



**UNIVERSITÀ
DEGLI STUDI
DI PADOVA**



**DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE**

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN BIOINGEGNERIA

**“SVILUPPO DI UN SISTEMA PROTOTIPALE PER IL
RICONOSCIMENTO AUTOMATICO DELL'OTITE MEDIA DA
IMMAGINI RACCOLTE CON OTOSCOPI PROFESSIONALI E LOW
COST”**

Relatore: Prof. Giovanni Sparacino

Laureanda: Sara Bertolini

Correlatore: Prof. Pietro Scimemi

ANNO ACCADEMICO 2022 – 2023

Data di laurea 17/04/2023

*“Doubt kills more dreams than
failure ever will.”*

- *Suzy Kassem*

INDICE

Indice delle Figure	9
Indice delle Tabelle	10
Sommario	13
1 Cenni di Fisiopatologia dell'Orecchio Medio e Scopo della Tesi	15
1.1 Anatomia dell'Orecchio.....	15
1.1.1 Orecchio Esterno	15
1.1.2 Orecchio Medio	16
1.1.3 Orecchio Interno	18
1.2 Patologie dell'Orecchio Medio	21
1.3 Strumenti per l'Indagine sull'Orecchio Medio	22
1.3.1 Otoscopi Analogici	22
1.3.2 Strumenti Digitali Professionali	23
1.3.3 Strumenti low Cost	24
1.4 Scopo Della Tesi e Presentazione	25
2 Database, Preprocessing e Criteri di Valutazione	27
2.1 Aspetti Generali	27
2.2 Database Principale.....	27
2.3 Database Aggiuntivi.....	28
2.4 Preprocessing	29
2.5 Criteri di Valutazione.....	30
3 Metodologia Basata su CNN per l'Analisi delle Immagini Timpaniche	35
3.1 Artificial Intelligence e Deep Learning	35
3.2 Le Convolutional Neural Networks (CNN).....	35
3.2.1 Convolutional Neural Networks.....	36
3.2.2 Multi-Layer Perceptron	38

3.3	Matlab.....	42
3.4	Descrizione delle Reti Usate nella Tesi.....	43
3.4.1	GoogLeNet.....	43
3.4.2	ResNet50.....	43
3.4.3	MobileNet V2	45
3.4.4	Transfer Learning.....	47
3.5	Descrizione degli Step Svolti	47
4	Risultati.....	49
4.1	Risultati ottenuti sul Database per valutare l' Accettabilità delle Immagini	49
4.2	Risultati ottenuti sul Database per il Riconoscimento di MT Destra e Sinistra	50
4.3	Risultati Ottenuti sul Database Principale.....	51
4.4	Problema dell'interpretabilità.....	57
4.5	Discussione.....	59
5	Conclusioni e Sviluppi Futuri.....	63
6	Appendice 1: Concetti Base delle Reti Neurali	65
6.1	Neuroni "Reali" e Neuroni Artificiali	65
6.1.1	Perceptron	66
6.1.2	Funzione Logistica come funzione di Attivazione	68
6.1.3	Il problema dell'XOR	68
7	Ringraziamenti.....	75
8	Bibliografia.....	77
9	Bibliografia Immagini	81

INDICE DELLE FIGURE

Fig. 1.1: Anatomia Dell'orecchio, Visione Generale.....	15
Fig. 1.2 Orecchio Esterno	16
Fig. 1.3 Orecchio Medio.....	18
Fig. 1.4 Membrana Timpanica	18
Fig. 1.5 Orecchio Interno.....	20
Fig. 1.6 Generazione Dell'impulso Nervoso.....	20
Fig. 1.7 Esempi Di MT Affetta Da OM	22
Fig. 1.8 Otoscopio Analogico.....	23
Fig. 1.9 Otoscopio Digitale Professionale	24
Fig. 1.10 Otoscopio Low Cost.....	25
Fig. 2.1 Esempio Mt Con Om	28
Fig. 2.2 Esempio Mt Con Om	28
Fig. 2.3 Esempio Immagine Con Tappo Di Cerume (Altro).....	28
Fig. 2.4 Esempio Mt Normale	28
Fig. 2.5 Esempio Mt Normale	28
Fig. 2.6 Esempio Immagine Con Miringosclerosi (Altro)	28
Fig. 2.7 Esempio Immagine Mt Sx (Accettabile).....	29
Fig. 2.8 Esempio Immagine Mt Dx (Accettabile)	29
Fig. 2.9 Esempio Immagine Non Accettabile	29
Fig. 2.10 Esempio Immagine Non Accettabile.....	29
Fig. 2.11 Esempio Confusion Matrix Per Classificazione Binaria.....	31
Fig. 2.12 Esempio Confusion Matrix Multi-Classe.....	32
Fig. 2.13 Esempio Curve Roc.....	33
Fig. 3.1 Schema Organizzazione Intelligenza Artificiale.....	35
Fig. 3.2 Schema CNN.....	36
Fig. 3.3 Schema Multi-Layer Perceptron	39
Fig. 3.4 Schema 1	40
Fig. 3.5 Schema 2	41
Fig. 3.6 Sgd.....	42
Fig. 3.7 Skip Connections	43
Fig. 3.8 Schema GoogLeNet	46
Fig. 3.9 Schema ResNet50	46
Fig. 3.10 Schema Transfer Learning	47

Fig. 3.11 Rete In Fase Di Addestramento	48
Fig. 4.1 Confronto Modelli Resnet50	49
Fig. 4.2 Esempio Di Classificazione	50
Fig. 4.3 confusion Matrix Per Riconoscimento Destra E Sinistra	51
Fig. 4.4 Esempio Di Classificazione Destra E Sinistra	51
Fig. 4.5 Confronto Confusion Matrix Per GoogLeNet	52
Fig. 4.6 Confronto Confusion Matrix ResNet50.....	52
Fig. 4.7 Confronto Confusion Matrix Per MobileNetv2.....	52
Fig. 4.8 Confronto ROC Per GoogLeNet	54
Fig. 4.9 Confronto ROC Per ResNet50.....	54
Fig. 4.10 Confronto ROC Per MobileNetv2	55
Fig. 4.11 Confronto AUC-ROC Per GoogLeNet.....	55
Fig. 4.12 Confronto AUC-ROC Per ResNet50.....	56
Fig. 4.13 Confronto AUC-ROC Per MobileNetv2	56
Fig. 4.14 Confronto AUC Otite Media Per Le Diverse Reti.....	56
Fig. 4.15 Esempio Predizione Normale	58
Fig. 4.16 Esempio Gradcam Normale.....	58
Fig. 4.17 Esempio Gradcam Otite Media	58
Fig. 4.18 Esempio Predizione Otite Media	58
Fig. 4.19 Esempio Gradcam Altro	58
Fig. 4.20 Esempio Predizione Altro	58
Fig. 6.1 Struttura Dei Neuroni	65
Fig. 6.2 Schema Del Perceptron.....	66
Fig. 6.3 Grafico Della Funzione Logistica.....	67
Fig. 6.4 Grafico Funzione Tangente Iperbolica	67
Fig. 6.5 Grafico Funzione ReLU	68
Fig. 6.6 Problema Dell'XOR.....	69

INDICE DELLE TABELLE

Tab. 1 Valori Accuracy E AUC-ROC Per ResNet50 Normalizzata.....	53
Tab. 2 Problema Dell'XOR	69

SOMMARIO

L'idea di questa tesi nasce dall'esigenza di riuscire a trovare un metodo automatico (che prescinde dal medico specialista) per diagnosticare in modo quanto più corretto possibile l'otite media (OM), patologia che nei bambini dai 3 mesi ai 3 anni è l'infezione batterica più frequente e la prima causa di prescrizione antibiotica e di consulenza pediatrica [1] [2] [3]. Inoltre se l'otite media non è diagnosticata correttamente può risultare particolarmente pericolosa in quanto potrebbe portare a sviluppi anche molto gravi come la perdita dell'udito, ma anche problemi legati allo sviluppo del linguaggio e delle abilità cognitive [4] [5].

Ad oggi per eseguire una diagnosi di OM è necessario rivolgersi a un medico specialista il quale, come unico mezzo di analisi, usa uno strumento chiamato otoscopio che, nella maggior parte dei casi, è di tipo analogico con un piccolo led alimentato a batteria che consente di illuminare il canale uditivo. Più raramente lo specialista si può servire anche di otoscopi digitali i quali sono però molto costosi e sono vincolati a specifici software da installare su PC; al momento non sono ancora disponibili sul mercato dispositivi medici certificati che possano sostituire i precedenti, tuttavia esistono degli strumenti low cost che possono svolgere lo stesso compito. Riuscire a combinare il prezzo conveniente di questi dispositivi con l'intelligenza artificiale potrebbe aiutare a ottenere una buona accuratezza nonostante la qualità inferiore delle immagini raccolte e l'utilizzo di un'applicazione su smartphone potrebbe risolvere il limite dato dalla necessità di avere un PC.

Lo scopo di questa tesi è quello di eseguire un "proof of concept", in altre parole valutare la fattibilità, di un'ipotetica applicazione che dovrebbe avere come obiettivo quello di distinguere le immagini delle membrane timpaniche con OM, utilizzando uno strumento dal costo contenuto.

