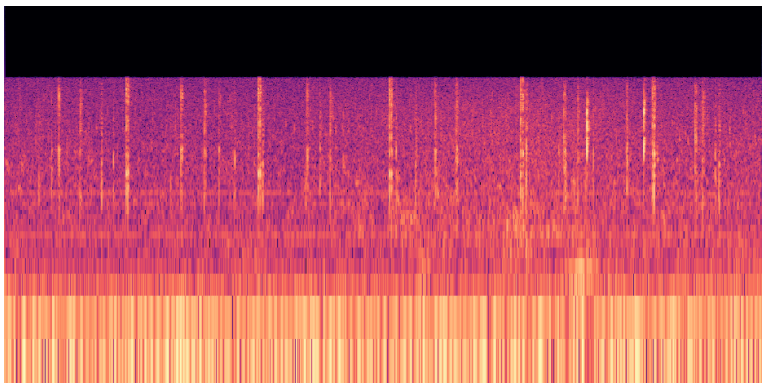

Figure 7: Sotto-Spettrogramma di un suono di un capodoglio dal dataset Watkins Marine Mammals Sound

Il dataset finale, ottenuto dalle operazioni di resampling, comprende un totale 67916 divisi nelle due classi descritte nelle precedenti sezioni, in particolare 10421 per i suoni di natura antropologica (Target) e 57.495 per i suoni animali (Non-Target).

4.2 Data Augmentation

A causa del numero ridotto di campioni di diverse classi soprattutto all'interno della cartella Non-Target, relativa ai suoni animali, è stato necessario uno step di Data Augmentation. Questa volta si è deciso di lavorare separatamente sulle classi Target e Non-Target. Prima di iniziare lo step di Data Augmentation si è deciso di analizzare il numero di campioni del nuovo dataset contenente i sotto-spettrogrammi degli audio. A tal proposito è stato redatto un grafico della distribuzione delle classi sia per la cartella Target che Non-Target.

