UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea in Ingegneria Informatica e dell'Automazione

TESI DI LAUREA

Classificazione Multiclasse delle Vocalizzazioni del Delfino Tramite Machine Learning

Multiclass Classification of Dolphin Vocalizations Through Machine Learning

Relatore:

Prof. David Scaradozzi

Tesi di Laurea di:

Ettore Ricci

Primo Correlatore:

Dott. Francesco di Nardo

Secondo Correlatore:

Dott. Rocco De Marco

Anno Accademico 2023/2024

Università Politecnica delle Marche
Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica e dell'Automazione
Via Brecce Bianche – 60131 Ancona (AN), Italy

INDICE

ELENCO DELLE FIGURE

ELENCO DELLE TABELLE

INTRODUZIONE

Nel panorama attuale della conservazione della biodiversità marina, la coesistenza tra le attività di pesca e la fauna marina spesso presenta conflitti. Principalmente con mammiferi marini come i delfini. I delfini per alimentarsi possono interferire con le attività di pesca, con il rischio di esser catturati (il cosiddetto *by-catch*) e quindi annegare, o di ferirsi gravemente (lesioni o difficoltà nel nuoto) o in maniera letale. I delfini infatti seguono i pescherecci in maniera opportunistica, in quanto reperiscono più facilmente il cibo. Per questo motivo, i delfini vengono visti come nemici/competitori da una parte dei pescatori, che vedono tali animali come fonte di preoccupazione e di minaccia, con ripercussioni di tipo economico dovute alla sottrazione di pesce dalle reti, alla non commerciabilità del pesce predato dai delfini, alla riduzione della performance di pesca, all'interruzione del lavoro in caso di cattura accidentale e ai danni alle attrezzature professionali. Per questo motivo nasce il progetto Life DELFI, progetto cofinanziato dalla Comunità europea nell'ambito del Programma Life, che mette insieme enti di ricerca, università, associazioni ambientaliste e aree marine protette nell'intento comune di sviluppare soluzioni e modelli di gestione sostenibili delle interazioni fra delfini e pesca [1].

Uno dei principali obiettivi del progetto è quello di sviluppare un sistema robotico intelligente capace di captare i suoni marini, identificare le vocalizzazioni dei delfini, e rispondere emettendo segnali acustici per dissuaderli dall'avvicinarsi alle imbarcazioni. Questo sistema si baserebbe su algoritmi avanzati di intelligenza artificiale, progettati per distinguere fischi caratteristici, click di ecolocalizzazione, burst pulse sound e feeding buzz dagli altri suoni subacquei, con l'obiettivo di identificare automaticamente la presenza di delfini. Inoltre, dovrebbe essere in grado di gestire la fase di emissione in tempo reale, o con un ritardo minimo per garantire che i delfini stiano lontani dalle imbarcazioni da pesca. Pertanto, una delle problematiche principali da affrontare è l'identificazione delle vocalizzazioni dei delfini.

I fischi dei delfini sono segnali acustici omnidirezionali a banda stretta con modulazione di frequenza, che rappresentano uno strumento essenziale di comunicazione per i delfini, trasmettendo informazioni sul loro ambiente, le interazioni sociali e l'identità individuale [2]. I click di ecolocalizzazione e i burst pulse sound sono vocalizzazioni impulsive a banda larga (principalmente ultrasonici) che si differenziano per la durata degli intervalli tra i click che è più breve per i burst pulse sound [2]. Anche i feeding buzz sono vocalizzazioni impulsive, ma sono caratterizzate da un contenuto in frequenza molto inferiore rispetto a click di ecolocalizzazione e burst pulse sound (non oltre i 5 kHz).

Molti studi hanno utilizzato il potenziale degli algoritmi di apprendimento automatico nel tentativo di migliorare l'efficienza del monitoraggio dei mammiferi marini e del rilevamento della loro presenza tramite le loro vocalizzazioni. Quasi tutti questi studi [3-6] hanno utilizzato reti neurali convoluzionali (CNN). Attualmente, non esiste un approccio unico in grado di risolvere in modo definitivo il complesso problema dell'identificazione delle vocalizzazioni dei delfini. Sfide come la diversità delle specie e delle loro vocalizzazioni, i limiti tecnologici, gli alti livelli di rumore ambientale e la limitata disponibilità di dati continuano a rappresentare ostacoli significativi per ottenere risultati affidabili e prestazioni elevate nei modelli di rete neurale.

Il presente studio, quindi, è stato progettato con l'obiettivo di testare le prestazioni di una rete neurale convoluzionale (CNN) multiclasse nel classificare quattro differenti vocalizzazioni dei delfini (*Tursiops truncatus*) da registrazioni audio subacquee. Le vocalizzazioni considerate sono: i fischi, i click di ecolocalizzazione, i burst pulse sound e i feeding Buzz. Gli approcci acustici convenzionali per identificare le vocalizzazioni dei delfini sono solitamente basati sull'analisi algoritmica degli spettrogrammi audio. La presente tesi sfrutta un approccio diverso per la preparazione dei dati, utilizzando il filtro Sobel. Il filtro Sobel viene generalmente impiegato nell'elaborazione delle immagini per il rilevamento dei bordi, e per

accentuare i contorni all'interno di un'immagine [7]. In questo lavoro è stato utilizzato per enfatizzare le forme delle diverse vocalizzazioni nello spettrogramma, con l'obiettivo di migliorare la procedura di addestramento della rete neurale e quindi le prestazioni di identificazione delle diverse vocalizzazioni. Per quanto ne sappiamo, questo studio rappresenta il primo tentativo di utilizzare il filtro di Sobel per evidenziare la forma d'onda delle vocalizzazioni del delfino, al fine di renderle più riconoscibile per una rete neurale multiclasse.

CAPITOLO 1

1.1 IL TURSIOPE

I delfini tursiopi (*Tursiops truncatus*), sono tra i più noti tra tutti i cetacei. Il loro areale si estende in quasi tutto il mondo, si trovano nelle acque marine temperate e tropicali, sia vicino alla costa che nelle acque al largo, con una stima di 600.000 esemplari (Figura 1).

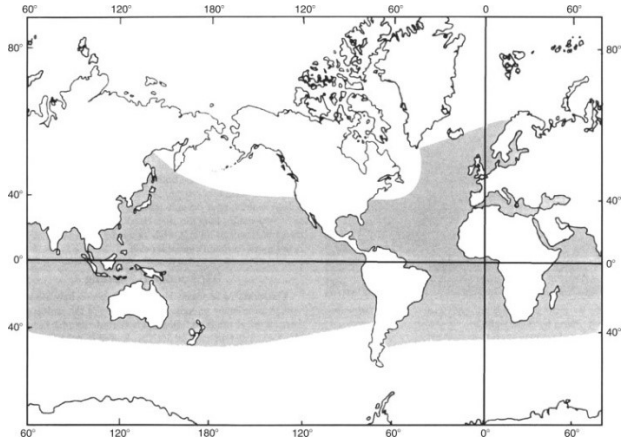

Figura 1 Distribuzione globale delle specie del delfino tursiope comune (Tursiops truncatus), si può notare l'assenza della specie nelle regioni polari dove le temperature dell'acqua sono più fredde.

Mostrano una grande quantità di variazioni morfologiche in base alle aree geografiche. Sono riconoscibili dal loro aspetto generalizzato: un corpo robusto di medie dimensioni, una pinna dorsale moderatamente falcata e una colorazione scura, con una netta demarcazione tra il melone e il breve rostro (Figura 2). Le lunghezze degli adulti variano da circa 2,5 m a circa 3,8 m [8].

Figura 2 Vista laterale di un delfino tursiope maschio adulto

I delfini tursiopi in natura sembrano essere attivi sia di giorno che di notte, alternando periodi di alimentazione, spostamenti, socializzazione e momenti di ozio o riposo. Si nutrono in una grande varietà di modi e habitat, principalmente come individui, ma si verifica anche la formazione cooperativa di banchi di pesci preda. Pesci e/o calamari costituiscono la maggior parte delle diete, sebbene i tursiopi sembrino mostrare una preferenza costante per sciaenidi, sgombridi e mugilidi. La maggior parte delle prede di pesce sono abitanti del fondale, ma anche alcuni abitanti di superficie o pesci pelagici sono rappresentati nelle diete. I pesci che producono rumore costituiscono una parte importante della dieta dei Tursiopi, presumibilmente perché il suono aiuta i delfini a localizzare le prede.[8]

I delfini tursiopi comuni si trovano in genere in gruppi di 2-15 individui, sebbene siano stati segnalati gruppi di oltre 1000 individui. La composizione del gruppo tende a essere dinamica, con sesso, età, condizioni riproduttive, relazioni familiari e storie di affiliazione che sembrano essere i fattori determinanti più importanti. In situazioni di cattività sono state osservate gerarchie di dominanza, con grandi maschi adulti di delfini tursiopi

comuni che dominano tutti gli altri compagni di piscina, mentre le femmine formano una gerarchia meno rigida, con le più grandi dominanti sugli animali più piccoli.

I Tursiopi sono anche ben noti per la loro complessa vita sociale e le loro capacità comunicative. I delfinidi hanno un senso dell'olfatto notevolmente ridotto e solo una visibilità limitata nel loro ambiente. Quindi, il canale acustico è quello principale disponibile per le interazioni sociali. [9] Sembrano essere gli unici mammiferi non umani conosciuti che utilizzano segnali appresi come etichette specifiche individuali per diversi compagni sociali nel loro sistema di comunicazione naturale; come se si "chiamassero per nome". Sono state riportate prove di capacità empatiche in questi mammiferi come osservazioni di aiuti mirati, mordendo le linee degli arpioni o tirandole fuori quando sono intrappolati nelle reti da pesca, sostenendo gli individui malati vicino alla superficie per evitare che anneghino, rimanendo vicino a una femmina in travaglio, interponendosi tra la barca di un cacciatore e un conspecifico ferito, o persino capovolgendo la barca. [10]

1.2VOCALIZZAZIONI DEL TURSIOPE

I delfini tursiopi utilizzano diversi tipi di segnali acustici: click di ecolocalizzazione, multiple burst pulse sound, fischi modulati in frequenza e i feeding buzz. I click possono essere definiti come esplosioni di suoni a banda larga con un fronte istantaneo nitido e uno spettro di frequenza continuo, questi click sono fondamentali per la navigazione e il foraggiamento, consentendo ai delfini di costruire immagini uditive dettagliate dell'ambiente circostante. Una serie di click emessi consecutivamente vengono definiti click train. La durata di un click train può variare da 0,001 secondi a diversi secondi di durata, per quanto riguarda il contenuto in frequenza gli studi iniziali hanno riportato variazioni da 20,00 a 120,00 kHz con frequenze occasionali fino a 170 kHz. Spesso viene utilizzata la frequenza di ripetizione, o intervallo tra i click, per descrivere i treni di click. I tassi di ripetizione possono variare da cinque a diverse centinaia di click al secondo. Ad esempio, durante il rilevamento di un bersaglio, le percentuali di ripetizione dei click iniziavano lentamente (10 click/s) per poi aumentare prima del raggiungimento del bersaglio (190 click/s) [11]. I multiple burst pulse sound sono definiti come un treno di click in cui la frequenza di ripetizione aumenta e poi diminuisce ciclicamente. I click che compongono i burst pulse sound sono strutturati in maniera irregolare e contengono bruschi cambiamenti nella loro composizione in frequenza. Svolgono un ruolo sociale, spesso associato a interazioni aggressive, come quelle osservate durante la depredazione, e possono essere utilizzate per risolvere conflitti di rango e ridurre la competizione tra i membri del gruppo. I burst pulse sound possono essere descritti come suoni a banda larga di breve durata, possono essere separati in burst pulse sound corti, di durata inferiore a 0,2 secondi, e lunghi per quelli di durata superiore. I burst pulse sound corti hanno una frequenza minima media di 4,12 kHz + 3,77, ed una frequenza massima di 25,97 kHz +-