Questa applicazione potrebbe risultare molto utile per eseguire uno screening nell'ambito familiare, ma anche nell'ambiente scolastico (dal momento che uno dei fattori che aumenta la probabilità di avere un'otite è proprio la socialità), consentendo a genitori e/o insegnanti di eseguire un controllo rapido, riducendo così i tempi tra la diagnosi e l'inizio di un'eventuale terapia. Un altro scenario in cui questa applicazione potrebbe risultare utile sono i paesi in via di sviluppo, in cui spesso le strutture non hanno budget sufficiente per l'acquisto di dispositivi costosi e dove è difficile trovare dei medici specializzati che possano occuparsi di questi casi, ad esempio, nell'afrika sud-sahariana c'è poco più di un otorinolaringoiatra per un milione di persone [6]. Inoltre, soprattutto dopo i recenti eventi legati alla pandemia, risulta di

fondamentale importanza investire in servizi di sanità sempre più decentralizzati, dando la possibilità di eseguire degli screening da casa per poi rivolgersi direttamente al medico specialista.

Per svolgere il compito di riconoscimento delle varie immagini si è pensato di applicare una particolare branca dell'intelligenza artificiale, ovvero quella del deep learning, su una serie di immagini eseguite con strumenti diversi, con qualità diversa e con una selezione di immagini di diverse patologie comuni. Le analisi sono state svolte con diversi tipi di rete neurale e una volta effettuato il training sono state controllate le performance su un set di immagini usato come test. I risultati trovati sono hanno tutti valori di accuracy superiori a 90% e valori di AUC-ROC superiori a 88%.

Da queste analisi si può vedere come, nonostante alcuni limiti che saranno discussi nel corso della trattazione, l'implementazione di una tale applicazione è possibile e con risultati che possono essere soddisfacenti e utili per la salute e la prevenzione di patologie legate all'udito.

1 CENNI DI FISIOPATOLOGIA DELL'ORECCHIO MEDIO E SCOPO DELLA TESI

Le informazioni per questo capitolo sono state tratte da [23] e [2].

1.1 ANATOMIA DELL'ORECCHIO

L'orecchio è formato in gran parte da organi contenuti nell'osso temporale e assicura la percezione di due tipi di stimolo, grazie alla presenza di due tipi di recettori: i recettori acustici per gli stimoli sonori e i recettori stato-cinetici per gli stimoli gravitari e di accelerazione.

Come si vede in Fig. 1.1 l'apparato dell'udito viene diviso in tre parti: orecchio esterno, orecchio medio e orecchio interno; i primi due sono di esclusiva pertinenza della sensibilità uditiva, mentre l'ultimo si divide in due parti contenenti i due diversi tipi di recettori.

Le sensibilità uditiva e stato-cinetica sono raccolte nel nervo cocleare la prima e nel nervo vestibolare la seconda.

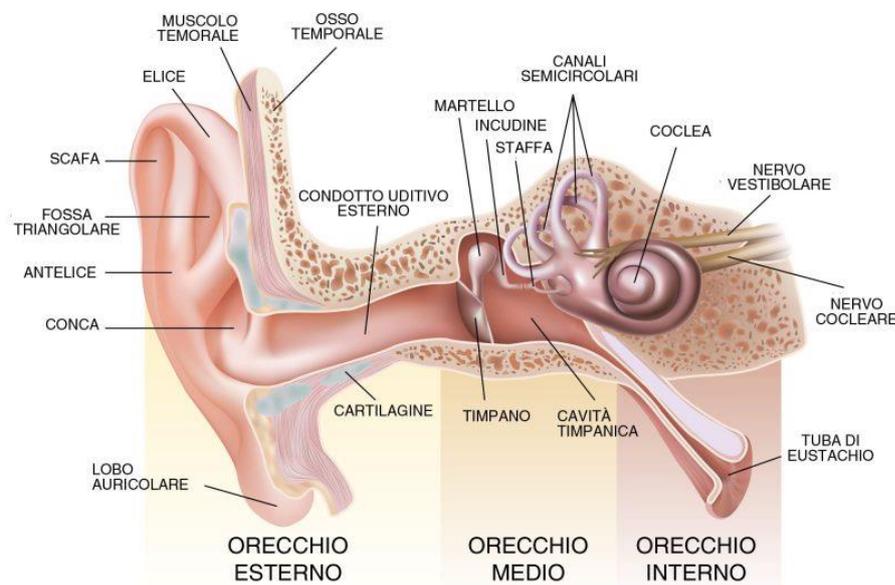


Fig. 1.1: Anatomia Dell'orecchio, Visione Generale

Fonte[1]

1.1.1 ORECCHIO ESTERNO

Come si può vedere in Fig. 1.2, l'orecchio esterno comprende il padiglione auricolare e il meato acustico esterno, ha la funzione di raccogliere le onde sonore e di convogliare verso la membrana del timpano.

Il padiglione esterno è costituito da uno scheletro fibrocartilagineo rivestito da cute e presenta una forma irregolare con diverse conche; dal punto di vista funzionale; il padiglione acustico nell'uomo ha una funzione ridotta, tuttavia è accertata una sua funzione nella localizzazione della sorgente sonora nello spazio e nella concentrazione dell'energia sonora verso il meato acustico esterno.

Il meato acustico esterno si estende dalla conca del padiglione all'orecchio medio, terminando a livello della membrana del timpano.

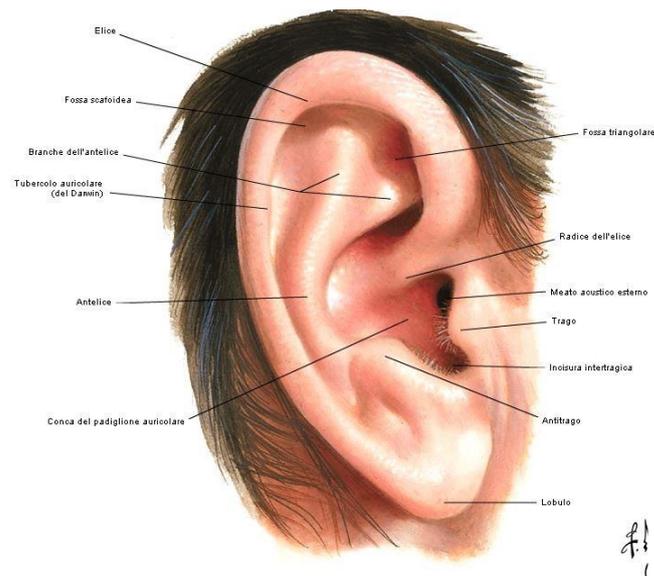


Fig. 1.2 Orecchio Esterno

Fonte [2]

1.1.2 ORECCHIO MEDIO

È costituito dalle seguenti parti: il cavo del timpano, la membrana timpanica (MT), la catena degli ossicini dell'udito, l'apparato mastoideo e la tuba uditiva.

IL CAVO DEL TIMPANO

Il cavo del timpano è una camera ossea posta tra l'orecchio esterno e l'orecchio interno, accoglie la catena degli ossicini e comunica con la faringe attraverso la tuba uditiva e con l'apparato mastoideo mediante l'aditus ad antrum.

LA MEMBRANA TIMPANICA

Come illustrato nella Fig. 1.4 la MT ha forma ellittica e chiude il fondo del meato acustico esterno, separandolo dal cavo del timpano. Presenta due superfici:

- Una rivolta verso l'esterno sulla quale, per trasparenza, si può intravedere la sagoma del manico del martello, è rivestita da un sottile strato cutaneo
- Una superficie rivolta verso l'interno sulla quale è applicato il manico del martello ed è tappezzata dalla mucosa del cavo timpanico.

Nella MT si distinguono due parti: una più estesa, mobile, la pars tensa e, sopra di questa un'altra parte più piccola di forma triangolare, la pars flaccida.

- La pars tensa include il manico del martello che termina nell'ombelico, sotto al quale è possibile vedere il riflesso della luce dell'otoscopio detto triangolo luminoso (in condizioni normali). Si compone di tre strati: uno strato cutaneo esterno, uno fibroso intermedio e uno mucoso interno; lo strato cutaneo e il continuo di quello del meato esterno; lo strato fibroso è costituito da fibre radiate, circolari e semifinali che conferiscono le caratteristiche strutturali che consentono alla pars tensa di vibrare; lo strato di mucosa che tappezza la faccia interna è la stessa che riveste il cavo del timpano, con il quale è in continuità.
- La pars flaccida è la porzione superiore della membrana e si mette in rapporto con il processo laterale del martello. La struttura della pars flaccida è uguale a quella della pars tensa, a parte per lo strato fibroso, che è sostituito da tessuto connettivo lasso.

LA CATENA DEGLI OSSICINI

La catena degli ossicini dell'udito, mostrata nella Fig. 1.3, è formata dal martello, dall'incudine e dalla staffa, tre piccole ossa tenute tra loro da un apparato legamentoso e dalla connessione del manico con la MT. Questi ossicini stabiliscono un collegamento tra MT e il vestibolo dell'orecchio interno. L'ampiezza dei movimenti trasmessi alla catena dalla membrana del timpano è regolata da due piccoli muscoli: il tensore del timpano e lo stapedico; questi infatti agiscono in maniera modulato, agonista o antagonista, per migliorare la percezione delle varie tonalità dei suoni. La catena degli ossicini è completamente rivestita dalla mucosa del cavo timpanico.

In altre parole la membrana del timpano, vibrando, trasmette il suo movimento al martello, e questo all'incudine e l'incudine alla staffa.

L'APPARATO MASTOIDEO

L'apparato mastoideo è composto da una serie di cavità comunicanti con il cavo del timpano e contenenti aria, che sono accolte nell'osso temporale. È composto di tessuto osseo spugnoso rivestito di tessuto osseo compatto; il suo ruolo è quello di dare sostegno e, grazie alla

presenza delle cavità mastoidee nell'osso spugnoso, di svolgere un ruolo anche nel bilanciamento della pressione dell'aria nel cavo del timpano.