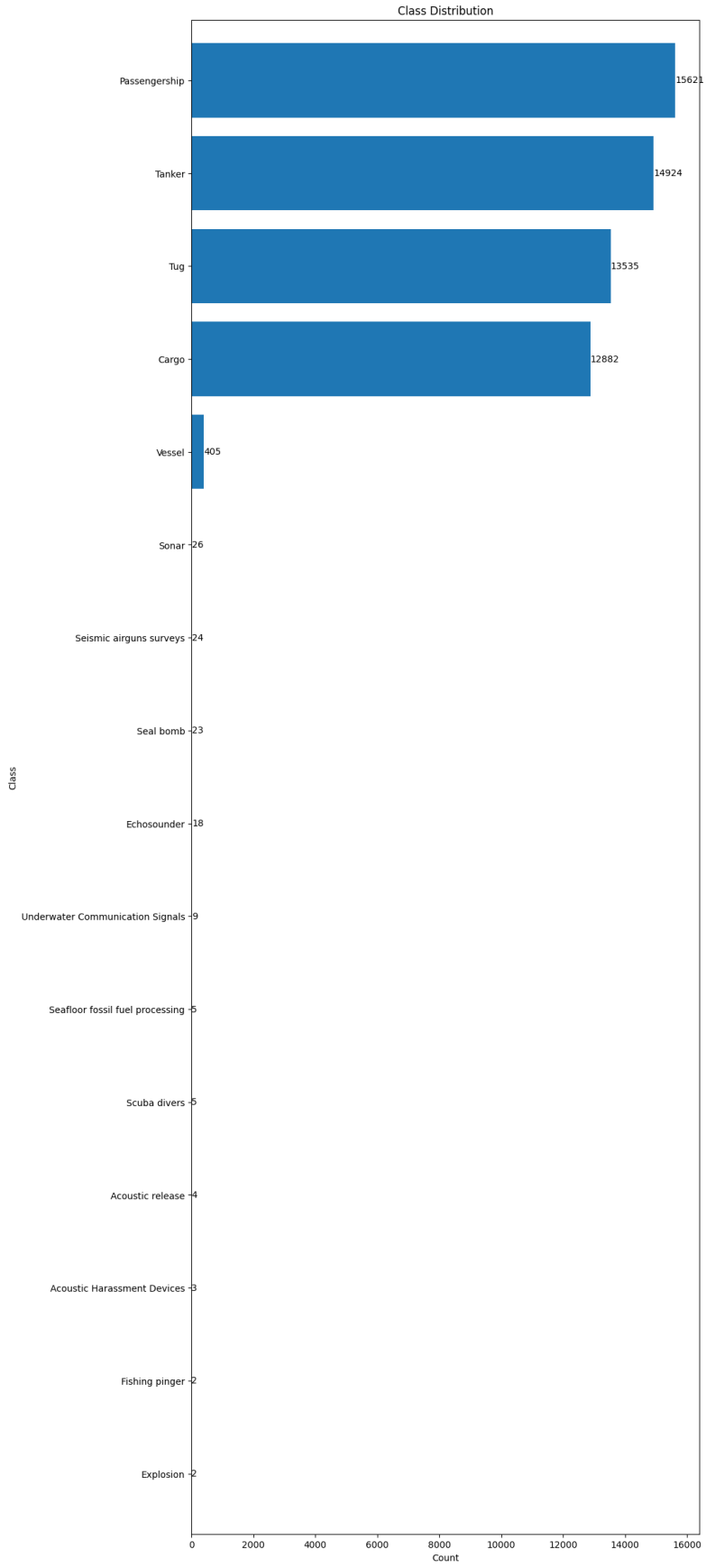

Figure 8: Distribuzione classi sotto-spettrogrammi Target

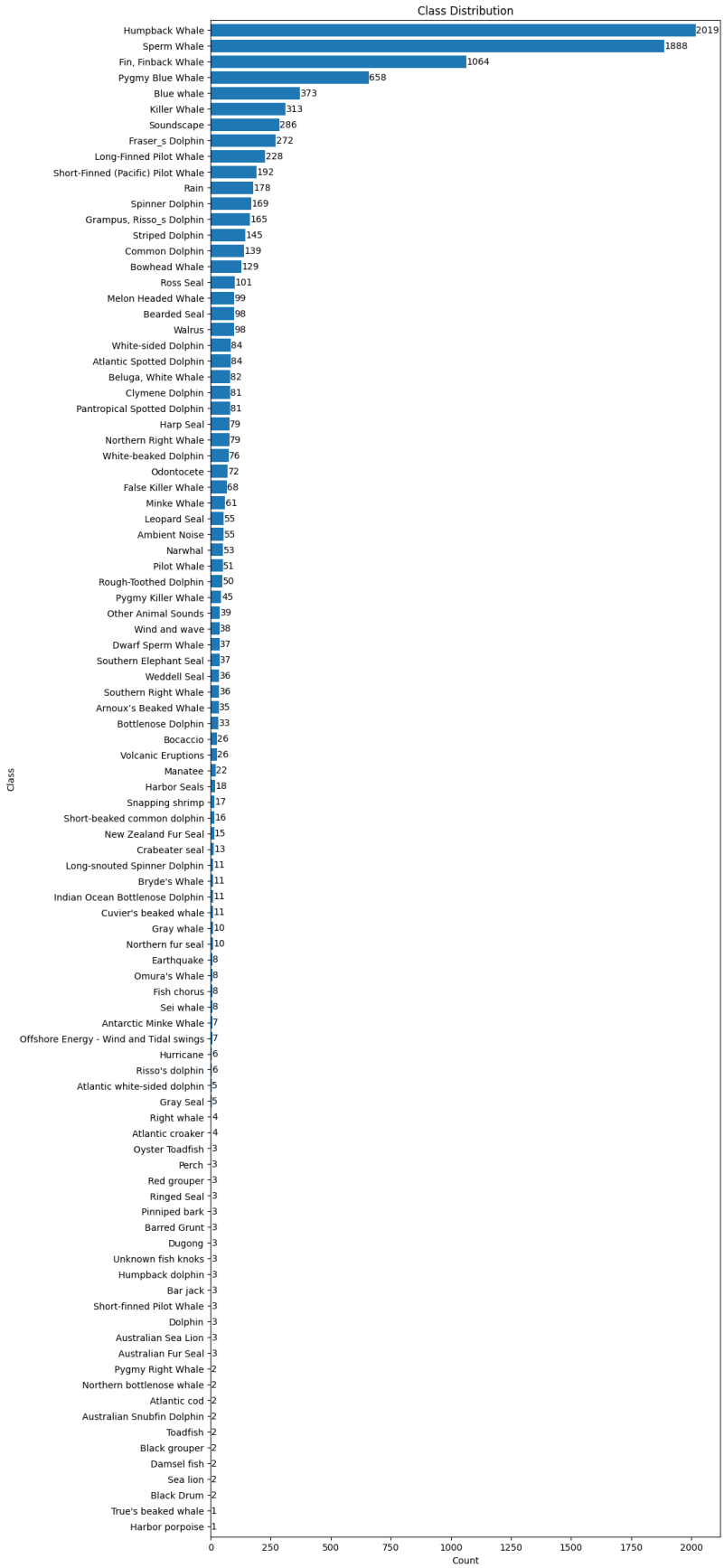

Figure 9: Distribuzione classi sotto-spettrogrammi Non-Target

In una prima istanza, studiando le distribuzioni dei dati, si è deciso di rimuovere le classi contenenti meno di 8 audio distinti, poiché non avrebbero dato un buon contributo durante il training e avrebbero complicato l'operazione di suddivisione del dataset per l'addestramento. La suddivisione è stata fatta in questo modo: `80% Training, 10% Validation, 10% Test`.

Per bilanciare le classi e aumentare le prestazioni, è stato effettuato un lavoro di data augmentation suddiviso in più fasi. Per prima cosa, si è scelto di bilanciare le sottoclassi relative alle singole classi di suoni. Questo bilanciamento è stato effettuato portando il numero di sotto-spettrogrammi di tutte le classi al numero di occorrenze maggiore delle sottoclassi di quella specifica sezione (Target o Non-Target). In questo modo, ciascuna classe all'interno di una sezione avrebbe avuto lo stesso numero di campioni. Per evitare la creazione di duplicati e mantenere la qualità dei dati, è stato eseguito un processo di augmentation controllato. Quindi, sono stati impostati dei range di valori specifici per le varie trasformazioni, in modo tale da non distorcere eccessivamente l'immagine dei sotto-spettrogrammi. Le trasformazioni utilizzate sono state le seguenti:

- **Shift Temporale**: Spostamento dell'intero spettrogramma in avanti lungo l'asse temporale, con l'aggiunta di silenzio. Questo permette di simulare variazioni temporali nei suoni senza alterare la loro frequenza. Tale shift è stato applicato con intervalli di 0,08 secondi in modo da coprire 19 trasformazioni tra i secondi 0,08 e 1,52
- **Aggiunta di Rumore**: Introduzione di rumore bianco per simulare condizioni ambientali diverse e aumentare la robustezza del modello nei confronti del rumore. I valori utilizzati differenziavano di 0,0001 e andavano da 0,0011 a 0,00,29, in modo da non alterare eccessivamente lo spettrogramma.
- **Increase dell'Audio**: Incremento dell'ampiezza del segnale audio, aumentando il volume del suono registrato. Questo aiuta il modello a riconoscere suoni a diversi livelli di intensità. L'incremento è stato applicato aumentando di 0,05 da 1,1 a 2.
- **Decrease dell'Audio**: Riduzione dell'ampiezza del segnale audio, diminuendo il volume del suono registrato. Questa trasformazione aiuta il modello a riconoscere suoni anche a bassa intensità. Il decremento è stato effettuato seguendo lo schema usato per l'increase, andando da 0,15 a 0,95.

Di seguito sono riportate alcune immagini di esempio delle trasformazioni effettuate:

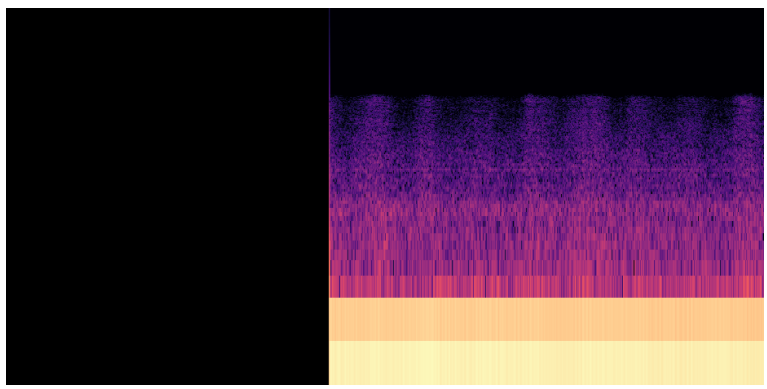

Figure 10: Operazione di shift temporale

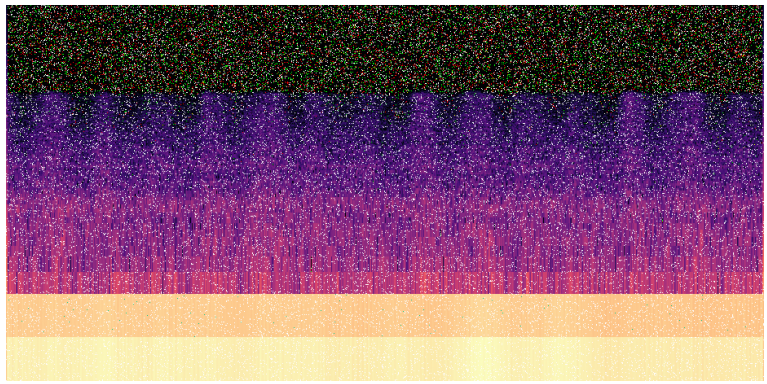

Figure 11: Aggiunta di rumore

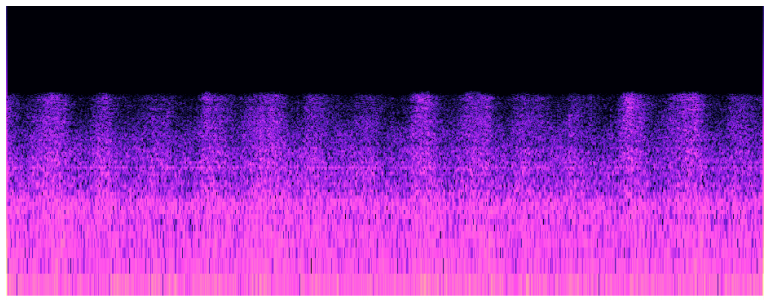

Figure 12: Increase dell'audio

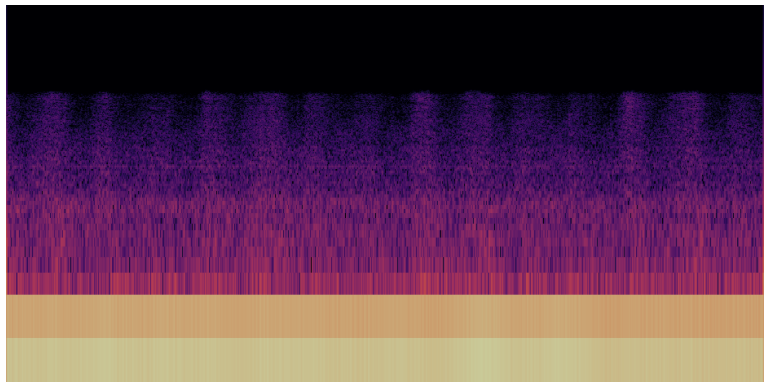

Figure 13: Decrease dell'audio

Successivamente, è stato effettuato un bilanciamento generale tra le classi Target e Non-Target. Portando il numero di occorrenze di ciascuna classe al numero di occorrenze massimo di entrambe le tipologie (Target e Non-Target). Questo passaggio avrebbe garantito che le classi Target e Non-Target fossero equamente rappresentate nel dataset finale. Per raggiungere il numero di occorrenze necessarie in Non Target c'è stata la necessità di fondere lo shift temporale e l'aumento del volume, in modo da ottenere un'ulteriore trasformazione. Alla fine di queste operazioni, abbiamo ottenuto 61.695 spettrogrammi per la classe Target e 61.695 spettrogrammi per la classe Non-Target.

5 Addestramento e validazione

Tutti gli esperimenti e l'addestramento dei modelli sono stati eseguiti utilizzando PyTorch, una libreria di deep learning che offre una grande flessibilità e supporto per il calcolo su GPU, facilitando l'implementazione e il training di modelli complessi. In particolare, gli esperimenti sono stati effettuati su due dispositivi aventi tali caratteristiche:

- PC con GPU Nvidia Geforce RTX 3050 con CUDA 10.1, 4GB VRAM, 16GB RAM;
- MacBook Pro con Chip M3 Pro (11-core), GPU 14-core, 18GB RAM.