8,57; formati da circa 287 +- 64 click e un ICI 0,004 + 0,001. In contesti aggressivi, gli impulsi burst hanno mostrato componenti di frequenza principali tra 60,0 e 150,0 kHz. Sia i click di ecolocalizzazione che gli impulsi burst sono composti da gruppi di singoli impulsi, ma è la velocità con cui tali click vengono emessi che cambia la loro funzione, i secondi hanno un intervallo tra i click 10 volte più breve [11]. I fischi sono suoni a banda stretta modulati in frequenza utilizzati prevalentemente per la comunicazione sociale [12]. I delfini producono una grande varietà di fischi, tra cui "fischi caratteristici" ampiamente stereotipati che sono individualmente specifici e sembrano essere usati per comunicare identità, posizione e possibilmente stato emotivo [8]. Queste vocalizzazioni sono caratterizzate da frequenze tipicamente comprese tra 1 e 25 kHz e durate variabili da 0,1 a pochi secondi [12]. I feeding buzz corrispondono a sequenze di click ad alta frequenza di ripetizioni che iniziano dopo click regolari, con una brusca diminuzione dell'ICI (intervallo inter-click). Hanno una durata media di circa 1 s e consistono in 100-600 click. I feeding buzz vengono emessi durante l'immersione, probabilmente quando si insegue una preda [13]. Inoltre, i delfini hanno anche dei segnali di soccorso, un sottoinsieme del loro comportamento biologico, che includono l'abbaiare, l'ululare e il trillo. I delfini emettono questi segnali quando provano emozioni come rabbia, paura, frustrazione e angoscia. Sebbene simili ai segnali di ecolocalizzazione, i segnali di soccorso sono caratterizzati da intervalli di impulsi più brevi e intensità inferiore. A causa della loro complessità, questi segnali sono difficili da caratterizzare [14].

Le diverse vocalizzazioni dei delfini possono essere analizzate visivamente attraverso il loro spettrogramma, grazie al fatto che per ogni vocalizzazione è possibile evidenziare delle forme d’onda molto riconoscibili. Per i fischi, ad esempio, dallo spettrogramma è possibile individuare una forma d’onda “sinusoidale” continua, per quanto riguarda click,

burst pulse sound e feeding buzz invece, essendo caratterizzati da insiemi di impulsi ad alta frequenza, nei loro spettrogrammi è possibile individuare onde rapide e strette con intervalli di tempo variabili tra due successive.

1.3 INTERAZIONI CON LE ATTIVITÀ DI PESCA

La pesca con le reti da traino è una delle attività più caratterizzate per la cattura di specie considerate accidentali. Comporta la cattura di molte specie, che spesso vengono rigettate in mare prive di vita, oltre alle specie bersaglio. Con il termine *bycatch* si indica quella parte della cattura che non rappresenta il target commerciale dell'attività di pesca nelle quale sono comprese le specie marine protette, come mammiferi, rettili, squali e gli uccelli marini. Per quanto riguarda l'impatto sui cetacei, quelli che maggiormente interagiscono con le attività di pesca sono i tursiopi, i quali sono spesso osservati nei pressi dei pescherecci che operano con reti da traino. Si ritiene che la natura dell'interazione catacei-traino pelagico veda la combinazione di varia attività, tra le quali si evidenziano la depredazione attuata verosimilmente all'esterno delle reti, lo sfruttamento della presenza della rete che interagendo con il fondo richiama pesci più facilmente cacciabili e l'alimentazione sui pesci scartati durante le fasi di cernita a bordo o fuggiti dalle maglie della rete [15]. Sono state segnalate catture accidentali di piccole quantità di *T. truncatus* per diverse attività di pesca, tra cui la pesca con reti a circuizione di tonni, sardine e acciughe. In alcuni casi, i delfini sono stati uccisi dai pescatori per impedire danni alle loro attrezzature da pesca o il furto del pescato o dell'esca. Negli Stati Uniti, l'impigliamento o l'ingestione di attrezzature da pesca ricreativa sta causando un numero crescente di mortalità di comuni delfini tursiopi [8].

L’impatto della pesca eccessiva sulle prede dei delfini può provocare un declino dei cetacei dovuto al depauperamento delle loro prede dall’altro lato il danno percepito dai pescatori in termini di depredazione è spesso superiore a quello oggettivamente quantificabile [15].

CAPITOLO 2

2.1 LE RETI NEURALI ARTIFICIALI

La rete neurale artificiale (ANN) è un modello che cerca di simulare l'elaborazione della rete neurale del cervello umano. La rete neurale è composta da un gran numero di nodi (o neuroni) collegati tra loro. Ogni nodo rappresenta una funzione di output specifica, chiamata funzione di attivazione. La connessione tra ogni due nodi rappresenta un peso per il segnale che passa attraverso la connessione, che è equivalente alla memoria della rete neurale artificiale. L'output della rete varierà a seconda di come la rete è connessa, del valore del peso e della funzione di attivazione. Tuttavia, la rete stessa è solitamente un'approssimazione di un qualche tipo di algoritmo o funzione in natura, oppure può essere un'espressione di una strategia logica. Il tipo di unità di elaborazione nella rete è diviso in tre categorie: unità di input, unità di output e unità nascosta (Figura3). L'unità di input accetta segnali e dati dal mondo esterno. L'unità di output realizza l'output del risultato di elaborazione del sistema. L'unità nascosta è un'unità che si trova tra le unità di input e output e non può essere osservata all'esterno del sistema. [16]

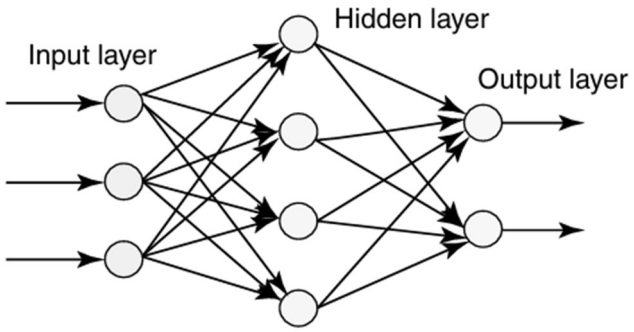

Figura 3 Rete neurale artificiale (ANN)

Le reti neurali profonde, note come *Deep Neural Network* (DNN) sono modelli computazionali costituiti da molte unità di elaborazione semplici che lavorano in parallelo e sono disposte in strati interconnessi. Le reti neurali semplici costituite da uno strato di input e uno di output; quando vengono impilati più strati, sono chiamate profonde. Una DNN impara a svolgere particolari attività tramite l'addestramento, durante il quale viene appresa la forza delle connessioni tra le unità. Successivamente, la DNN addestrata viene utilizzata per svolgere la stessa attività su nuovi input.[17]

2.2 PARADIGMI DI APPRENDIMENTO

I paradigmi di apprendimento nelle reti neurali possono essere classificati in tre tipi distinti (Figura4). Si tratta dell'apprendimento supervisionato, dell'apprendimento non supervisionato e dell'apprendimento per rinforzo.[18]

Figura 4 Paradigmi di apprendimento

2.2.1 APPRENDIMENTO SUPERVISIONATO

L'apprendimento supervisionato è l'attività di apprendimento automatico di una funzione che mappa un input su un output in base a coppie input-output di esempio, ovvero l'input e l'output desiderati sono predefiniti. L'algoritmo ottiene il set di input e i risultati appropriati corrispondenti. L'algoritmo confronta i risultati effettivi con i risultati appropriati per ottenere l'output. Successivamente, il modello viene aggiornato di conseguenza [19]. Viene dedotta una funzione dai dati di addestramento etichettati costituiti da un insieme di esempi di addestramento. Gli algoritmi di apprendimento automatico supervisionato sono quegli algoritmi che necessitano di assistenza esterna. L'apprendimento supervisionato è la tecnica più comune per l'addestramento per reti neutre e alberi decisionali. Entrambi dipendono dalle informazioni fornite dalla classificazione predeterminata [20]. Il set di dati di input è suddiviso in set di dati di training e set di dati di test. Il set di dati dell'addestramento ha una variabile di output che deve essere prevista o classificata. Tutti gli algoritmi apprendono alcuni tipi di modelli dal set di dati di addestramento e li applicano al set di dati di test per la previsione o la classificazione (Figura 5). [21]

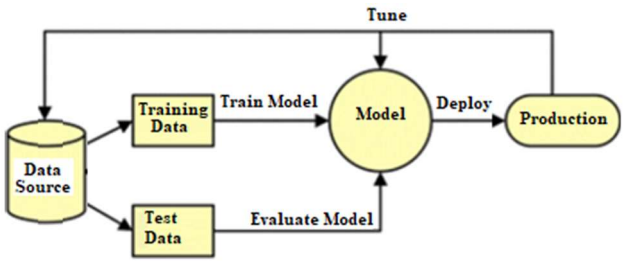

Figura 5 Workflow apprendimento supervisionato

2.2.2 APPRENDIMENTO NON SUPERVISIONATO

Nell'apprendimento non supervisionato (o auto-organizzazione), un'unità (di output) viene addestrata a rispondere a cluster di modelli all'interno dell'input. In questo paradigma, si suppone che il sistema scopra le caratteristiche statisticamente salienti della popolazione in ingresso. A differenza del paradigma dell'apprendimento supervisionato, non esiste un insieme a priori di categorie in cui i modelli devono essere classificati; piuttosto, il sistema deve sviluppare la propria rappresentazione degli stimoli in ingresso [18]. La principale differenza tra l'apprendimento supervisionato e l'apprendimento non supervisionato è che l'apprendimento supervisionato richiede la mappatura dall'input all'output essenziale. D'altra parte, l'apprendimento non supervisionato non cerca di produrre output nel feedback dell'input specifico, ma rivela modelli nei dati [19]. Viene utilizzato principalmente per il clustering e la riduzione delle funzionalità [21].