LA TUBA Uditiva

La tuba uditiva è un condotto mediante il quale la cavità timpanica comunica con la rinofaringe attraverso due muscoli collegati con il velo del palato. Dal punto di vista funzionale, la tuba uditiva ha il compito di drenare nella faringe le secrezioni del cavo timpanico e di permettere la penetrazione dell'aria, dalla faringe, nel cavo del timpano e nelle cavità mastoidee. Questa seconda funzione permette alla MT di vibrare in condizioni ottimali: infatti l'aria che dalla faringe penetra, per mezzo della tuba uditiva, nel cavo del timpano, equilibra, sulla faccia interna della MT, la pressione che l'aria contenuta nel meato acustico esterno esercita sulla faccia esterna. Un'alterazione della pervietà tubarica provoca perciò alterazioni della pressione normalmente esistente nel cavo del timpano, con conseguente compromissione del buon funzionamento della membrana del timpano e del sistema di trasmissione delle onde sonore.

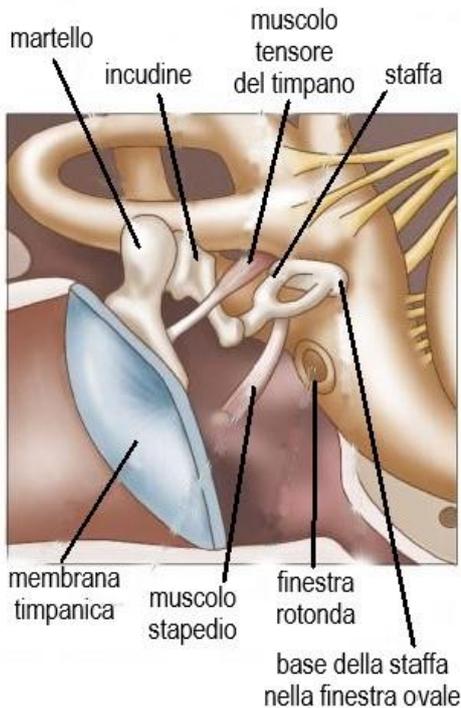


Fig. 1.3 Orecchio Medio

Fonte [3]

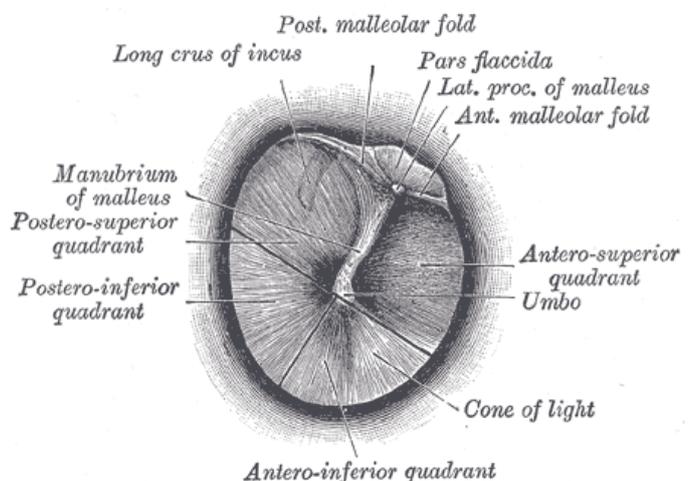


Fig. 1.4 Membrana Timpanica

Fonte [4]

1.1.3 ORECCHIO INTERNO

È costituito dal labirinto osseo, un complesso sistema di cavità scavate nello spessore della piramide del temporale, e dal labirinto membranoso, un insieme di vescicole e condotti membranosi contenuti nella cavità del labirinto osseo. A separare il labirinto osseo da quello

membranoso si interpone lo spazio perilinfatico, formato da un complesso di fessure fra loro comunicanti nelle quali è contenuto un liquido, la perilinfa.

IL LABIRINTO OSSEO

Come si può vedere in Fig. 1.5 il labirinto osseo è costituito da una parte posteriore, o vestibolare, che comprende il vestibolo e i canali semicircolari osseo e l'acquedotto del vestibolo (mette in comunicazione la cavità del vestibolo con la cavità cranica); e una parte anteriore, o acustica, costituito dalla chiocciola (o coclea) e dell'acquedotto della chiocciola. Mediamente al labirinto osseo si trova il meato acustico interno.

I canali semicircolari ossei sono tre: anteriore, posteriore e laterale e sono situati posteriormente in alto rispetto al vestibolo, sono orientati nei tre piani dello spazio e per questo sono un sistema adatto alla rilevazione di accelerazioni angolari prodotte dalla rotazione della testa.

LA CHIOCCIOLA (O COCLEA)

La chiocciola ossea corrisponde al fondo del meato acustico interno e si avvolge a spirale. Nella coclea avviene per merito delle cellule ciliate la trasduzione mecano-elettrica.

IL MEATO ACUSTICO INTERNO

Il meato acustico interno è un canale osseo che dall'osso temporale si dirige verso l'orecchio interno.

IL LABIRINTO MEMBRANOSO

Tutte le cavità menzionate sono rivestite da una sottile membrana connettiva e contengono un complesso schema di sacchi membranosi ai quali si dà il nome di labirinto membranoso. Il labirinto membranoso è costituito da: tre canali semicircolari (corrispondenti a quelli del labirinto osseo), l'utricolo, il sacco, il sacco endolinfatico e la coclea, o chiocciola membranosa.

All'interno del condotto cocleare c'è l'Organo di Corti, è formato da un insieme di cellule di sostegno, e sensitive che costituiscono la membrana reticolare, una struttura rigida in cui sono incasellate le parti apicali delle cellule sensitive che presentano i fascetti di stereociglia. I vari movimenti della membrana reticolare e la sovrastante membrana tectoria permettono la deflessione delle ciglia e la creazione di impulsi nervosi.

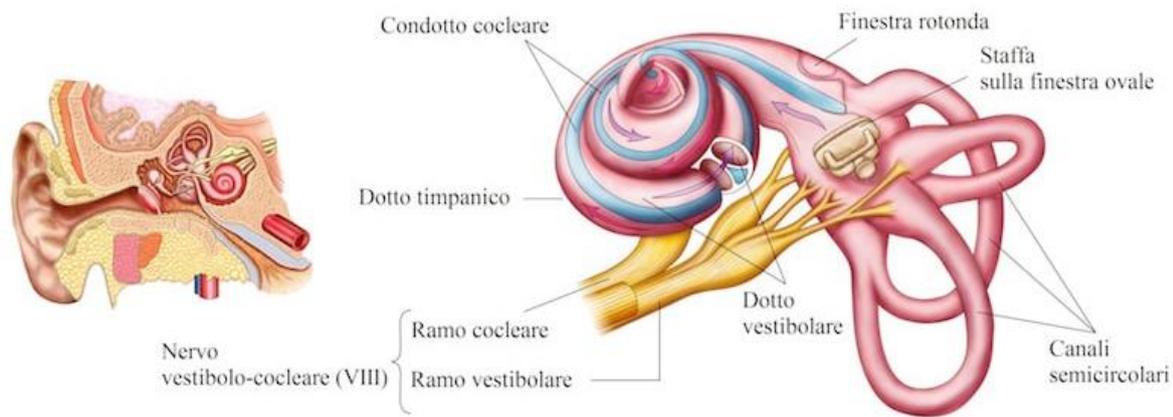


Fig. 1.5 Orecchio Interno

Fonte [5]

GENERAZIONE DELL'IMPULSO NERVOSO

Come mostrato nella Fig. 1.6, gli spostamenti di fluido contenuto nel labirinto membranoso generano dei movimenti della membrana basilare che si ripercuotono nell'Organo di Corti in modo tale che l'organo e la sovrastante membrana tectoria deflettano le stereociglia delle cellule acustiche che portano a cambiamenti del potenziale di membrana, in maniera molto simile a quello che accade nelle cellule ciglia dell'apparato vestibolare.

Questi cambiamenti sono percepiti dal cervello come un cambiamento di frequenza di scarica delle fibre afferenti del nervo vestibolare e tali informazioni vengono poi confrontate con segnali visivi o somatosensitivi che permettono di rilevare la posizione della testa.

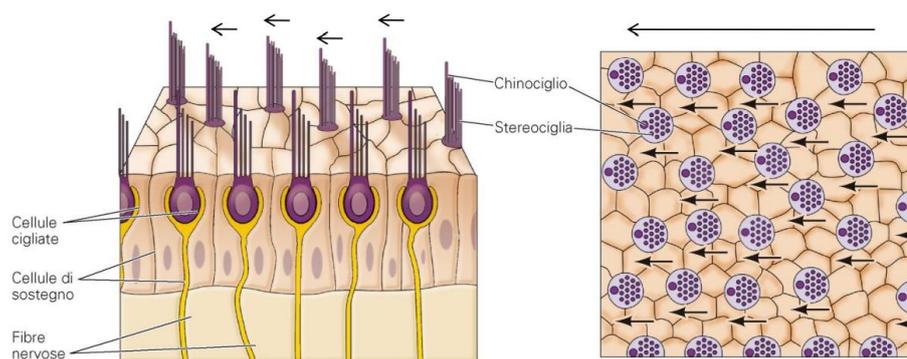


Fig. 1.6 Generazione Dell'impulso Nervoso

Fonte [6]

1.2 PATOLOGIE DELL'ORECCHIO MEDIO

Le principali patologie dell'orecchio medio sono riconducibili alle otiti, l'OM è un processo infiammatorio della mucosa timpanica che può essere provocato da virus o batteri. Le otiti, in base all'essudato che viene a formarsi possono essere distinte in catarrali o purulente, in base al decorso clinico si possono classificare in acute e croniche.