Per velocizzare le esecuzioni e prevenire l'overfitting, è stata applicata la tecnica dell'**early stopping**. Questo metodo interrompe l'addestramento se la prestazione del modello su un set di validation non migliora dopo un numero definito di epoche consecutive. Questo ci ha permesso di ottimizzare il tempo di addestramento e migliorare la generalizzazione del modello. I vari esperimenti sono stati condotti modificando il valore del parametro di ***Average***, utilizzato per calcolare le metriche di valutazione durante il training e la validation. Le diverse modalità di calcolo utilizzate sono state:

- **Binary**: Utilizzato per il training binario, calcola le metriche considerando la classificazione tra due classi (target e non-target).
- **Micro**: Calcola le metriche globali considerando tutte le classi come se fossero una singola classe.
- **Macro**: Calcola le metriche per ogni classe individualmente e poi fa la media senza tener conto del numero di esempi per classe.
- **Weighted**: Simile a macro, ma pondera la media in base al numero di esempi in ciascuna classe, riducendo l'impatto delle classi sbilanciate.

Sono stati utilizzati tre modelli di deep learning pre-addestrati: GoogleNet, ResNet50 e AlexNet. Tutti inizializzati con pesi pre-addestrati su ImageNet (IMAGENET1K-V1), consentendo un'ottimizzazione più rapida e prestazioni migliori grazie al trasferimento di conoscenza da un dominio simile. I parametri utilizzati per l'addestramento dei modelli sono stati i seguenti:

- Batch Size: 32
- Numero di Epoche: 50 (con early stopping con parametro di 5 epoche)
- Learning Rate: 0.001

Durante l'addestramento, sono state salvate le metriche di prestazione per ogni epoca (Accuracy, Precision, Recall, F1Score), sia per il training che per la validation. Queste metriche sono state utilizzate per creare grafici che ci hanno permesso di confrontare le prestazioni del modello nel tempo. Questo monitoraggio ci ha fornito preziose informazioni sull'efficacia del training e ha aiutato a identificare eventuali problemi di overfitting o underfitting. Sono state adottate due diverse tipologie di classificazione:

1. **Classificazione binaria**: Classificazione tra suoni target e non-target per differenziare suoni generati dall'uomo da suoni animali.
2. **Classificazione multiclasse**: Classificazione tra tutte le classi di suoni, sia utilizzando `one hot encoding` che senza di essa.

5.1 Addestramenti Binari

La classificazione binaria è stata effettuata utilizzando il dataset bilanciato durante la fase di augmentation. I risultati sono stati ottimi, con valori sia nel training che nella validation nell'ordine del 95%.

- Il Risultato migliore è stato ottenuto con **GoogLeNet** impostando il parametro `Average="Macro"`. Con questo addestramento sono stati raggiunti valori superiori al 95% per quanto riguarda precision, recall e f1-score in fase di validazione.
- Invece, il migliore di **AlexNet**, è stato ottenuto con il parametro `Average="Weighted"`, facendo registrare circa il 97,6% per le tre metriche di riferimento.

5.2 Addestramenti Multiclasse

Gli addestramenti multiclasse sono stati gestiti tramite One-Hot-Encoding, sfruttando gli stessi modelli e pesi pre-addestrati sfruttati per l'addestramento binario con le dovute modifiche. Questi training sono stati effettuati su un dataset sbilanciato in quanto i risultati ottenuti con il bilanciato non erano adeguati. Con il dataset bilanciato il modello sembra andare in overfitting generando risultati nel training nell'ordine anche del 70% ma risultati nella validation di molto inferiori al 10%.
Ciò è probabilmente dovuto alla natura controllata dell'augmentation effettuata. Non solo buona parte dei campioni nel dataset bilanciato erano frutto dell'augmentation ma presentavano anche differenze poco significative. Si è così deciso di effettuare diversi esperimenti di training utilizzando il dataset non bilanciato. I risultati ottenuti da questa prova sono stati più incoraggianti con valori intorno al 20-30%. Ne riportiamo due dei migliori di seguito:

- Il risultato migliore registrato con **GoogLeNet**, è stato ottenuto con il parametro Average uguale a **Weighted** e con il metodo del **One Hot Encoding**, con valori del 24,899% per quanto riguarda le tre metriche della validation.
- Il migliore di **ResNet50**, è stato ottenuto con il parametro `Average="Weighted"` e con la tecnica di **One Hot Encoding**, registrando valori intorno al 26% per le tre metriche della validation.
- Il migliore di **AlexNet**, è stato ottenuto con il parametro `Average="Weighted"` e con la tecnica di **One Hot Encoding**, raggiungendo il 27,58% per le tre metriche della validation.