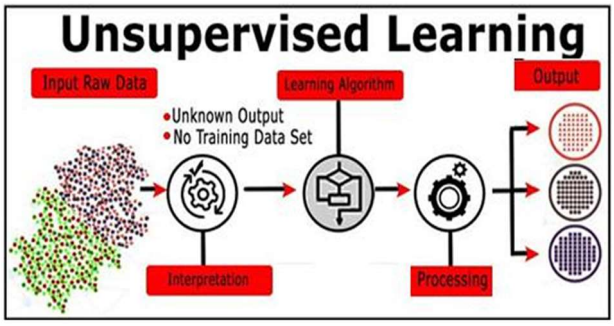

Figura 6 Workflow apprendimento non supervisionato

2.2.3 APPRENDIMENTO PER RINFORZO

L'apprendimento per rinforzo è un'area dell'apprendimento automatico che si occupa del modo in cui gli agenti software dovrebbero intraprendere azioni in un ambiente al fine di massimizzare una certa nozione di ricompensa cumulativa [21] L'apprendimento per rinforzo consiste nell'imparare cosa fare, come mappare le situazioni in azioni, in modo da massimizzare un segnale di ricompensa numerico. Al learner non viene detto quali azioni intraprendere, come nella maggior parte delle forme di apprendimento automatico, ma deve invece scoprire quali azioni producono la massima ricompensa provandole. Nei casi più interessanti e impegnativi, le azioni possono influenzare non solo la ricompensa immediata, ma anche la situazione successiva e, attraverso di essa, tutte le ricompense successive. Queste due caratteristiche, la ricerca per tentativi ed errori e la ricompensa ritardata sono le due caratteristiche distintive più importanti dell'apprendimento per rinforzo.[18]

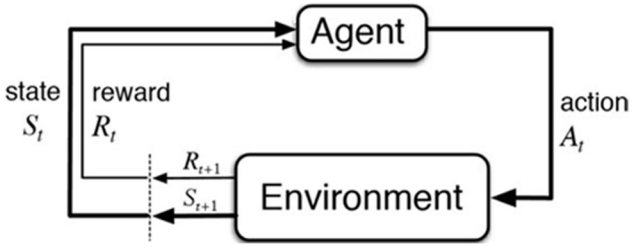

Figura 7 Workflow apprendimento per rinforzo

2.3 RETI NEURALI CONVOLUZIONALI

La CNN (*Convolutional Neural Network*) è una rete neurale profonda in grado di estrarre caratteristiche dai dati con strutture di convoluzione. A differenza dei metodi tradizionali di estrazione delle caratteristiche, la CNN non ha bisogno di estrarre le caratteristiche manualmente. L'architettura della CNN si ispira alla percezione visiva. Ha una struttura basata su strati convoluzionali che le consente di catturare informazioni visive attraverso una serie di kernel o filtri. I *kernel* (o filtri) dei neuroni biologici fungono da recettori per una varietà di caratteristiche e le funzioni di attivazione simulano la trasmissione del segnale solo quando l'input supera una soglia specifica. Le funzioni di perdita e gli ottimizzatori sono utilizzate per insegnare all'intero sistema CNN a capire cosa ci aspettiamo. La CNN possiede molti vantaggi: 1) connessioni locali: ogni neurone non è più connesso a tutti i neuroni dello strato precedente, ma solo a un piccolo numero di neuroni, il che è efficace nel ridurre i parametri e accelerare la convergenza. 2) Condivisione del peso: un gruppo di connessioni può condividere gli stessi pesi, riducendo ulteriormente i parametri. 3) Riduzione delle dimensioni di *downsampling*: un livello di pooling sfrutta il principio della correlazione locale dell'immagine per sotto campionare un'immagine, che può ridurre la quantità di dati conservando le informazioni utili. Può anche ridurre il numero di parametri rimuovendo le funzioni banali. Queste tre caratteristiche interessanti rendono la CNN uno degli algoritmi più rappresentativi nel campo del deep learning. [22]

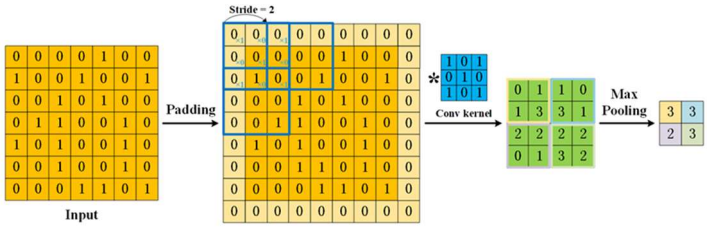

Figura 8 Procedura di una CNN 2D

Le CNN sono composte da diversi tipi di strati. Quando questi livelli sono impilati, si è formata un'architettura CNN.

Strato convoluzionale: come suggerisce il nome, lo strato convoluzionale svolge un ruolo fondamentale nel funzionamento delle CNN. I parametri dei livelli si concentrano sull'uso di kernel apprendibili. Questi nuclei sono solitamente piccoli in termini di dimensionalità spaziale, ma si diffondono lungo l'intera profondità dell'input. Quando i dati raggiungono un livello convoluzionale, il livello convolge ogni filtro attraverso la dimensionalità spaziale dell'input per produrre una *feature map* 2D. A mano a mano che passiamo attraverso l'input, il prodotto scalare viene calcolato per ogni valore in quel kernel. Un ruolo fondamentale lo ricopre la funzione di attivazione *ReLu*, che sta per unità di rivestimento rettificato ed è una funzione di attivazione non lineare ampiamente utilizzata nelle reti neurali. Il vantaggio dell'utilizzo della funzione *ReLU* è che tutti i neuroni non vengono attivati contemporaneamente. Può essere definito matematicamente come:

$$f(x) = \max(0, x) \tag{1}$$

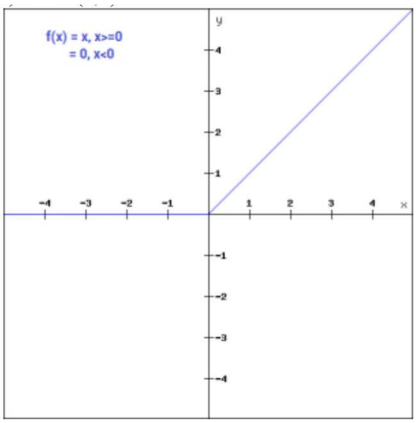

Figura 9 Grafico rappresentativo funzione di attivazione ReLu

Strato di pooling: i livelli di pooling mirano a ridurre gradualmente la dimensionalità della rappresentazione e quindi a ridurre ulteriormente il numero di parametri e la complessità computazionale del modello. Il livello di pooling opera su ogni *feature map* nell'input e scala la sua dimensionalità utilizzando la funzione "MAX".

Strato di *flatten*: dopo la fase di convoluzione e di pooling, la rete convoluzionale passa attraverso lo strato di *flatten*, il cui ruolo è quello di trasformare le feature map bidimensionali in vettori unidimensionali, poiché gli strati completamente connessi richiedono un input di tipo vettoriale.

Strato completamente connesso: lo strato completamente connesso contiene neuroni che sono direttamente collegati ai neuroni nei due strati adiacenti, senza essere collegati ad alcun strato al loro interno. Questo è analogo al modo in cui i neuroni sono disposti nelle forme tradizionali di ANN [23].

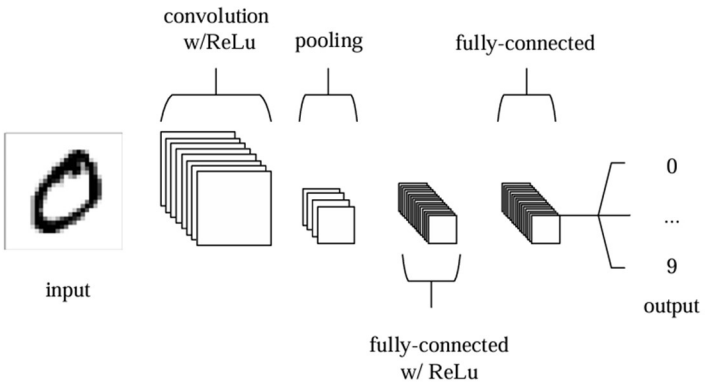

Figura 10 Architettura di una CNN con 5 strati

Nei problemi di calssificazione spesso lo strato di uscita, che fornisce la probabilità di appartenenza a ciasciuna classe, utilizza una funzione *softmax* per produrre una distribuzione di probabilità sulle classi possibili. La funzione *Softmax* è una combinazione di più funzioni *sigmoide*. Una funzione sigmoidea ritorna valori nell'intervallo da 0 a 1, questi possono essere trattati come probabilità dei punti dati di una particolare classe. La funzione *Softmax* a differenza delle funzioni sigmoidali che sono utilizzati per la classificazione binaria, possono essere utilizzati per problemi di classificazione multiclasse. La funzione, per ogni punto restituisce la probabilità di appartenenza a tutte le singole classi. Può essere espressa come: [24]

$$\sigma(x)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad for\ j = 1, \dots, K. \tag{2}$$

2.4 TECNICHE DI ADDESTRAMENTO

Uno degli approcci più utilizzati per la fase di addestramento è la tecnica *K-fold Cross-Validation* (KCV), che permette sia di regolare gli iperparametri sia di stimare l'errore di generalizzazione del classificatore. Il KCV consiste nel suddividere un set di dati in k sottoinsiemi; Quindi, in modo iterativo, alcuni di essi vengono utilizzati per apprendere il modello, mentre gli altri vengono sfruttati per valutarne le prestazioni [25]. L'utilizzo della KCV è utile per evitare che la rete si adatti eccessivamente ai dati di addestramento, fenomeno noto come *overfitting*. A causa dell'esistenza dell'*overfitting*, il modello funziona perfettamente sul set di addestramento, mentre si adatta male sul set di test. Ciò è dovuto al fatto che il modello sovradimensionato ha difficoltà a gestire parti delle informazioni nel set di test, che possono essere diverse da quelle nel set di addestramento [26] (Figura 11) [27].

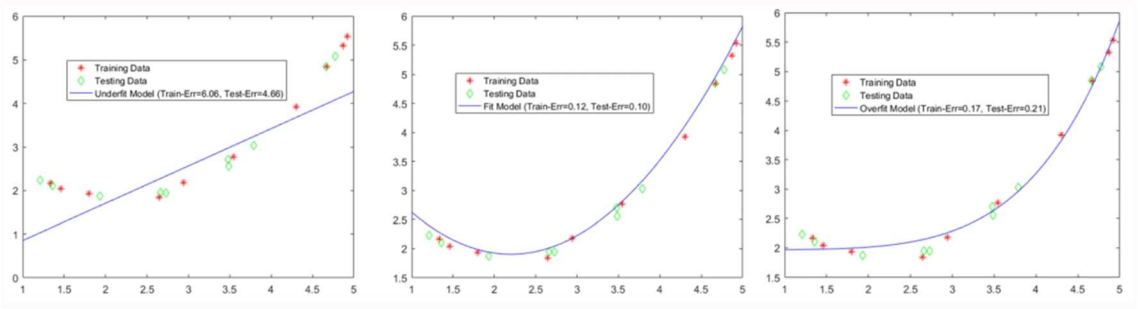

Figura 11 Esempio dei problemi di underfitting, fitting normale e overfitting.