OTITE MEDIA

L'otite media (OM) è un'infezione batterica della mucosa dell'orecchio medio, alcuni esempi di come si presenta sono visibili in Fig. 1.7, particolarmente frequente nell'infanzia dai 6 mesi ai 3 anni; l'incidenza varia in funzione dell'età. Nel bambino dai 3 mesi ai 3 anni è l'infezione batterica più frequente e la prima causa di prescrizione antibiotica e di consulenza pediatrica.

E' una patologia multifattoriale, la disfunzione tubarica e l'imaturità del sistema immunitario giocano un ruolo essenziale nello sviluppo dell'OM. Il bambino fino ai 6 mesi è protetto dagli anticorpi materni grazie all'allattamento, successivamente la diminuzione delle IgA secretorie della mucosa delle alte vie aeree, associata all'imaturità del sistema immunitario favorisce la colonizzazione batterica della rinofaringe, che diventa un reservoir di agenti patogeni, che possono diffondere all'orecchio medio attraverso la tuba di Eustachio.

I fattori predisponenti sono molteplici, alcuni sono: collettività (frequentazione dell'asilo nido), familiarità (sia per cause genetiche che socio-economiche), stagionalità (le OM sono più frequenti nel periodo invernale poiché il principale fattore di rischio è rappresentato dalle infezioni virali delle alte vie aeree), precocità del primo episodio otitico (le recidive sono più frequenti nei bambini in cui il primo episodio infettivo dell'orecchio medio si sia verificato nei primi sei mesi di vita).

Nell'adulto i sintomi sono rappresentati da otodinia, descritta dal paziente inizialmente come un dolore sordo, continuo nel fondo del Condotto Uditivo Esterno (CUE) poi, nella fase essudativa, come dolore pulsante, lancinante con crisi parossistiche per lo più notturne. Si associano otorrea nel caso di perforazione timpanica, ipoacusia di tipo trasmissivo, autofonia, fullness auricolare.

Le principali caratteristiche della membrana timpanica da osservare in endoscopia sono: il colore, l'infiltrazione, la posizione e la motilità. All'inizio l'OM si manifesta per la comparsa di un timpano più opaco e scomparsa del triangolo luminoso con iperemia perimalleolare; in seguito l'iperemia interessa tutta la membrana con ispessimento della stessa. L'aumento della

permeabilità vasale e la costituzione di un versamento interno alla cassa estroflettono la membrana timpanica che appare bombata e “a pelle di coccodrillo” (fase di essudazione); l’evoluzione può essere o meno verso la perforazione della membrana timpanica (fase della perforazione) con otorrea muco-purulenta nel condotto uditivo esterno.

L’OM evolve verso la guarigione spontanea nella maggior parte dei casi.



Fig. 1.7 Esempi Di MT Affetta Da OM

Fonte [7]

L’OM può essere anche di tipo secretivo, che comporta una flogosi subacuta o cronica dell’orecchio medio accompagnata da un versamento endotimpanico senza i segni ed i sintomi di infezione acuta.

La presenza di questa patologia si può dedurre da alcuni sintomi come otalgia, spesso accompagnata da ipoacusia; per i bambini più piccoli questo porta a irritabilità, difficoltà nell’addormentarsi e febbre. L’esame otoscopico però è il metodo principale con cui fare diagnosi più sicure per i casi di OM, questo infatti può evidenziare MT estroflessa, iperemica e scomparsa del triangolo di luminoso. Un altro esame che può essere utilizzato nei casi più ambigui è quello della timpanometria, che in presenza di OM mostrerà una MT con mobilità ridotta.

1.3 STRUMENTI PER L’INDAGINE SULL’ORECCHIO MEDIO

1.3.1 OTOSCOPI ANALOGICI

Questo tipo di otoscopio è quello più usato per eseguire gli esami otoscopici, è utilizzato sia dai medici specialisti sia dai medici di base e dai pediatri, è il classico otoscopio visibile in Fig. 1.8.



Fig. 1.8 Otoscopio Analogico

Fonte [8]

Questa categoria di otoscopi è la più semplice disponibile in commercio, in quanto constano di pochi elementi quali: una luce necessaria a illuminare il canale uditivo, una lente d'ingrandimento che permette di visualizzare meglio i dettagli all'interno dell'orecchio, degli specula, intercambiabili (usa e getta oppure sterilizzabili) necessari per questioni di praticità e di igiene ed infine una batteria per alimentare la sorgente luminosa (led o fibra ottica).

Questo tipo di otoscopi è il più diffuso, come detto prima, poiché è estremamente semplice da usare, può essere usato senza bisogno di un supporto software e ha un range di prezzo molto ampio, che va dai 30€ ai 180€, che rende accessibile quasi a tutti l'acquisto di uno di questi strumenti.

1.3.2 STRUMENTI DIGITALI PROFESSIONALI

Questo tipo di otoscopio è meno diffuso, sicuramente per il costo estremamente elevato: ad esempio Aurical Otocam 300 (visibile in figura), l'otoscopio che sarà menzionato successivamente nella strumentazione utilizzata in questa tesi, ha un prezzo di circa 2000€.



Fig. 1.9 Otoscopio Digitale Professionale

Fonte [9]

Come si può vedere dalla Fig. 1.9, questo tipo di otoscopio non prevede l'utilizzo di lenti, ma contiene una piccola telecamera posta sulla punta dell'otoscopio che è in grado di registrare l'immagine e proiettarla, attraverso il collegamento via cavo, su un PC con il supporto software necessario per l'interfaccia (in questo caso Ototsuite) attraverso il quale sarà possibile acquisire le immagini, memorizzarle e modificarle.

Ci sono molti altri strumenti simili a questo, tuttavia non sono molto diffusi e si trovano solo in reparti, ambulatori o studi specializzati proprio per via del loro costo elevato.

1.3.3 STRUMENTI LOW COST

Questa è una categoria di otoscopi che si sta evolvendo, al momento la maggior parte di questi strumenti sono reperibili online (sui siti di grande distribuzione come Amazon) ma non sono in commercio in quanto dispositivi medici, ovvero non dispongono della marcatura CE (marcatura eseguita dal fabbricante per i dispositivi medici, è obbligatoria per tutti i prodotti disciplinati dai Regolamenti comunitari “dispositivi medici e dispositivi medici in vitro”, e ne attesta la conformità).

Il vantaggio di questi strumenti, al momento utilizzati in via sperimentale per applicazioni cliniche, è quello di essere estremamente economici e dalle dimensioni compatte, hanno inoltre la possibilità di interfacciarsi con qualsiasi sistema operativo (sia da PC sia da smartphone) rendendoli molto versatili e portabili.



Fig. 1.10 Otoscopio Low Cost

Fonte [10]

Quello in Fig. 1.10 ad esempio è lo strumento che è stato utilizzato per l'acquisizione di alcune immagini che sono poi state successivamente utilizzate per questa tesi.

1.4 SCOPO DELLA TESI E PRESENTAZIONE

Lo scopo di questa tesi è duplice: si propone di capire se è possibile utilizzare sistemi di Intelligenza Artificiale come le Convolutional Neural Networks per eseguire in modo automatico il riconoscimento dell'otite media, ma vuole anche indagare la possibilità di utilizzare uno strumento low cost come quello menzionato in precedenza per acquisire le immagini.

Nei prossimi capitoli saranno presentati rispettivamente:

- Nel capitolo 2 sono descritti il database creato, il software utilizzato, le operazioni preliminari svolte e i criteri che sono stati usati per valutare le prestazioni dei futuri modelli;
- Nel capitolo 3 vengono illustrati la metodologia utilizzata, le reti utilizzate per questa tesi e perché sono state scelte;
- Nel capitolo 4 sono presentati i risultati che sono stati raggiunti, con un'analisi critica dei risultati e un confronto con i risultati trovati in precedenti studi simili;
- Nel capitolo 5 ci sono le conclusioni e quelli che potrebbero essere gli sviluppi futuri di questa tesi.

2 DATABASE, PREPROCESSING E CRITERI DI VALUTAZIONE

2.1 ASPETTI GENERALI

Il database creato per condurre questa valutazione è composto principalmente da immagini contenute in database pubblici, già disponibili online [7] [8], ed è in seguito stato integrato con alcune immagini raccolte in collaborazione con l'ambulatorio del reparto di Audiologia dell'Ospedale Civile SS Giovanni e Paolo di Venezia e con gli studenti del corso di Tecniche Audioprotesiche dell'università di Padova. Tutte le immagini, sia quelle reperite in letteratura, sia quelle acquisite in loco, sono state associate a una classe.

Gli strumenti usati per prelevare le immagini aggiuntive sono:

- Natus, Aurical Otocam 300: risoluzione dichiarata di 720x720 pixel, collegato con il computer fisso del laboratorio [9]
- Qimic, Wireless Intelligent Visual Ear Cleaning Rod: risoluzione dichiarata di 1080x1080 pixel, collegato con un tablet Lenovo e un iPad sui quali è stata installata l'applicazione indicata dal produttore (Suear, disponibile sia per Android che per iOS) [10]

La seconda telecamera nello specifico è disponibile in commercio online, ma non è venduta come dispositivo medico diagnostico, è stata tuttavia scelta per il suo prezzo contenuto (inferiore ai 50€ al momento dell'acquisto) e la sua accessibilità (distribuito su Amazon). Per questa telecamera è stata però riscontrata una discrepanza tra la risoluzione dichiarata e la qualità effettiva dell'immagine, che risulta visivamente minore rispetto a quella delle immagini catturate con l'otoscopio professionale.