6 Testing

L'ultimo step è stato quello di testare i modelli migliori ottenuti in fase di training. I risultati sono stati in linea con quanto ottenuto nelle relative fasi di validation.

6.1 Test addestramenti Binari

Modello	Average	Precision	Recall	F1
GoogLeNet	Binary	**0.9973**	**0.9974**	**0.9974**
	Weighted	0.9946	0.9945	0.9945
	Macro	0.9899	0.9836	0.9867
	Micro	0.9949	0.9949	0.9949
AlexNet	Weighted	0.9755	0.9625	0.9666

6.2 Test Addestramenti Multiclasse

Dataset	Modello	Encoding	Average	Precision	Recall	F1
Sbilanciato	GoogLeNet	One Hot Encoding	Weighted	0.2326	0.2246	0.2262
			Micro	0.1966	0.1966	0.1966
	AlexNet		Weighted	0.2617	0.2367	0.2419
			Micro	0.2424	**0.2424**	**0.2424**
	ResNet50		Weighted	**0.2651**	0.1630	0.1883
			Micro	0.1857	0.1857	0.1857
Bilanciato	AlexNet	One Hot Encoding	Weighted	0.1376	0.1275	0.1297

7 Conclusioni e problemi aperti

I risultati del lavoro hanno permesso di ottenere un classificatore binario che offrisse ottimi risultati. La classificazione multiclasse ha infine dato risultati con valori delle metriche di precision, recall ed F1-Score intorno al 20-30% portando comunque il modello ad un possibile overfitting seppur non grave quanto negli addestramenti con dataset bilanciato (Con valori delle metriche intorno al 10%). Questi valori sono causati molto probabilmente dalla natura fortemente sbilanciata del data set, il quale non ha beneficiato di un processo controllato di augmentation.
Per ottenere un miglioramento nell'addestramento dei modelli multiclasse sarebbe possibile, ad esempio, modificare il parametro del trimming degli audio. Il trimming a 3 secondi potrebbe aver portato ad un aumento del volume dei dati eccessivo. Un'altra idea potrebbe essere quella di tentare nuove strade nel resampling, modificando la frequenza di ricampionamento. Con l'integrazione di nuovi dati o con un processo di augmentation che diversifichi maggiormente i campioni, le performance potrebbero migliorare ulteriormente.

References

[1] Shuang Yang et al. "Underwater acoustic target recognition based on sub-band concatenated Mel spectrogram and multidomain attention mechanism". In: *Engineering Applications of Artificial Intelligence* (2024).

[2] Boyang Zhang Jared Leitner Sam Thornton Dept. of Electrical and San Diego Computer Engineering University of California. "Audio Recognition using Mel Spectrograms and Convolution Neural Networks". In: (2024).

[3] Mohammad Reza Khalilabadi. "Underwater ship-radiated acoustic noise recognition based on mel-spectrogram and convolutional neural network". In: *International Journal Of Coastal, Offshore And Environmental Engineering (ijcoe)* 8.1 (2023), pp. 10–15.

[4] B Mishachandar and S Vairamuthu. "Diverse ocean noise classification using deep learning". In: *Applied Acoustics* 181 (2021), p. 108141.

[5] Yunqi Zhang and Qunfeng Zeng. "MSLEFC: A low-frequency focused underwater acoustic signal classification and analysis system". In: *Engineering Applications of Artificial Intelligence* 123 (2023), p. 106333.

[6] Shuang Yang et al. "Underwater acoustic target recognition based on sub-band concatenated Mel spectrogram and multidomain attention mechanism". In: *Engineering Applications of Artificial Intelligence* 133 (2024), p. 107983.

[7] BZJLS Thornton. "Audio recognition using mel spectrograms and convolution neural networks". In: (2019).

[8] B Mishachandar and S Vairamuthu. "Diverse ocean noise classification using deep learning". In: *Applied Acoustics* 181 (2021), p. 108141.

[9] Lucas CF Domingos et al. "A survey of underwater acoustic data classification methods using deep learning for shoreline surveillance". In: *Sensors* 22.6 (2022), p. 2181.