L'applicazione della tecnica KCV risolve anche la problematica di ottenere dei risultati influenzati dalla scelta casuale dei dati nel dataset di addestramento e test. Dei risultati simili sarebbero inaffidabili e non rifletterebbero le funzionalità del modello.

2.5 METRICHE DI VALUTAZIONE DEL MODELLO

Dopo la fase di addestramento è prevista una fase di test, che nel processo di sviluppo di una rete neurale convoluzionale è fondamentale per permettere di misurare le prestazioni del modello, in particolare la sua capacità nel generalizzare le previsioni effettuate su dati indipendenti da quelli utilizzati nella fase di addestramento. La rete utilizza quello che ha appreso dai dati di addestramento sui dati di test, generando delle previsioni che successivamente vengono confrontate con le etichette effettive. Le previsioni effettuate dalla rete determinano la *Confusion Matrix*, una matrice di dimensione NxN, dove N è il numero delle classi, formata da quattro parametri, definiti in un contesto multi-classe come:

- True Positive: casi in cui un campione è correttamente classificato come appartenente alla classe C_i.
- True Negative: casi in cui un campione che non appartiene alla classe C_i è correttamente classificato come non appartenente a quella classe.
- False Positive: casi in cui un campione che non appartiene alla classe C_i è erroneamente classificato come appartenente a quella classe.
- False Negative: casi in cui un campione che appartiene alla classe C_i è erroneamente classificato come appartenente ad una classe diversa da C_i.

I modelli sono stati valutati secondo le misure più comunemente utilizzate ovvero *Precision, Recall, Accuracy* e *F1-Score*.

La *Precision* determina la frazione di valori positivi recuperati che sono corretti, è definita come:

$$Precision = \frac{TP}{TP+TF} \quad (3)$$

Il *Recall*, determinando la frazione di valori positivi che è stata classificata come tale, è definito come:

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

L*'Accuracy*, determina la capacità del modello di classificare in maniera esatta i dati, è definita come:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

L'*F1-Score*, combinando *Precision* e *Recall* otteniamo questa metrica che non è altro che la media armonica di questi due valori, definito come:

$$F1score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (6)$$

CAPITOLO 3

MATERIALI E METODI

3.1 RACCOLTA DATI

La raccolta dei dati è un processo molto importante per garantire una corretta implementazione in ogni fase di sviluppo della ricerca. Diverse sessioni di registrazioni sono state effettuate negli ultimi anni, in differenti ambienti e condizioni operative, e queste hanno permesso di creare un dataset strutturato e performante per l'addestramento della rete neurale convoluzionale.

3.1.1 ACQUISIZIONE E REGISTRAZIONE

La raccolta del dataset è stata eseguita in collaborazione con l'IRBIM (*Istituto per le Risorse Biologiche e le Biotecnologie Marine*), una divisione del CNR (*Consiglio Nazionale delle Ricerche*). Tale attività è parte integrante del progetto Life DELFI, che ha destinato risorse importanti per la creazione di una grande banca dati a supporto degli studi scientifici. Le registrazioni sono state effettuate in un contesto controllato, ovvero presso il delfinario Oltremare di Riccione (Figura 12).

Figura 12 Laguna dei Delfini al delfinario di Oltremare

Tale approccio agevola l'acquisizione di un elevato numero di vocalizzazioni dei tursiopi. Il delfinario di Riccione ospita sette esemplari di tursiope, numero corrispondente alla media dei componenti di un gruppo sociale.

La registrazione è effettuata tramite un registratore subacqueo autonomo UREC384K capace di campionare in modo continuativo equipaggiato con un idrofono Sensor Technology SQ26-08 (Figura 13), in questo lavoro I segnali sono stati acquisiti alla frequenza di campionamento di 192 kHz e memorizzati come file wav a 16 bit della durata di 5 minuti ciascuno. Le sue dimensioni sono di circa 35 cm di lunghezza e 13 cm di diametro. L'apparecchio, alimentato da tre batterie alcaline di tipo D, possiede un'autonomia superiore alle 72 ore, ed è capace di memorizzare registrazioni della durata di cinque minuti nel formato WAV (*Waveform Audio File Format*). L'UREC382K viene posizionato nella vasca di maggiori dimensioni della "Laguna dei Delfini" alloggiato all'interno di una botola appositamente progettata per evitare interazioni fisiche dirette con i delfini, garantendo così la non intrusività del metodo di raccolta dati acustici.

Figura 13 Registratore subacqueo utilizzato durante le prove

3.1.2 ETICHETTATURA

Una volta effettuate le registrazioni, gli audio sono stati sottoposti a degli operatori PAM (*Passive Acoustic Monitoring*) qualificati dal CNR. Attraverso l'utilizzo del software *Audacity*, gli operatori hanno eseguito un'attenta osservazione delle tracce audio per identificare e classificare le vocalizzazioni. I file vengono successivamente contrassegnati con delle etichette specifiche che indicano la presenza di fischi, click di ecolocalizzazione, burst pulse sound, feeding buzz oppure interferenze rumorose. Tale procedura ha l'obiettivo di identificare la *Ground Truth*, cioè il set di dati reali essenziali per l'addestramento e la validazione dei modelli. *Ground Truth* può essere visto come un termine concettuale legato alla conoscenza della verità riguardante una domanda specifica. È il risultato ideale atteso [28]. Durante il processo di inferenza, un modello di classificazione genera previsioni di etichette che possono essere confrontate con le etichette della *Ground Truth* con il fine di valutare precisione ed efficienza del modello stesso.

3.2 PRE-ELABORAZIONE DATI

Il dataset utilizzato corrisponde alle registrazioni effettuate al delfinario di Oltremare, il quale presenta un vantaggio significativo poiché le registrazioni sono risultati di stimoli sottoposti ai cetacei, indotti ad emettere delle vocalizzazioni attraverso attività come fasi di nutrizione o di gioco.

Partendo da 303 registrazioni in formato WAV, ognuna con le rispettive etichette, il processo di pre-elaborazione dei dati è utile per generare immagini da utilizzare nelle fasi di addestramento e test della rete neurale convoluzionale. Le registrazioni vengono suddivise in blocchi dalla durata di 0,8 secondi con un overlap di 0,4 secondi e vengono generate delle immagini di 300x150 pixel. È stato scelto di generare immagini in scala di grigi poiché è preferita per l'addestramento e l'inferenza della rete a causa del minore onere computazionale [29]. Viene anche effettuata la normalizzazione del segnale per garantire livelli di ampiezza uniformemente scalati.

3.2.1 GENERAZIONE SPETTROGRAMMI

Per ogni blocco di 0,8 secondi viene generato uno spettrogramma frammentando il segnale in 512 campioni con un overlap di 256 campioni. L'utilizzo di tale overlap e della finestra di *Hann* serve ad attenuare un segnale, moltiplicando una finestra per un frammento estratto dal segnale, riducendo l'importanza dei bordi [30]. Per trasformare il segnale dal dominio del tempo al dominio della frequenza è stata utilizzata la FFT (trasformata di Fourier veloce) impostando una frequenza minima e massina delle vocalizzazioni pari a 3kHz e 96kHz.

La FFT calcola in modo efficiente la trasformata discreta di Fourier (DFT), una mappatura di una sequenza complessa di lunghezza n nel suo spettro complesso di lunghezza n. La riduzione del calcolo si ottiene notando che solo la metà dei valori complessi deve essere calcolata poiché gli altri sono poi noti dalla simmetria coniugata complessa [31]. In questo lavoro gli spettrogrammi sono stati generati utilizzando la libreria *Numpy*; successivamente sono state convertite le ampiezze delle frequenze in decibel, ed infine i dati sono stati organizzati in una matrice che mostra come il segnale è distribuito tra le diverse frequenze nel tempo.

3.2.2 FILTRO SOBEL

L'operatore di *Sobel* esegue una misurazione del gradiente spaziale 2D sulle immagini. Il trasferimento di un array di pixel 2D in un set di dati statisticamente non correlati migliora la rimozione dei dati ridondanti, di conseguenza, è necessaria una riduzione della quantità di dati per rappresentare un'immagine digitale. Il rivelatore di bordi *Sobel* utilizza una coppia di maschere di convoluzione, una che stima il gradiente nella direzione x e l'altra che stima il gradiente nella direzione y. Il rivelatore *Sobel* è incredibilmente sensibile al rumore nelle immagini, le evidenzia efficacemente come bordi. In questo lavoro il filtro *Sobel* è stato applicato con una matrice convoluzionale 7x7 lungo la direzione verticale per le vocalizzazioni dei fischi e per i rumori, mentre è stato utilizzato lungo la direzione orizzontale per click, burst pulse sound e feeding buzz. Successivamente è rappresentata l’applicazione del filtro sulle diverse vocalizzazioni (Figura 14 – 15 – 16 – 17).

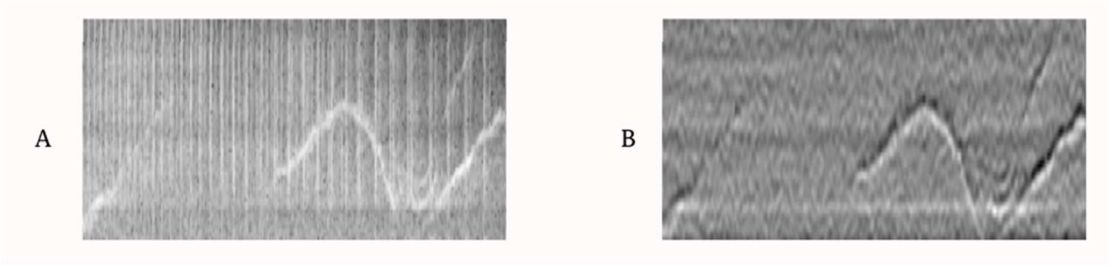

Figura 14 Immagine A: Spettrogramma di un fischio di delfino con presenza di rumore verticale. Immagine B: Stesso spettrogramma dell'immagine A ma con filtraggio sobel verticale.

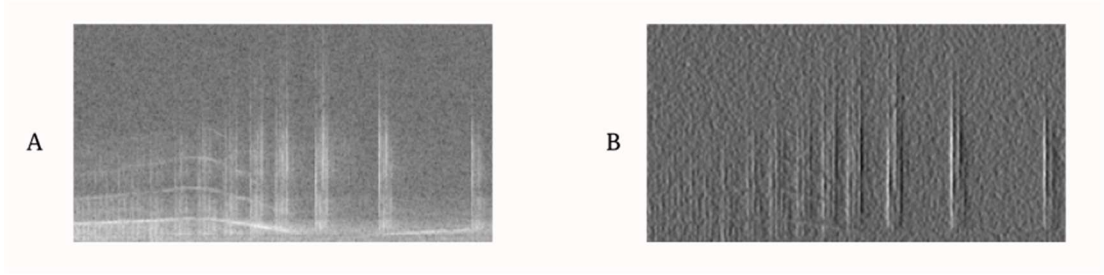

Figura 15 Immagine A: Spettrogramma di un click train di un delfino con presenza di rumore orizzontale. Immagine B: Stesso spettrogramma dell'immagine A ma con filtraggio sobel orizzontale.