Le immagini acquisite in loco sono state raccolte richiedendo ai 14 studenti volontari del corso di laurea di Tecniche Audioprotesiche di acquisire, per ciascuno, almeno quattro immagini: MT destra e sinistra, sia con l'otoscopio Otoscan e che con il dispositivo low cost. Gli è stato inoltre chiesto di acquisire alcune immagini "sbagliate" (MT parziale o fuori fuoco), che sono state usate in seguito per addestrare la rete riguardante l'accettabilità delle foto. Tutte queste immagini sono state salvate a colori (RGB) nei formati .jpg o .png.

2.2 DATABASE PRINCIPALE

Il database principale è composto di 3 categorie di immagini: MT normale (214 immagini, alcuni esempi sono visibili nelle Fig. 2.4 e Fig. 2.5), Otite Media (107 immagini, alcuni esempi sono quelli nelle Fig. 2.1 e Fig. 2.2) e Altro (123); quest'ultima categoria include due

casi che possono presentarsi comunemente: miringosclerosi (Fig. 2.6) e tappi di cerume (Fig. 2.3).



Fig. 2.2 Esempio Mt Con OM

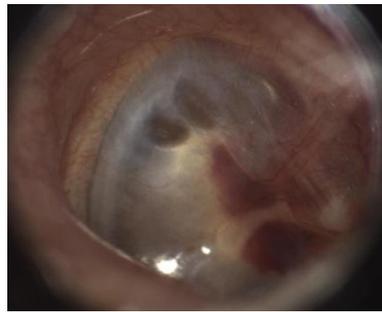


Fig. 2.1 Esempio Mt Con OM



Fig. 2.3 Esempio Immagine Con Tappo Di Cerume (Altro)



Fig. 2.5 Esempio Mt Normale



Fig. 2.4 Esempio Mt Normale



Fig. 2.6 Esempio Immagine Con Miringosclerosi (Altro)

2.3 DATABASE AGGIUNTIVI

Sono stati utilizzati anche altri due database:

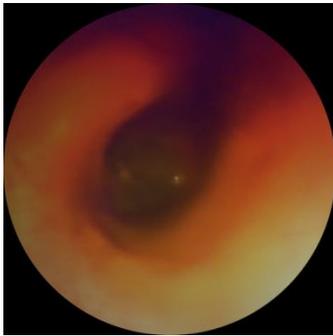
- Il primo, utile nell'ottica della futura creazione di un'applicazione, ha due categorie: accettabile (78 immagini) e non accettabile (80 immagini). Come "accettabile" si è ritenuto di classificare tutte quelle immagini in cui la MT fosse visibile per la maggior parte (almeno $\frac{3}{4}$), correttamente illuminata e a fuoco; come "non accettabile" tutte le immagini che non rispondevano ai criteri appena citati, di queste categorie sono visibili degli esempi nelle Fig. 2.7 e Fig. 2.8.
- Il secondo ha l'obiettivo di distinguere MT destra da MT sinistra e ha due categorie: destra (27 immagini) e sinistra (30 immagini). Per questa categoria sono riportati degli esempi nelle Fig. 2.7, Fig. 2.8, Fig. 2.9 e Fig. 2.10.



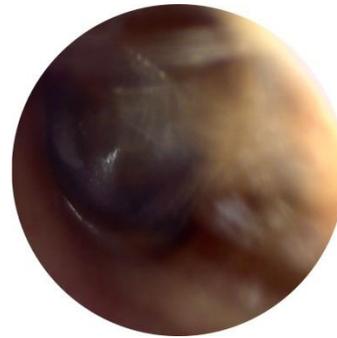
**Fig. 2.8 Esempio Immagine Mt
Sx (Accettabile)**



**Fig. 2.7 Esempio Immagine Mt
Dx (Accettabile)**



**Fig. 2.10 Esempio Immagine
Non Accettabile**



**Fig. 2.9 Esempio Immagine
Non Accettabile**

2.4 PREPROCESSING

Non è stata eseguita nessuna operazione di preprocessing sulle immagini, fatta eccezione per la rimozione di alcune informazioni riportate automaticamente nelle immagini raccolte con Otoscan per motivi legati alla privacy, il software infatti riporta luogo e data di acquisizione nelle immagini che vengono esportate.

Poiché una CNN addestrata può essere “biased”, ovvero presentare una sorta di preferenza, per le classi con un numero maggiore di immagini, per cercare di evitare questo problema è stata creata una seconda versione del database principale in cui il numero di immagini è stato normalizzato rimuovendo in modo casuale le immagini dalle classi più numerose. Così facendo ogni categoria aveva lo stesso numero di immagini della classe più piccola (Otite Media, 107). La normalizzazione è stata applicata solo al database principale, in quanto i database secondari sono stati creati appositamente con lo stesso numero di immagini.

I database sono stati successivamente divisi randomicamente in tre sottogruppi: training (60%), validation (20%) e test set (20%).

Il training set è composto dalle immagini utilizzate per l'apprendimento, il validation set è usato per validare le performance del modello durante il training ed eseguire l'affinamento degli iperparametri, ovvero ci fa capire se il training sta andando bene o no. L'idea dietro l'utilizzo di due set separati di immagini durante la fase di training è quella di evitare l'overfitting, in altre parole diventare molto bravo a classificare le immagini del training set ma non essere in grado di generalizzare ed eseguire predizioni corrette per immagini che non aveva mai visto prima.

Per verificare effettivamente che la rete fosse in grado di riconoscere correttamente le immagini raccolte con il dispositivo low cost alcune delle immagini raccolte sono state inserite nel database che è poi stato diviso in modo casuale, ma sono state tenute da parte alcune immagini, tra cui una patologica, per verificarne l'effettivo riconoscimento.

2.5 CRITERI DI VALUTAZIONE

ACCURACY E CONFUSION MATRIX

L'accuracy ("accuratezza" in italiano) è una metrica che ci dice quante volte il modello ha classificato correttamente un'immagine nel nostro database rispetto al numero di immagini totale e si calcola come segue:

$$Accuracy = \frac{\text{numero_risposte_corrette}}{\text{numero_risposte_totali}}$$

Questa è una misura strettamente legata al contesto nel quale ci troviamo, per questo non è sempre la misura più affidabile da considerare (soprattutto se le classi sono sbilanciate, come nel caso del nostro database originale non normalizzato).

Per eseguire un'analisi più approfondita e corretta vanno valutati anche altri valori, come quello di sensitivity e specificity, per calcolarli usiamo una Confusion Matrix ("Matrice di Confusione" in italiano), rappresentata in Fig. 2.11 queste matrici hanno sulla diagonale le predizioni corrette.

		Predicted	
		Negative (N) -	Positive (P) +
Actual	Negative -	True Negatives (TN)	False Positives (FP) Type I error
	Positive +	False Negatives (FN) Type II error	True Positives (TP)

Fig. 2.11 Esempio Confusion Matrix Per Classificazione Binaria

Fonte [11]

Il significato di ciò che si trova nei riquadri è:

- True Negatives (Veri Negativi, TN): immagini classificate correttamente come negativi
- True Positives (veri Positivi, TP): immagini classificate correttamente come positivi
- False Negatives (Falsi Negativi, FN): immagini classificati in modo errato come negativi, quando in realtà sarebbero positivi
- False Positives (Falsi Positivi, FP): immagini classificate in modo errato come positivi, quando in realtà sarebbero negativi

Quindi l'accuracy si può scrivere secondo questa matrice:

$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN}$$

La precisione, o specificity, è definita come:

$$specificity(TNR) = \frac{TN}{TN + FP}$$

E definisce quanto è sensibile il modello quando deve riconoscere l'immagine, di fatto ci dice quanto spesso classifichiamo correttamente quando classifichiamo un'immagine come positiva.

La sensitivity invece è utile quando ci interessa riconoscere più immagini positive possibile, questo vuol dire includere anche più falsi positivi, e si calcola come:

$$sensitivity(TPR) = \frac{TP}{TP + FN}$$

Queste metriche sono tra loro complementari, quindi quando una delle due cresce, l'altra deve calare.

Nel caso di un classificatore multi classe come quello in Fig. 2.12, la sensitivity (TPR) e la specificity (TNR) si calcolano per una classe specifica, considerando tutte le classi diverse da quella in analisi come una classe singola, come si vede in figura Fig. 2.12

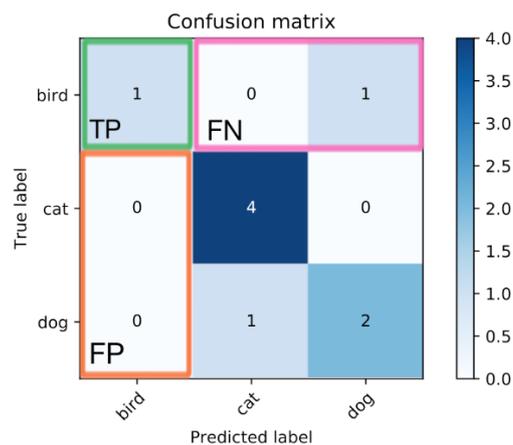


Fig. 2.12 Esempio Confusion Matrix Multi-Classe
Fonte [12]

Le formule viste per il caso del classificatore binario possono essere usate, con le dovute modifiche, anche per il caso multi-classe.

ROC E AUC-ROC

È una curva che rappresenta il rapporto tra riconoscimenti corretti (TPR) e falsi riconoscimenti (1-FPR) e si può rappresentare solo per classificatori binari; nel caso multi classe si può usare la strategia one-vs-all, ovvero considero una classe alla volta come positive e tutte le altre come negativo (otterrò quindi un numero di curve pari al numero di classi che il mio modello deve classificare), come si vede nell'immagine Fig. 2.13.