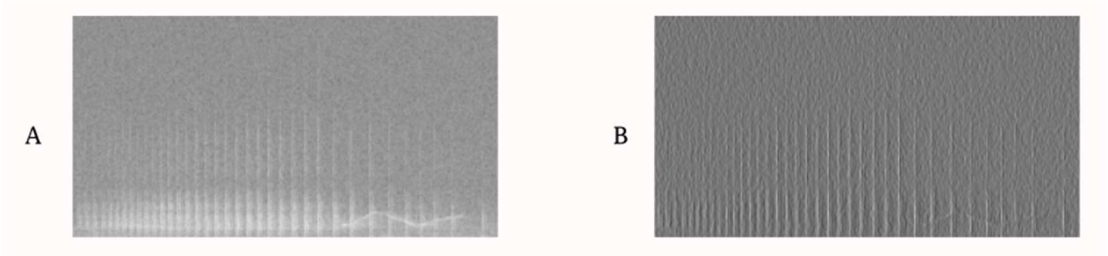

Figura 16 Immagine A: Spettrogramma di un burst pulse sound di delfino con presenza di rumore orizzontale. Immagine B: Stesso spettrogramma dell'immagine A ma con filtraggio sobel orizzontale

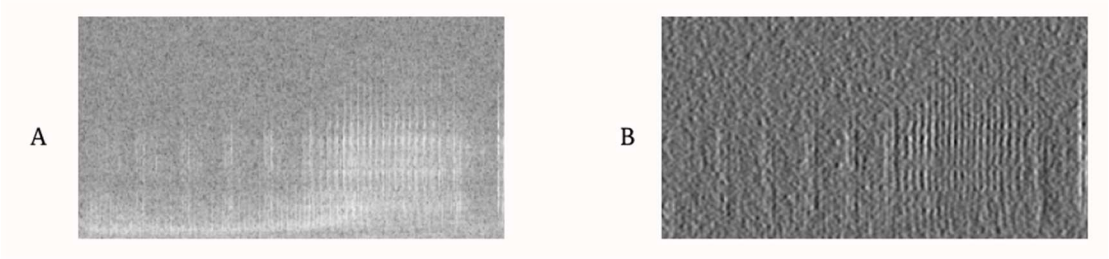

Figura 17 Immagine A: Spettrogramma di un feeding buzz di delfino con presenza di rumore orizzontale. Immagine B: Stesso spettrogramma dell'immagine A ma con filtraggio sobel orizzontale.

3.2.3 GENERAZIONE IMMAGINI

Le immagini che sono state utilizzate durante la fase di addestramento della rete sono state generate utilizzando le etichette delle vocalizzazioni corrispondenti alle registrazioni. Tutti i segnali con durata superiore ai 0,2 secondi sono presentati, non scalati e centrati all'interno di una finestra temporale fissa. Se la durata del segnale è maggiore di 0,2 secondi e la dimensione della finestra temporale è minore di 0,8 secondi, la selezione viene estesa di 0,3 secondi prima e di 0,3 secondi dopo il segnale. Vengono così generate tre immagini raffiguranti la stessa vocalizzazione divise in tre posizioni (Figura 18).

Figura 18 Immagini rappresentanti un click train di durata inferiore a 0,8 secondi esteso e sezionato in tre diverse posizioni

Per i segnali con durata superiore agli 0,8 secondi, vengono generate più immagini per coprire l'intera durata della vocalizzazione. La finestra di riferimento scorre lungo la vocalizzazione con un overlap specifico, 0,4 secondi. Viene generata una immagine per ogni scorrimento (Figura 19).

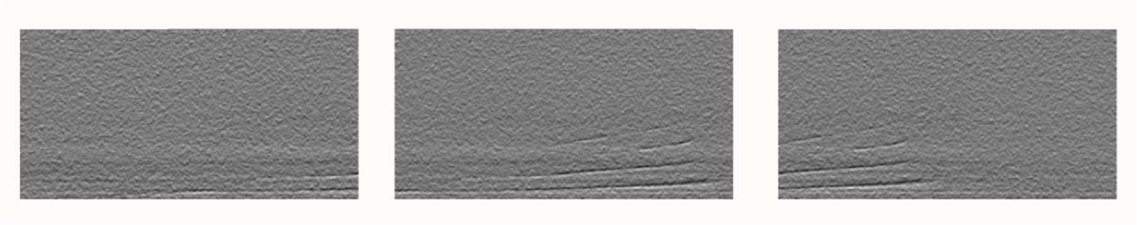

Figura 19 Immagini rappresentanti un fischio di delfino dalla durata superiore di 0,8 secondi, con un avanzamento di 0,5 secondi

Queste metodologie utilizzate per la generazione delle immagini ci garantiscono un maggior numero di immagini all'interno del nostro dataset. Inoltre, il diverso posizionamento delle varie vocalizzazioni renderà l'apprendimento della rete più performante. Per tutte le finestre etichettate sono stati generati spettrogrammi e immagini ottenute applicando il filtro *Sobel* allo stesso spettrogramma, quest'ultime verranno utilizzate per la fase di addestramento.

3.3 PREPARAZIONE DEL DATASET

Per la fase di addestramento vengono utilizzate le immagini generate tramite l'etichettatura delle registrazioni, in totale il dataset utilizzato contiene 3000 immagini rappresentati fischi (classe 1), 2000 rappresentanti click (classe 2), 1400 rappresentanti pulse busrt sound (classe 3), 420 per i feeding buzz (classe 4) e infine 3000 per i rumori (classe 0).

In questo lavoro è stata applicata la tecnica *10-fold Cross-Validation* (**vedi capitolo 2.4**), quindi il 90% del dataset (k-1) sarà utilizzato per la fase di addestramento, una porzione considerevole dell'intero dataset per assicurarci che la rete neurale apprenda al meglio le caratteristiche fondamentali e i legami intrinsechi tra i dati. La restante parte, ovvero il

10% del dataset viene utilizzato per la fase di test, necessaria per confermare che la rete sia stata addestrata correttamente. Le immagini che fanno parte del 90% del dataset utilizzate per la fase di addestramento non vengono mai utilizzate per la fase di test del modello stesso.

3.4 ARCHITETTURA CNN

La rete neurale convoluzionale utilizzata ha una struttura sequenziale, creata per elaborare immagini di dimensione 300x150 pixel in scala di grigi. Tale struttura è formata da tre strati convoluzionali, ognuno seguito da uno strato di Pooling, successivamente uno strato di *Flatten* e in fine uno strato completamente connesso utilizzato per la classificazione. Lo schema della rete è rappresentato nella Figura 20.

Il primo strato della nostra rete è uno strato convoluzionale che impiega 32 filtri con un *Kernel* 3x3, ed utilizza la funzione di attivazione non lineare *ReLu* che facilità l'identificazione di caratteristiche dell'immagine (**vedi capitolo 2**). Successivamente è stato applicato uno strato di Pooling con funzione di attivazione *Max Pooling* con finestra 2x2, il cui obiettivo è quello di ridurre la dimensione della *feature map*, mantenendo solo le informazioni più significative, cercando così di evitare l'*overfitting*. Questi due strati vengono ripetuti per due volte con una variazione agli strati convoluzionali: il secondo strato utilizza 64 filtri per il kernel mentre il terzo 128. Dopo questi tre strati convoluzionali seguiti dai tre strati di pooling, è presenta uno strato di *Flatten* che ha la funzione di trasformare le *feature map* bidimensionali risultanti dagli strati convoluzionali in un vettore unidimensionale. Questo vettore verrà processato da uno strato denso, o strato completamente connesso, composto da 128 unità con funzione di

attivazione *ReLu*. Infine, è presente un ulteriore strato denso, ma con una funzione di attivazione *Softmax*, configurata per una classificazione multi-classe, l'output sarà una distribuzione di probabilità di appartenenza ad una delle cinque classi, rumore, fischio, click, pulse burst sound e feeding buzz. L'architettura è concepita per estrarre le caratteristiche dall'input visivo, ed attraverso l'analisi fornita da ogni strato convoluzionale prende decisioni tramite gli strati completamente connessi.

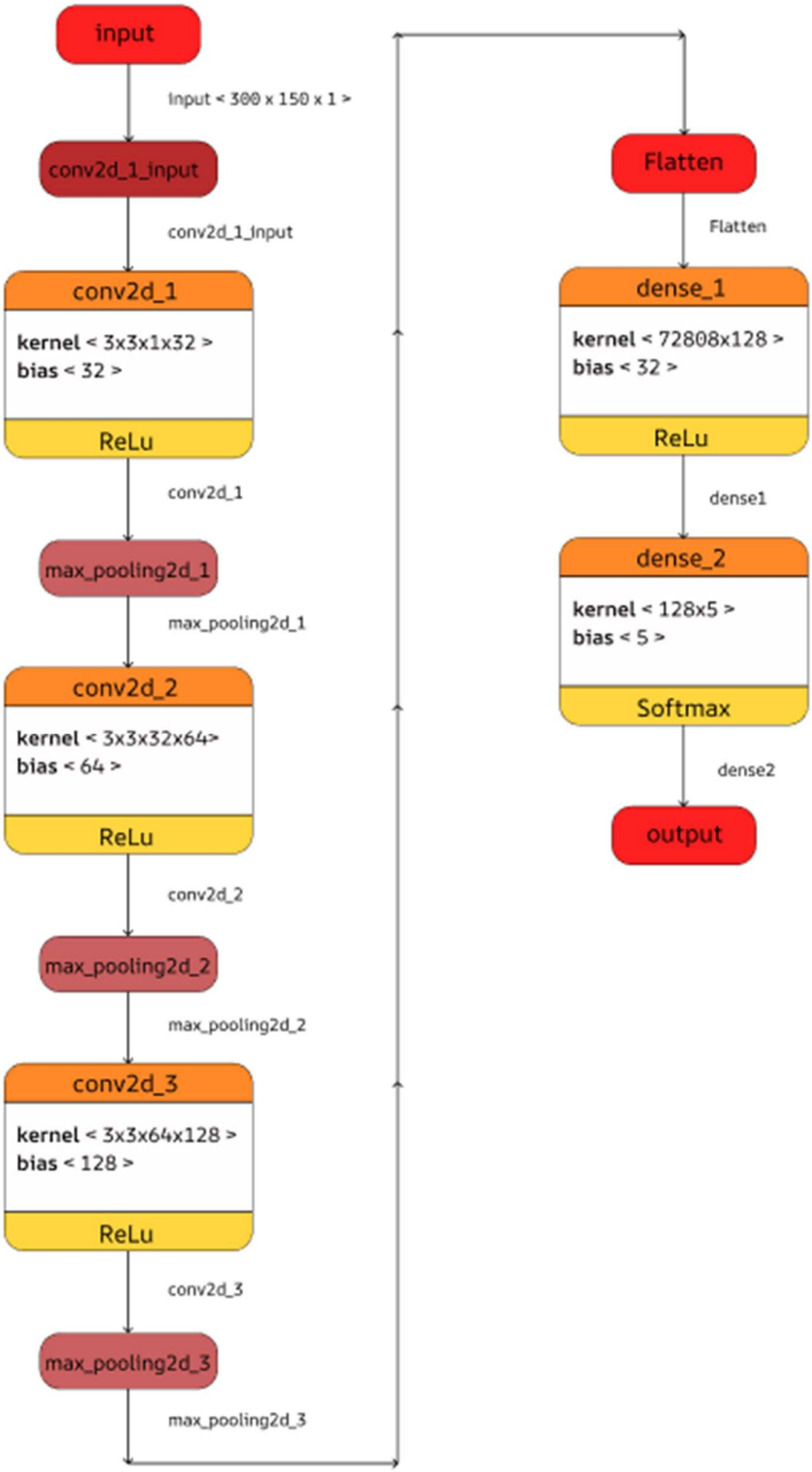

Figura 20 Schema dell'architettura della rete neurale convoluzionale utilizzata

3.5 ADDESTRAMENTO

Nel machine learning, la fase di addestramento riguarda l'ottimizzazione dei parametri interni della rete neurale per permettere lo svolgimento di attività cognitive, come il riconoscimento di immagini in questo caso di studio. Durante il training, la rete apprende dai dati di esempio riducendo progressivamente una funzione di costo che quantifica l'errore tra le previsioni della rete e i valori attesi (o *Ground Truth*). Lo scopo è ottenere una rete in grado di generalizzare adeguatamente anche su dati nuovi, non inclusi nel set di addestramento. Una giusta suddivisione del dataset (**vedi paragrafo 3.3**) e una corretta gestione dei parametri migliorano significativamente la capacità di apprendimento della rete. La discesa stocastica del gradiente (SGD) è una tecnica iterativa per l'ottimizzazione di funzioni differenziabili ed è una variante della discesa del gradiente (GD). Questo metodo è particolarmente efficace quando la funzione obiettivo che si vuole minimizzare può essere rappresentata come la somma di funzioni di costo individuali calcolate su ogni esempio nel dataset. Nel contesto della statistica e del Machine Learning, spesso si lavora con funzioni $Q(w)$, definite come:

$$Q(w) = \frac{1}{n}\sum_{i=1}^{n} Q_i(w) \tag{7}$$

Ogni termine di tale espressione corrisponde al costo associato alla i-esima osservazione. Tale problema appare tipicamente in applicazioni del metodo dei minimi quadrati o del metodo della massima verosimiglianza, e gli stimatori ottenuti come soluzione di tali problemi sono chiamati stimatori M. Il metodo di discesa del gradiente esegue iterazioni nella forma:

$$w := w - \eta \nabla Q(w) = w - \eta \frac{1}{n}\sum_{i=1}^{n} \nabla Q_i\,(w) \tag{8}$$

Dove η è un iperparametro che controlla l'ampiezza di ogni passo, e nel contesto dell'apprendimento automatico è chiamato tasso di apprendimento (learning rate) [32]. Il *Learning Rate* controlla l'ampiezza del passo da compiere nella direzione del gradiente negativo. Seguendo questo gradiente negativo per ogni nuovo campione o lotto di campioni scelti dal set di dati, si ottiene una stima locale di quale direzione minimizza il costo e si fa riferimento alla discesa stocastica del gradiente. La scelta del *Learning Rate* comporta tipicamente una procedura di messa a punto in cui il tasso più alto possibile viene scelto a mano. Scegliere un tasso più alto di questo può causare la divergenza del sistema in termini di funzione obiettivo, e scegliere questo tasso troppo basso si traduce in un apprendimento lento [33]. Per evitare queste problematiche si possono utilizzare tecniche di decadimento del tasso di apprendimento, che prevedono una riduzione graduale nel tempo. Queste tecniche prevedono di accelerare il raggiungimento di una soluzione accettabile compiendo passi più ampi nella fase iniziale di addestramento, per poi successivamente ridurre il passo quando ci sia avvicina alla soluzione ottimale.

Il metodo utilizzato per ottimizzare funzioni differenziabili a SGD che includono meccanismi di adattamento automatico del *Learning Rate* in base all'andamento dell'addestramento è l'algoritmo Adam (*Adaptive Moment Estimation*), che ha la capacità di regolare in maniera dinamica il tasso di apprendimento in base al cambiamento dei gradienti durante la fase di addestramento. Questo metodo comporta una grande vantaggio per quanto riguarda l'adattabilità e i risultati.

Un'altra tecnica è quella di utilizzare il parametro dell'*Early Stopping*. In questo caso l'addestramento viene interrotto quando le prestazioni su un set di validazione non migliorano oppure peggiorano per un numero consecutivo di epoche. Questo metodo,

oltre a prevenire l'*overfitting*, permette anche di risparmiare risorse computazionali e ottenere modelli più performanti su nuovi dati. (Figura 21).

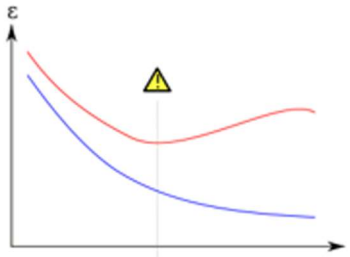

Figura 21 Grafico che mostra il funzionamento dell'Early Stopping

3.6 IMPLEMENTAZIONE RETE NEURALE

In questo paragrafo sono state approfondite le componenti hardware e software impiegate per lo sviluppo del sistema utilizzato. Le librerie utilizzate sono diverse, come prima si cita *Keras*, una libreria open-source, scritta in *Python*, progettata per il Deep Learning che facilita la sperimentazione con reti neurali. La versione utilizzata è la 3.6.0. Questa libreria è stata integrata in *Tensorflow*, framework per il machine learning open source, molto utilizzato per le su ottime capacità di elaborazione e flessibilità. *Tensorflow* è stato sviluppato per migliorare l'uso delle risorse hardware, in particolare l'utilizzo della GPU per velocizzare i calcoli necessari durante l'addestramento dei modelli. La versione utilizzata è la 2.17.0. Come ambiente di sviluppo è stato utilizzato Visual Studio Code nella verisone 1.94., ambiente conosciuto per le sue funzionalità, come l'integrazione con sistemi di controllo di versione, evidenziazione della sintassi e auto completamento. Per l'etichettatura e l'elaborazione delle registrazioni è stato utilizzato *Audacity*, un software open source di registrazione e modifica di audio, apprezzato per la sua semplice

interfaccia.

Per le operazioni matematiche e per lavorare con gli array, è stata utilizzata la libreria *NumPy*, che offre ottime prestazioni utili per la gestione di grandi quantità di dati. È stata anche fondamentale per la creazione degli spettrogrammi in scala di grigi tramite la trasformata veloce di Fourier applicata ai segnali audio. La versione utilizzata è la 1.26.4. Per l'applicazione del filtro *Sobel* alle immagini degli spettrogrammi è stato utilizzato il framework *OpenCV*, il quale supporta diverse operazioni sulle immagini come la manipolazione base o il rilevamento di oggetti.

Per progettare e implementare il software è stato utilizzato il sistema hardware Acer Swift SF314-43, processore AMD Ryzen 7 5700U, memoria RAM da 16GB e sistema operativo Windows 11 a 64 bit. La GPU è una Radeon Graphics 1.80 GHz che non è stata sfruttata per questo lavoro poiché non compatibile con il framework *Tensorflow*.

CAPITOLO 4

RISULTATI

Per addestrare la rete neurale convoluzionale è stata utilizzata la *10-Cross Fold Validation* e per ogni modello è stato impostato un numero massimo pari a 90 epoche, con un *Early Stopping* impostato per interrompere l'addestramento nel caso in cui non migliorasse per cinque epoche consecutive. Una volta completato il test sul dataset derivante dalle etichette delle registrazioni con 50 immagini per classe.

I modelli sono stati valutati attraverso le metriche di *Precision*, *Recall*, *Accuracy* e *F1-Score*, di cui sono stati riportati i valori medi (Figura 22 – Figura 23 – Figura 24 – Figura 24), mentre nella Tabella 1 sono state riportate le medie di questi valori tra le 10 fold con la loro deviazione standard (dispersione dei dati rispetto alla media), infine i valori di TP, TN, FP e FN sono stati riportati nella Tabella 2.