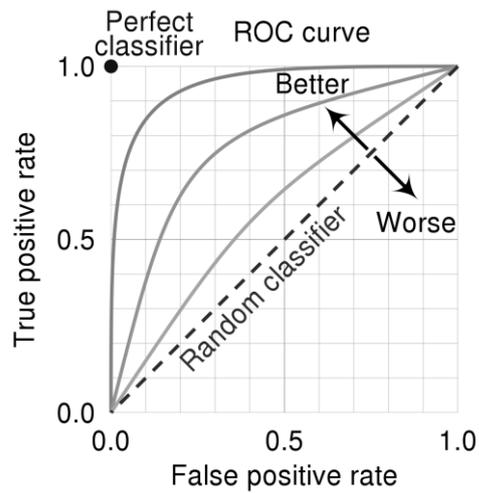


Fig. 2.13 Esempio Curve ROC

Fonte: [13]

AUC sta per Area Under the Curve (Area Sotto la Curva), è una metrica che riassume in un numero le performance del modello permettendo di poterlo confrontare con altri senza guardare solo le curve. Come si vede dal disegno abbiamo un modello perfetto quando TPR e $FPR = 1 - TNR$ sono entrambi pari a 1, ne consegue che tanto più la curva è vicina al punto di coordinate (1,1) e tanto sarà migliore il modello che la genera, e quindi sarà anche maggiore il valore corrispondente di AUC.

3 METODOLOGIA BASATA SU CNN PER L'ANALISI DELLE IMMAGINI TIMPANICHE

3.1 ARTIFICIAL INTELLIGENCE E DEEP LEARNING

L'intelligenza artificiale (Artificial Intelligence, AI) ha come obiettivo quello di fornire algoritmi e tecniche per risolvere i problemi che gli esseri umani risolvono in modo intuitivo e quasi automatico, ma che sono invece complessi per i computer.

Il machine learning è una branca dell'AI che si concentra sul riconoscimento dei pattern in un gruppo di dati, delle tante tecniche che ne fanno parte ci concentreremo su quella del Deep Learning, con particolari strutture e algoritmi molto efficienti nello svolgimento di compiti come la classificazione dei dati, il riconoscimento di oggetti, etc; uno schema che rappresenta questa suddivisione è visibile in Fig. 3.1.

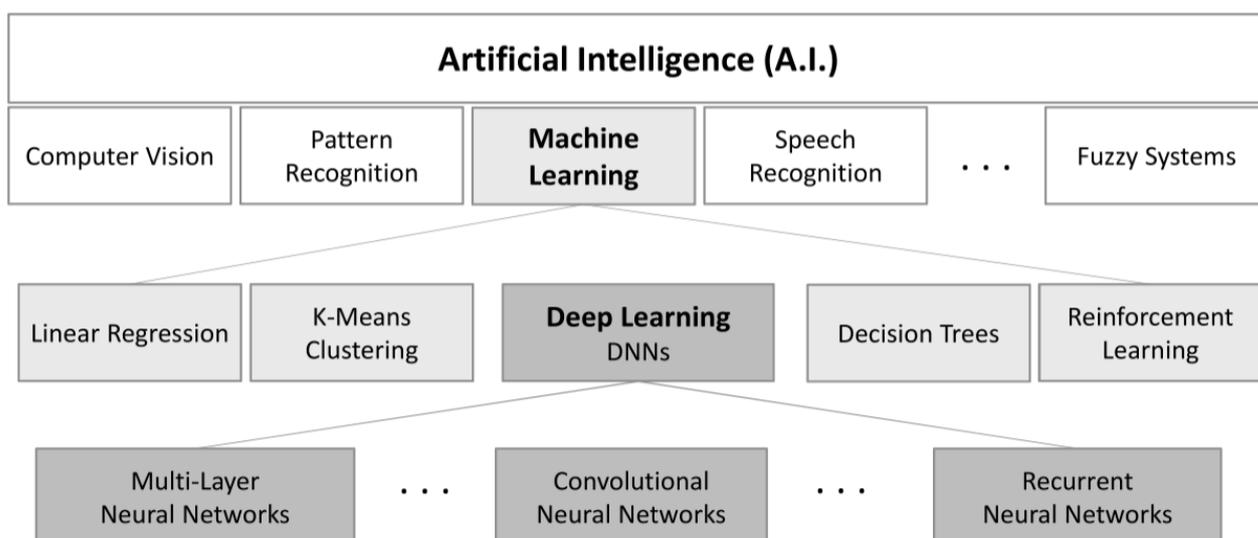


Fig. 3.1 Schema Organizzazione Intelligenza Artificiale

Fonte [14]

In particolare in questa tesi la metodologia scelta è quella delle CNN che sarà presentata nel prossimo capitolo.

3.2 LE CONVOLUTIONAL NEURAL NETWORKS (CNN)

Per la presentazione delle Convolutional Neural Network utilizzate si daranno per scontate alcune nozioni di base che, se necessario sono ricordate nell'appendice 1.

3.2.1 CONVOLUTIONAL NEURAL NETWORKS

Le Convolutional Neural Networks (Reti Neurali Convoluzionali, CNN), il cui schema è rappresentato in Fig. 3.2, sono state sviluppate nel 1990 da Y. LeCun (per il suo lavoro gli è stato assegnato il premio Turing nel 2019) [11], che ha osservato il funzionamento della corteccia cerebrale.

La struttura base delle CNN è composta da:

1. Convolutional Layer
2. Pooling Layer
3. Fully Connected Layer (come MLP)

I vantaggi delle CNN sono:

- Sparse connectivity (connettività scarsa): il numero dei pesi è ridotto rispetto a una rete completamente connessa in quanto tra un layer e l'altro non tutti i neuroni sono connessi tra loro
- Parameter Sharing (condivisione dei parametri): gli stessi pesi vengono usati per tutti i pixel dell'immagine o gruppi di pixel dell'immagine principale.

La sostituzione di uno strato completamente connesso come quello del Multi-Layer Perceptron (che verrà descritto in seguito) consente di ridurre drasticamente il numero di parametri (o pesi) della rete portando dunque ad una maggiore capacità di estrazione delle informazioni rilevanti (nelle immagini è giusto supporre che i pixel vicini tra loro siano più importanti di quelli distanti).

Si parla di deep network in questo caso in quanto il numero di layer è alto.

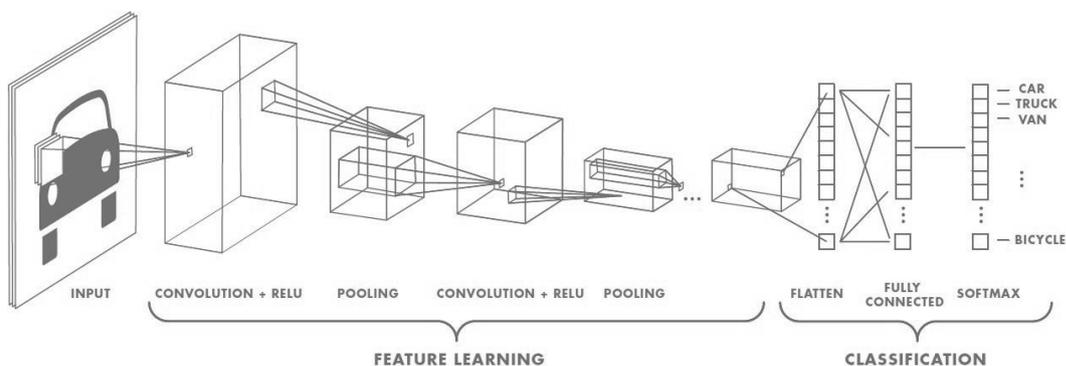


Fig. 3.2 Schema CNN

Fonte [15]

CONVOLUTIONAL LAYER

Il compito di questo strato è quello di estrarre delle caratteristiche dall'input (matrice quadrata $N \times N$), per farlo ci si avvale di diversi filtri (di dimensione $n \times n$, inferiore a quella dell'immagine) che sono usati per eseguire la convoluzione con l'immagine data come input. Ognuno di questi filtri estrae una caratteristica creando una "feature map", ovvero una nuova matrice che contiene in ogni pixel il risultato della convoluzione (maggiore il valore calcolato nel pixel della feature map, maggiore sarà la corrispondenza tra il filtro e l'immagine di input).

All'inizio i filtri vengono inizializzati con dei pesi scelti in modo casuale e sono successivamente raffinati con l'avanzare delle epoche. Due parametri importanti per questa operazione sono:

- **Padding:** possibilità di aggiungere dei pixel fittizi (solitamente contenuti zeri) per evitare di perdere l'informazione contenuta sui bordi dell'immagine (che altrimenti verrebbe persa troppo velocemente).
- **Stride:** dimensione del passo fatto dal filtro nelle direzioni orizzontale e verticale durante la convoluzione

Entrambi questi parametri vanno a definire la dimensione della feature map.

Questa operazione viene svolta per tutte le dimensioni dell'immagine (1 dimensione se l'immagine è in bianco e nero, 3 dimensioni se l'immagine è a colori, RGB) e i filtri vengono successivamente combinati tra loro per estrarre caratteristiche sempre più complesse.

POOLING LAYER

Questo layer è responsabile della procedura per ridurre la dimensione della feature map riducendo così la complessità considerando solo una parte dei dati, questo consente di ridurre la complessità della CNN e l'overfitting, ma anche di rendere la rete resistente alle traslazioni (ovvero la rete sarà in grado di capire se una determinata caratteristica è presente o meno indipendentemente dalla sua posizione), aumentando così la capacità di generalizzazione della rete.