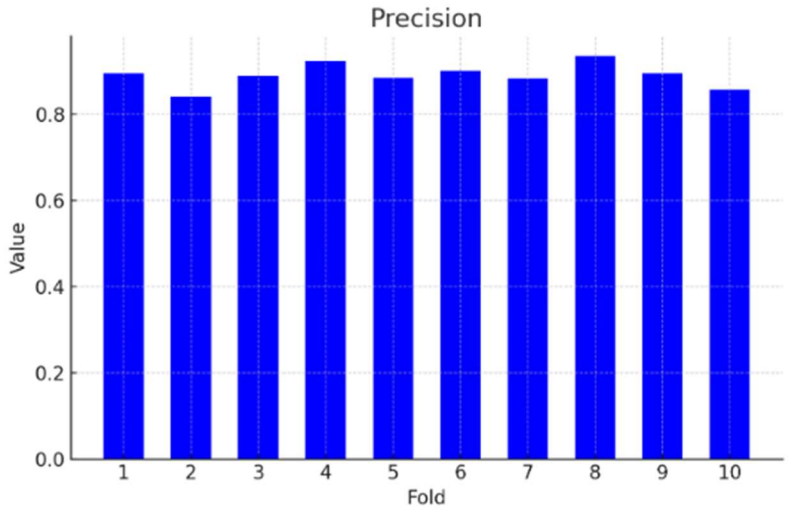

Figura 22 Valori della Precision nelle 10 fold

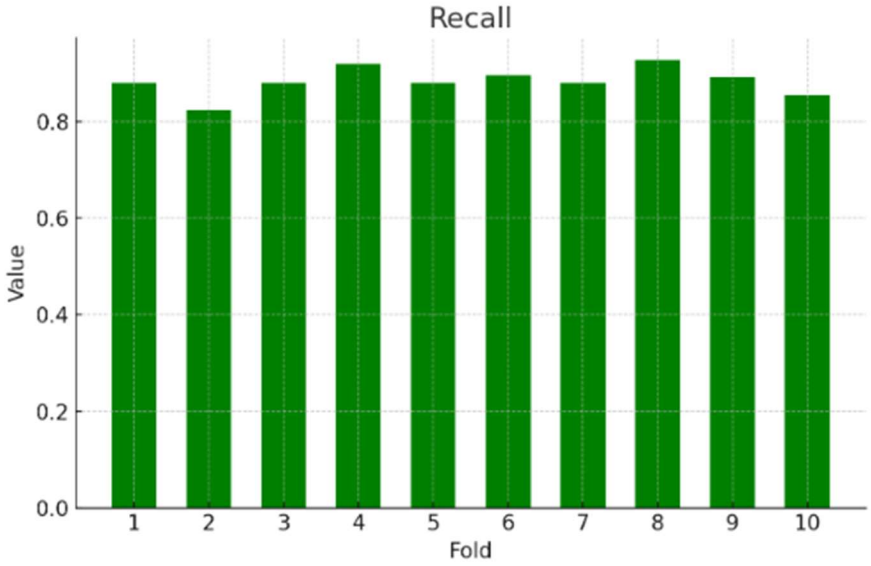

Figura 23 Valori del Recall nelle 10 fold

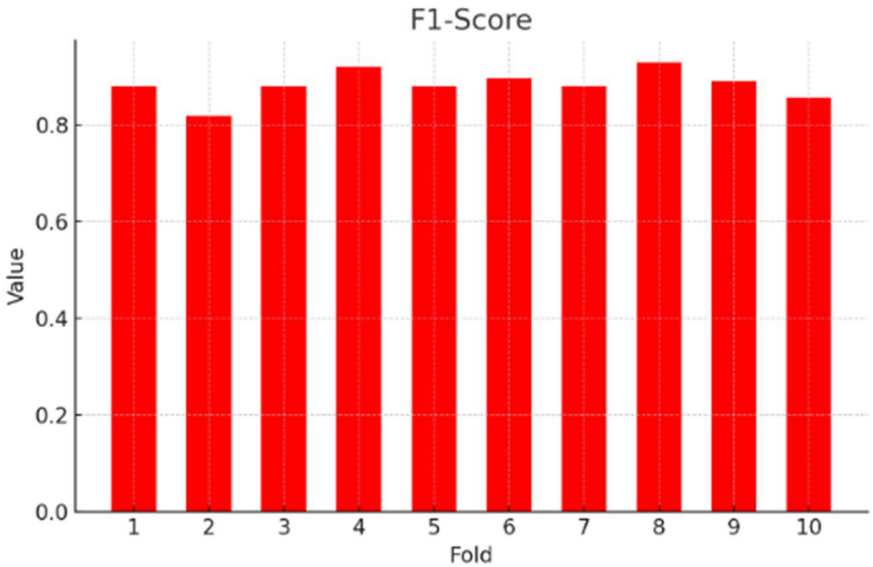

Figura 24 Valori dell' F1-score nelle 10 fold

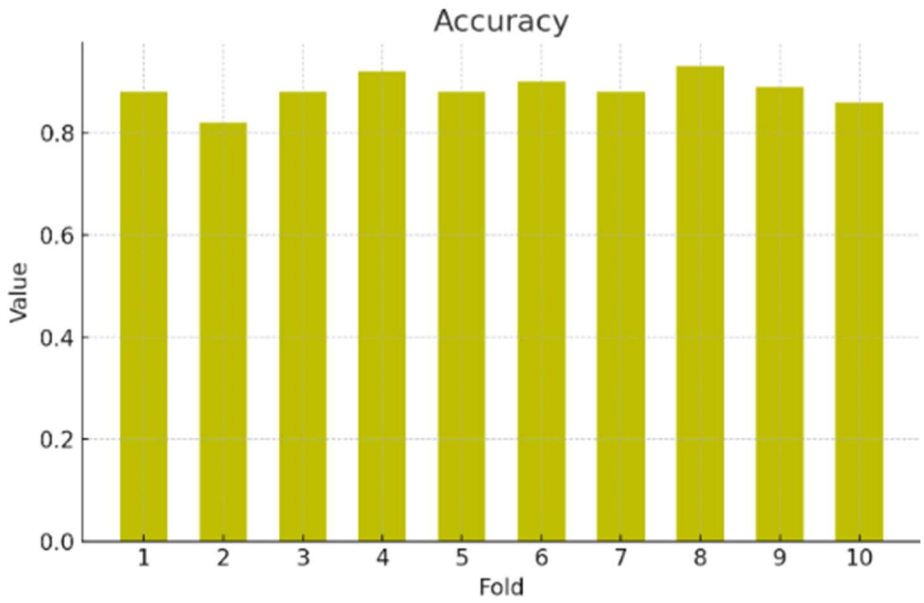

Figura 25 Valori dell' Accuracy nelle 10 fold

Tabella 1 Risultati 10-Cross Validation

	PRECISION (%)	**RECALL (%)**	**F1-SCORE (%)**	**ACCURACY (%)**
Average	89.1	88.4	88.4	88.4
Dev. St.	2.7	2.9	2.9	2.9

Tabella 2 Parametri TP, TN, FP e FN

FOLD / CLASSI		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Valori medi**
0 - RUMORE	TP	48	42	49	47	48	50	44	50	48	49	47.5
	TN	197	195	197	200	198	194	200	199	198	197	197.5
	FP	3	5	3	0	2	6	0	1	2	3	2.5
	FN	2	8	1	3	2	0	6	0	2	1	2.5
1 - FISCHIO	TP	47	45	47	50	48	44	50	49	48	47	47.5
	TN	189	192	199	197	198	200	194	200	198	199	196.6
	FP	2	8	1	3	2	0	6	0	2	1	2.5
	FN	3	5	3	0	2	6	0	1	2	3	2.5
2 - CLICK	TP	40	47	47	45	45	42	42	44	46	40	43.8
	TN	197	189	185	197	188	196	195	197	188	192	192.4
	FP	3	11	15	3	12	4	5	3	12	1	6.9
	FN	10	3	3	5	5	8	8	6	4	3	5.5
3 - BPS	TP	47	45	41	47	41	46	45	47	41	40	44
	TN	180	181	191	189	188	187	188	187	191	185	187.3
	FP	20	19	9	11	12	13	12	13	9	15	13.3
	FN	3	5	9	3	9	4	5	3	9	10	6
4 - BUZZ	TP	38	27	36	41	38	42	39	42	40	38	38.1
	TN	198	199	198	197	198	197	193	199	198	191	196.8
	FP	2	1	2	3	2	3	7	1	2	9	3.2
	FN	12	23	14	9	12	8	11	8	10	12	11.9

Dalle tabelle possiamo notare che i valori medi delle metriche di valutazione (*Precision*, *Recall*, *Accuracy*, *F1-Score*) sono tutte superiori all' 88%. Le immagini utilizzate per il test sono pre-elaborate e generate allo stesso modo delle immagini utilizzate per la fase di addestramento, e questo ha ridotto la variabilità dei dati consentendo alla rete di raggiungere prestazioni elevate.

La *Confusion Matrix* evidenzia come le predizioni si distribuiscono tra le cinque classi. Nella Figura 23 è riportata la *Confusion Matrix* normalizzata del modello con le prestazioni più elevate.

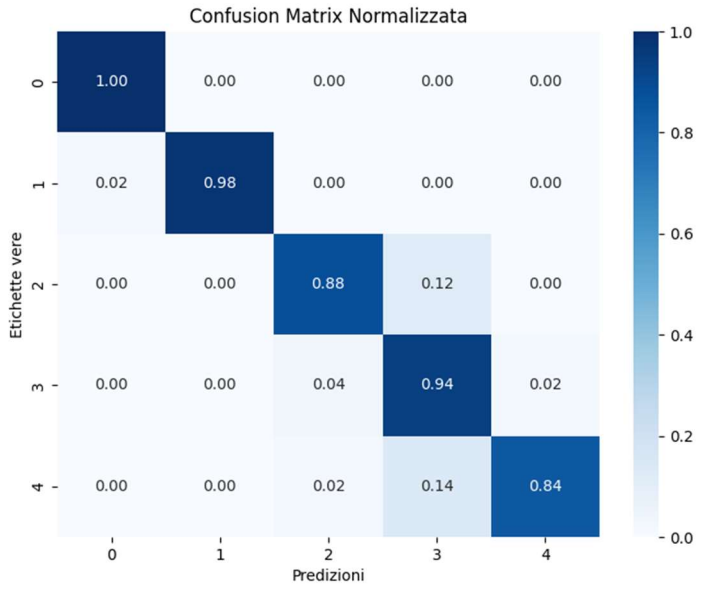

Figura 26 Confusion Matrix normalizzata

Come si può notare dal grafico della *Confusion Matrix*, il modello commette qualche errore sull'identificazione della classe 4, che corrisponde alla classe dei feeding buzz, confondendoli con la classe 2 e la classe 3, ovvero con click e burst pulse sound.

CAPITOLO 5

DISCUSSIONI E CONCLUSIONI

Il presente lavoro è stato svolto con l'intento di introdurre un nuovo approccio basato sul deep learning per identificare le diverse vocalizzazioni dei tursiopi comuni partendo dalle registrazioni audio subacquee, fornendo un monitoraggio acustico più completo del comportamento del delfino. Siccome le vocalizzazioni dei delfini vengono comunemente studiate nel dominio della frequenza attraverso l'analisi di opportuni spettrogrammi, l'idea di base del presente lavoro è classificare le diverse vocalizzazioni tramite le forme d'onde presenti negli spettrogrammi. Nello specifico, le vocalizzazioni sono state classificate in cinque differenti classi e cioè: rumore (o assenza di vocalizzazione), fischio, click di ecolocalizzazione, burst pulse sound e feeding buzz.

Il filtro di Sobel è una tecnica di filtraggio di immagini che viene tipicamente impiegata in computer vision e nell'elaborazione avanzata delle immagini per evidenziare i contorni e i bordi [7]. L'idea alla base del presente lavoro risiede nell'applicazione del filtro Sobel (orizzontale o verticale secondo le necessità) agli spettrogrammi rappresentanti le specifiche vocalizzazioni dei delfini al fine di evidenziare i bordi e i contorni delle immagini, migliorando così il processo di addestramento di una eventuale rete neurale e, di conseguenza, le prestazioni nell'identificazione di tali vocalizzazioni. Utilizzando in modo opportuno le proprietà di questo filtro, è possibile enfatizzare la forma d'onda di una specifica vocalizzazione in modo da renderla più riconoscibile a un possibile approccio machine/deep learning. I fischi dei delfini sono segnali acustici omnidirezionali a banda stretta con modulazione di frequenza, che rappresentano uno strumento essenziale di comunicazione per i delfini, trasmettendo informazioni sul loro ambiente,

le interazioni sociali e l'identità individuale [2]. In uno spettrogramma, il fischio si sviluppa principalmente sul piano orizzontale. Applicando il filtro Sobel verticale, è quindi possibile eliminare le componenti verticali del segnale audio nello spettrogramma, incluse le tipiche vocalizzazioni impulsive del delfino, isolando così nell'immagine i soli fischi. I click di ecolocalizzazione e i burst pulse sound sono vocalizzazioni impulsive a banda larga (principalmente ultrasonici) che si differenziano per la durata degli intervalli tra i click che è più breve per i burst pulse sound [2]. Anche i feeding buzz sono vocalizzazioni impulsive, ma sono caratterizzate da un contenuto in frequenza molto inferiore rispetto a click di ecolocalizzazione e burst pulse sound (non oltre i 5 kHz). Analogamente a quanto discusso per i fischi, utilizzando il filtro Sobel orizzontale, si eliminano le componenti orizzontali del segnale audio nello spettrogramma (principalmente i fischi), evidenziando così nell'immagine le sole componenti verticali a differente frequenza e durata degli intervalli tra i click. Per quanto ne sappiamo, questo è il primo tentativo di utilizzare il filtro di Sobel per evidenziare la forma d'onda delle vocalizzazioni del delfino, al fine di renderle più riconoscibile per una rete neurale multiclasse.

L'efficacia del presente approccio è supportata dalla Confusion Matrix riportata in Figura 23, dove i rumori sono stati riconosciuti al 100%, i fischi al 98%, i click all'88%, i burst pulse sound al 94% e infine i feeding buzz all'84%. Le prestazioni dettagliate di classificazione riportate in Tabella I sembrano confermare quanto emerso dalla matrice di confusione e cioè che i risultati presentati possono presentare un punto di partenza incoraggiante per la classificazione multiclasse delle vocalizzazioni dei delfini. Sono stati infatti ottenuti valori medi incoraggianti per l'accuracy di classificazione (88.4%), per la precision (89.1%), per la recall (88,4) e per l'F1-score medio (88.4%). Le basse deviazioni

standard associate ai valori medi di questi tra i 10 fold (mai oltre il 3%) indicano che il modello ha operato in modo coerente sui diversi sottoinsiemi di dati (folds). Questo suggerisce che le prestazioni osservate riflettono effettivamente le capacità del modello, piuttosto che essere influenzate da variazioni casuali nei dati, dimostrando così un'elevata coerenza e affidabilità delle prestazioni del modello. Tali promettenti valori sono stati ottenuti anche grazie all'adozione della stessa procedura per la generazione degli spettrogrammi sia nei set di test e che in quelli di addestramento, centrando la vocalizzazione all'interno della finestra di 0,8 secondi. Questo procedimento ha incrementato la coerenza dei set di dati migliorando le prestazioni della rete. Centrare la vocalizzazione all'interno della finestra si è rivelato molto efficace; tuttavia, questo approccio è perseguibile solamente in scenari in cui è possibile il post-processing del segnale registrato.

I risultati della presente analisi (Tabella 2 e Figura 23) hanno evidenziato che la classe con performance peggiori, e cioè dove la rete neurale commette più errori di classificazione, è la classe 4 e cioè quella caratterizzata dai feeding buzz. Le principali motivazioni di questo fenomeno sono da attribuirsi probabilmente al numero limitato di spettrogrammi utilizzate sia in fase di training che di test rispetto alle altre vocalizzazioni. Questo purtroppo è un limite intrinseco del dataset utilizzato che presenta un numero piuttosto limitato di feeding buzzes poiché queste vocalizzazioni sono emesse dai delfini più raramente e solamente durante la fase di nutrizione. Un'altra motivazione potrebbe essere la loro somiglianza con le altre vocalizzazioni poiché la loro forma d'onda è molto simile alle forme d'onda di click e burst pulse sound.

In conclusione, i presenti risultati hanno evidenziato il potenziale dell'utilizzo integrato di tecniche deep learning e monitoraggio acustico per affrontare sfide ambientali

complesse. I risultati ottenuti suggeriscono che le tecniche di intelligenza artificiale possono contribuire significativamente al monitoraggio dei delfini, anche in condizioni difficili come la presenza di vocalizzazioni multiple contemporanee da parte dei delfini. I progressi nelle tecniche di intelligenza artificiale e di apprendimento automatico, infatti, hanno il potenziale di rivoluzionare il modo in cui gli ecosistemi marini vengono studiati e monitorati. Per sviluppi futuri ci si potrebbe concentrare sull'ottimizzare le prestazioni e migliorare la robustezza, attraverso filtri più selettivi e centrati sulla specifica vocalizzazione. Un altro punto di sviluppo potrebbe essere l'ampliamento del dataset per l'addestramento della rete neurale con più dati etichettati provenienti da diversi fonti e condizioni ambientale; questo aiuterebbe il modello a diventare più versatile, riducendo il rischio di overfitting e migliorando la capacità della rete di funzionare in condizioni variabili. Un ulteriore importante miglioramento del presente approccio potrebbe essere quello di provare un secondo approccio per la generazione delle immagini, che tenti di simulare le reali condizioni di rilevamento, per testare le funzionalità della rete neurale convoluzionale in condizioni dinamiche, simulando una identificazione e classificazione in tempo reale delle vocalizzazioni dei delfini.

BIBLIOGRAFIA

[1] Life DELFI. "Il progetto Life DELFI." https://lifedelfi.eu/progetto/.

[2] M. O. Lammers, J. N. Oswald, Analyzing the acoustic communication of dolphins. In Dolphin Communication and Cognition Past, present and Future (MIT Press, Cambridge, MA, 107–137, 2015).

[3] E. L. White, P. R. White, J. M. Bull, D. Risch, S. Beck & E. W. Edwards, More than a whistle: Automated detection of marine sound sources with a convolutional neural network. Frontiers in Marine Science, 9, 879145, 2022.

[4] C. Bergler, M. Schmitt, A. K. Maier, H. Symonds, P. Spong, S. R. Ness, … & E. Nöth. ORCA-SLANG: An Automatic Multi-Stage Semi-Supervised Deep Learning Framework for Large-Scale Killer Whale Call Type Identification. In Interspeech (pp. 2396-2400), 2021.

[5] G. Frainer, E. Dufourq, J. Fearey, S. Dines, R. Probert, S. Elwen, & T. Gridley. Automatic detection and taxonomic identification of dolphin vocalisations using convolutional neural networks for passive acoustic monitoring. Ecological Informatics, 78, 102291, 2023

[6] S. Liu, "Classification of cetacean whistles based on convolutional neural network." 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP). IEEE, 2018.

[7] N. Kanopoulos, N. Vasanthavada R. L. Baker, Design of an image edge detection filter using the Sobel operator. IEEE J. Solid-State Circuits 23(2), 358-367 (1988).

[8] R. S. Wells, M. D. Scott, Common Bottlenose Dolphin: Tursiops truncatus, Encyclopedia of Marine Mammals (Second Edition), Pages 249-255, 2009.

[9] V. M. Janik, Chapter 4 Acoustic Communication in Delphinids, Advances in the Study of Behavior, Academic Press, Volume 40, Pages 123-157, 2009.

[10] M. Lalot, F. Delfour, B. Mercera, *et al.* Prosociality and reciprocity in bottlenose dolphins (*Tursiops truncatus*), *Animal Cognition*, *24*(5), 1075-1086, 2021.

[11] B. Jones, M. Zapetis, M. M. Samuelson, S. Ridgway. Sounds produced by bottlenose dolphins (Tursiops): A review of the defining characteristics and acoustic criteria of the dolphin vocal repertoire. Bioacoustics, 29(4), 399-440, 2020.

[12] D. Scaradozzi, R. De Marco, D. Li Veli, A. Lucchetti, L. Screpanti, F. Di Nardo. Convolutional Neural Networks for Enhancing Detection of Dolphin Whistles in a Dense Acoustic Environment, *IEEE Access,* 2024.

[13] R. Carluccia, G. Cipriano, M. Bonato, G. Buscaino, R. Crugliano, C. Fanizza, S. Gatto, R. Maglietta, C. Papetti, M. Pelagatti, P. Riccia, F.C. Santacesaria, E. Papale. Anthropogenic noise effects on Risso's dolphin vocalizations in the Gulf of Taranto (Northern Ionian sea, central Mediterranean sea), *Ocean & Coastal Management*, *254*, 107177, 2024.

[14] M. Xiang, Y. Chen, Z. Li, K. Li, Z. Liu, Z. Zhao, J. Chen, Classification of Typical Tursiops aduncus Whistle Signals Using Convolutional Neural Networks. ACTA ACUSTICA, 41(2): 181-188, 2016.

[15] Sala, A., Brčić, J., De Carlo, F., Lucchetti, A., Pulcinella, J., Virgili, M., Valutazione delle catture accidentali di specie protette nel traino pelagico: BYCATCH Estensione 2013. Relazione finale, 58 pagine, 2014.

[16] Y. Wu, J. Feng, Development and Application of Artificial Neural Network. Wireless Pers Commun 102 , 1645–1656, 2018

[17] Cichy, M. Radoslaw et al. Deep Neural Networks as Scientific Models, Trends in Cognitive Sciences, Volume 23, Issue 4, 305 - 317, 2019

[18] A. Ajith, Artificial Neural Networks. In Handbook of Measuring System Design (eds P.H. Sydenham and R. Thorn), 2005.

[19] R. Sharma, S. Kavya, K. Apurva. "Study of supervised learning and unsupervised learning." International Journal for Research in Applied Science and Engineering Technology 8.6, 588-593, 2020.

[20] V. Nasteski, An overview of the supervised machine learning methods. Horizons. b, 4(51-62), 56, 2017.

[21] B. Mahesh, Machine learning algorithms-a review. International Journal of Science and Research (IJSR). [Internet], 9(1), 381-386, 2020.

[22] Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, 33(12), 6999-7019. 2021.

[23] K. O'Shea, "An introduction to convolutional neural networks." arXiv preprint arXiv:1511.08458, 2015.

[24] S. Sharma, S. Sharma, & Athaiya, A., Activation functions in neural networks. Towards Data Sci, 6(12), 310-316, 2017.

[25] D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto, & S. Ridella. The'K'in K-fold Cross Validation. In *ESANN* (Vol. 102, pp. 441-446). 2012.

[26] X. Ying. An overview of overfitting and its solutions. In *Journal of physics: Conference series* (Vol. 1168, p. 022022). IOP Publishing, 2019.

[27] M.M. Bejani, M. Ghatee, A systematic review on overfitting control in shallow and deep neural networks. *Artif Intell Rev* **54**, 6391–6438, 2021.

[28] Y. Lemoigne, & Caner, A. *Molecular imaging: Computer reconstruction and practice*. Springer Science & Business Media, 2008.

[29] W. T. Zhang, D. Cui & S. T. Lou. Training images generation for CNN based automatic modulation classification. *IEEE Access*, *9*, 62916-62925, 2021.

[30] N. Pielawski, C. Wählby. Introducing Hann windows for reducing edge-effects in patch-based image segmentation. *PloS one*, *15*(3), e0229839, 2020.

[31] H. V. Sorensen, D. Jones, M. Heideman, C. Burrus. Real-valued fast Fourier transform algorithms. *IEEE Transactions on acoustics, speech, and signal processing*, *35*(6), 849-863, 1987.

[32] G. Sabbatini. Algoritmi di Machine Learning per classificare il comportamento di drosophhile in presenza di mutazioni genetiche, 2018.

[33] M. D. Zeiler. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

SITOGRAFIA

1) https://www.oltremare.org/

2) https://www.cnr.it/it/nota-stampa/n-10824/l-algoritmo-che-puo-salvare-i-delfini

3) https://adrianoamalfi.com/di-cosa-parliamo-quando-parliamo-di-ai/