Per eseguire questa operazione si considerano delle sottomatrici della feature map ottenuta nel layer precedente e, tramite tecniche come average pooling e max pooling, si effettuano operazioni di calcolo del valore medio o del valore massimo della sottomatrice.

FLATTENING

Prima di raggiungere l'ultimo layer è necessario passare da uno spazio bidimensionale (matrici) a uno unidimensionale (vettore), questa operazione si chiama flattening. Questo vettore sarà poi passato come input del fully connected layer per eseguire la classificazione.

FULLY CONNECTED LAYER

È il layer responsabile della classificazione, funziona come un tradizionale Multi-Layer Perceptron (MLP).

3.2.2 MULTI-LAYER PERCEPTRON

È una versione più avanzata del perceptrone, proposta nel 1986 da D. Rumelhart, G. Hinton e R. Williams nell'articolo "Learning Internal Representations by Error Propagation" [12], ed è l'esempio più semplice di Artificial Neural Network (il cui funzionamento è descritto in Appendice 2).

È composto da un input layer, strato che presenta neuroni che ricevono dei segnali in input, un output layer, ovvero uno strato di neuroni che producono un output, e da una serie di hidden layers che contengono neuroni (detti hidden neurons) con funzioni di attivazione non lineari (una rete con hidden layers che hanno funzioni di attivazione lineari possono risolvere problemi risolvibili da una rete con un singolo layer, per questo è necessaria la non linearità).

Siccome non conosciamo l'output di ogni singolo neurone di ciascun hidden layer, per il training applico semplicemente l'input e lo lascio propagare attraverso tutta la rete (dopo aver inizializzato i pesi in modo randomico) e calcolo l'output finale, questo step è detto forward propagation. Una volta ottenuto il risultato calcolo l'errore come la differenza tra il valore dell'output prodotto dalla rete con l'output target e retro propago l'errore in modo che l'informazione riguardo questa differenza possa essere utile per aggiornare il valore dei pesi e, ripetendo l'operazione, ottenere un apprendimento sempre maggiore e migliore.

La tecnica dell'error back-propagation (retro propagazione dell'errore) di solito si usa in combinazione con un metodo di ottimizzazione, il più comune è lo Stochastic Gradient Descent (discesa stocastica del gradiente, SGD) che serve a trovare il minimo della funzione di errore, una rappresentazione grafica di questo processo è mostrata in Fig. 3.3.

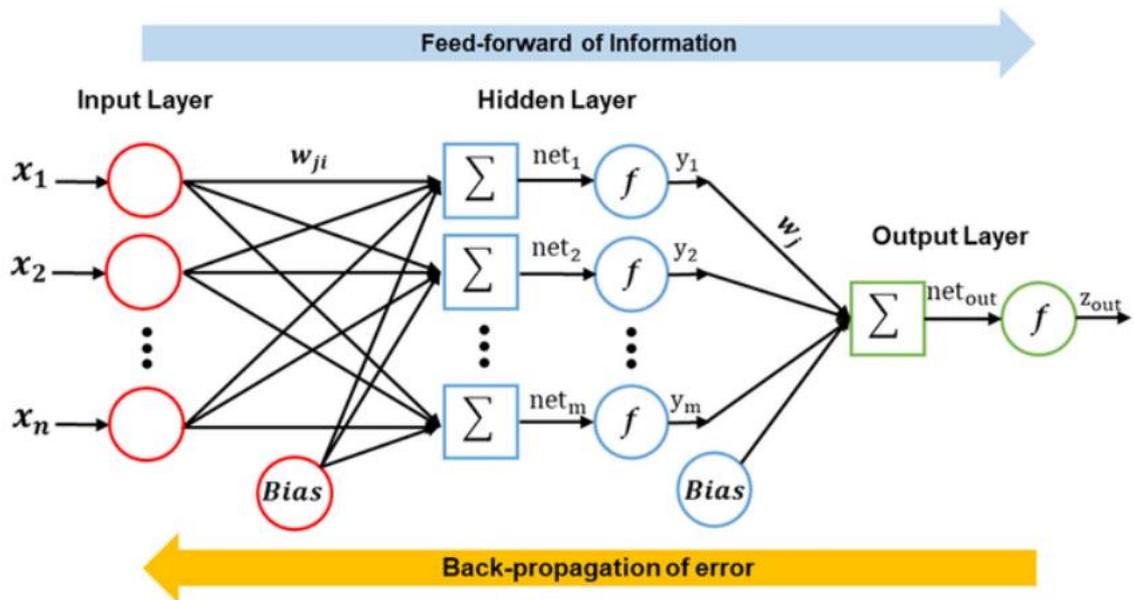


Fig. 3.3 Schema Multi-Layer Perceptron

Fonte [16]

STOCHASTIC GRADIENT DESCENT

L'algoritmo di discesa del gradiente aggiorna continuamente i parametri del modello durante ogni periodo di addestramento per ridurre l'errore. Per aggiornare correttamente tali parametri, prima calcola il gradiente della funzione obiettivo utilizzando la derivata di primo ordine rispetto ai parametri del modello, quindi per minimizzare l'errore aggiorna il parametro nella direzione opposta alla pendenza (gradiente) come mostrato in Fig. Questo processo di aggiornamento dei parametri viene eseguito durante la retro propagazione dell'errore.

ERROR BACK-PROPAGATION

Processo in cui il gradiente su ciascun neurone si propaga nuovamente a tutti i neuroni appartenenti al suo livello precedente.

Consideriamo due neuroni qualsiasi i e j , visibile nella figura seguente (Fig. 3.4):

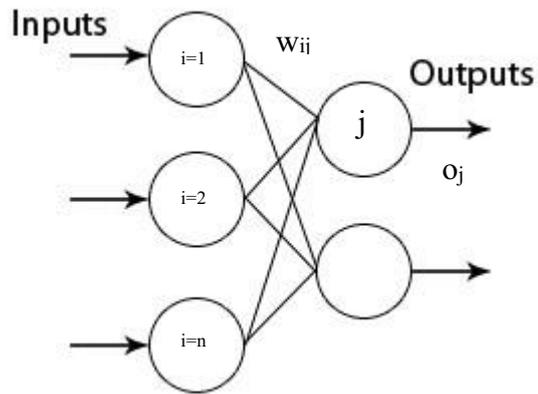


Fig. 3.4 Schema 1

E definiamo

$$net_j \triangleq \sum_{i=1}^n x_i w_{ij}$$

$$\delta_j \triangleq - \frac{\partial E}{\partial net_j}$$

Dove

Applicando la regola della catena ottengo:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}}$$

Vediamo i singoli elementi

$$\frac{\partial E}{\partial o_j} = \frac{\partial}{\partial o_j} \left[\frac{1}{2} \sum (t_j - o_j)^2 \right] = -(t_j - o_j)$$

$$o_j = f(net_j) \Rightarrow \frac{\partial o_j}{\partial net_j} = f'(net_j)$$

$$\frac{\partial net_j}{\partial w_{ij}} = x_i$$

L'equazione precedente quindi risulta

$$\frac{\partial E}{\partial w_{ij}} = -(t_j - o_j) f'(net_j) x_i = -\delta_j x_i$$

Consideriamo adesso uno step in più: aggiungiamo un layer intermedio, come in Fig. 3.5

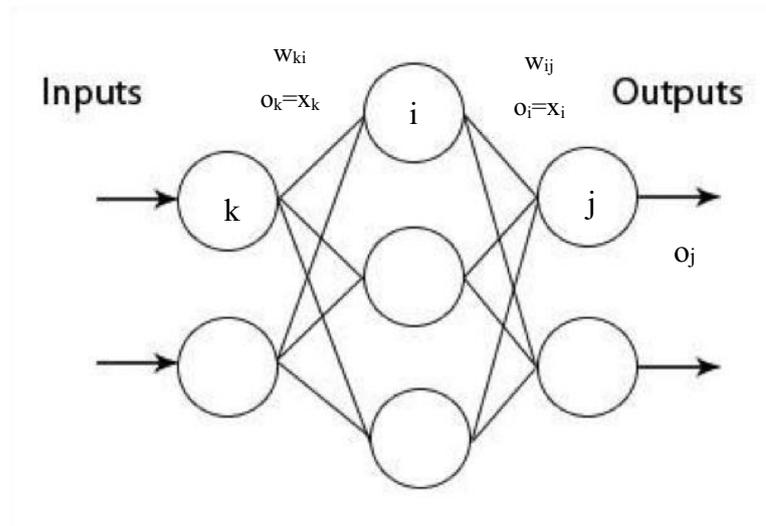


Fig. 3.5 Schema 2

Per l'ultimo layer possiamo scrivere quello che abbiamo scritto nel caso precedente:

$$\frac{\partial E}{\partial w_{ij}} = -(t_j - o_j) f'(net_j) x_i = -\delta_j x_i = -\delta_j o_i$$

Ma per i layer intermedi questo non è possibile in quanto non conosciamo il valore $(t_j - o_j)$; quindi scriviamo:

$$\begin{aligned} \frac{\partial E}{\partial w_{ki}} &= \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial o_i} \frac{\partial o_i}{\partial net_i} \frac{\partial net_i}{\partial w_{ki}} \\ &= -\delta_j w_{ij} f'(net_i) x_k = -\delta_i o_k \end{aligned}$$

Il contributo dato all'errore di un singolo neurone nascosto dipende da tutti i neuroni del layer successivo a cui quel neurone è collegato.

$$\delta_i = f'(net_i) \sum_{s=1}^m \delta_s w_{is}$$

Dove s è l'indice che indica tutti i neuroni del layer nascosto successivo connessi al neurone del layer precedente.

L'idea è quella di calcolare la derivata, ovvero il gradiente, dell'errore rispetto ai pesi e trovare dove la derivata si annulla (ovvero trovare il minimo), quindi aggiornare i pesi seguendo in direzione opposta alla derivata della funzione, come mostrato nella Fig. 3.6.

$$\Delta w_{ij} = \eta \delta_j x_i$$

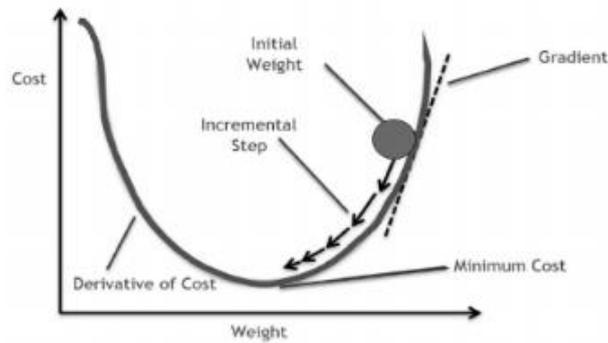


Fig. 3.6 SGD

Fonte [17]

3.3 MATLAB

Il linguaggio scelto è Matlab, questo perché “dispone di strumenti e funzioni per la gestione di grandi set di dati, oltre a toolbox specifici per l’utilizzo delle reti neurali”, inoltre Matlab usa come struttura base le matrici.

Tra le funzioni più utili c’è la possibilità di gestire il controllo della rete, del database e della fase di training in una sola finestra rendendo i vari test estremamente semplici da gestire, dando anche la possibilità di salvare i risultati con i relativi parametri e grafici.

È stata installata l’ultima versione del software: Matlab R2022b sul sistema operativo Windows 10, il pacchetto utilizzato è Deep Learning Toolbox ed è stato installato automaticamente durante la fase di installazione, ma è comunque possibile installarlo in qualsiasi momento.

Durante la fase di addestramento delle reti è stata usata anche l’applicazione “Experiment Manager”, che ha permesso l’addestramento simultaneo di più reti con diversi parametri selezionati precedentemente.

3.4 DESCRIZIONE DELLE RETI USATE NELLA TESI

3.4.1 GOOGLNET

La rete è riportata nell'articolo "Going deeper with convolution" di Szegedy et al. [13], fa uso di "inception module", che permettono di eseguire in parallelo delle analisi con filtri di più dimensioni, in Fig. 3.8 si può vedere la struttura della rete.

È stata usata la versione pre-addestrata standard disponibile su Matlab, modificando esclusivamente il layer "loss3-classifyier" con un "fully connected layer", in modo da poter stabilire il numero di classi per la nostra applicazione, e il "classification layer", per rendere il numero di output pari al numero delle classi.

3.4.2 RESNET50

Questa rete è stata ideata da K. He e descritta nel suo articolo "Deep residual learning for image recognition" [14]. È una rete che usa dei "residual blocks" per evitare il problema del "vanishing gradient", ovvero la scomparsa del gradiente, quando si hanno reti con molti layer invece di aspettare che il gradiente si propaghi indietro (back-propagation) un livello alla volta, viene creato un percorso di skip connection, il cui schema è visibile in Fig. 3.7, che consente di raggiungere i nodi iniziali saltando quelli intermedi.

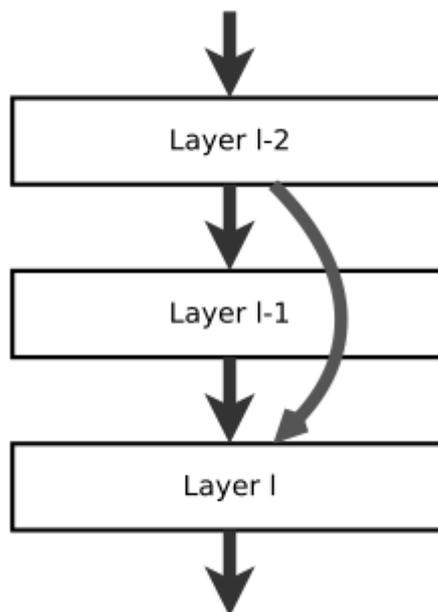


Fig. 3.7 Skip Connections

Fonte [18]

Questa rete, insieme alle altre reti ResNet, sono quelle principalmente usate per la classificazione di immagini mediche [15]

È stata usata la versione pre-addestrata standard disponibile su Matlab, modificando esclusivamente il layer “fc1000” con un “fully connected layer”, in modo da poter stabilire il numero di classi per la nostra applicazione, e il “classification layer”, per rendere il numero di output pari al numero delle classi. Lo schema è visibile in FIG. 3.8

PROBLEMI: VANISHING GRADIENT ED EXPLODING GRADIENT

Quello della scomparsa del gradiente (in inglese “vanishing gradient”) è un problema che causa notevoli difficoltà durante l'addestramento di una CNN, nello specifico coinvolge i pesi nei primi strati della rete neurale.

Il problema sorge durante l’error back-propagation, che calcola il gradiente rispetto a un particolare peso e poi usa questo valore per aggiornare il peso stesso (come visto nel paragrafo riguardante l’Error Back-Propagation), se il gradiente è troppo piccolo allora questo aggiornamento sarà irrilevante. Mano a mano che continuiamo a muoverci all’indietro nella rete, questi gradienti diventeranno sempre più piccoli, di conseguenza, i pesi di questi strati riceveranno un aggiornamento sempre più piccolo, imparando quindi molto lentamente e in modo inefficiente.

Questo problema può verificarsi anche in senso opposto: non un gradiente che svanisce ma piuttosto un gradiente che esplode (in inglese “exploding gradient”): ciò che abbiamo appena detto riguardo al rimpicciolirsi del gradiente, vale anche nel caso in cui il gradiente sia grande, dando però il risultato opposto, ovvero quello di ottenere uno spostamento tale che forse il valore ottimale non sarà mai raggiunto perché la proporzione a cui il peso viene aggiornato ad ogni epoca è troppo grande.

POSSIBILI SOLUZIONI

Le soluzioni non sono molte, ma sono efficaci:

1. Possiamo usare funzioni attivazione come ReLU nell'architettura CNN, che aiuta a evitare questo problema in quanto il valore della derivata parziale avrà solo valori 0 e 1 prevenendo la scomparsa del gradiente. Con questa funzione c’è però il problema legato al caso in cui il gradiente sia 0, che può tuttavia essere risolto molto facilmente utilizzando delle varianti come leaky ReLU e noisy ReLU.
2. L’inizializzazione dei pesi, come ad esempio l’inizializzazione di Xavier; questa tecnica non garantirà la risoluzione del problema, ma renderà la scomparsa del gradiente più difficile.

3. Cambiare architettura della rete, come quella delle reti residuali (ad esempio ResNet), che hanno strutture che collegano diversi strati tra di loro (chiamate skip connections). Questo tipo di connessioni agisce come una scorciatoia che consente al gradiente di risalire più velocemente.
4. La normalizzazione dei batch può rendere la rete più robusta al fenomeno del vanishing/exploding gradient.

3.4.3 *MOBILENET V2*

Questa rete è stata selezionata in quanto progettata per essere eseguita su dispositivi con una capacità computazionale ridotta (principalmente dispositivi mobili e sistemi embedded), l'idea è quindi quella di ottimizzare la velocità e la portabilità delle operazioni svolte con questa rete; questo tuttavia compromette l'accuratezza generale [16].

Anche per questa rete è stata usata la versione pre-addestrata standard disponibile su Matlab, modificando esclusivamente il layer "logits" con un "fully connected layer", in modo da poter stabilire il numero di classi per la nostra applicazione, e il "classification layer", per rendere il numero di output pari al numero delle classi.

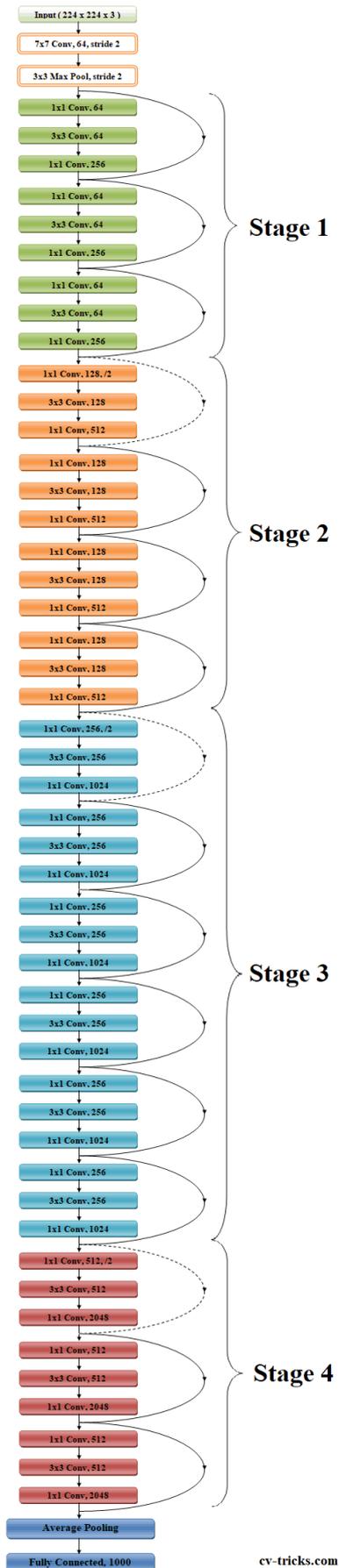


Fig. 3.9 Schema ResNet50
Fonte [19]

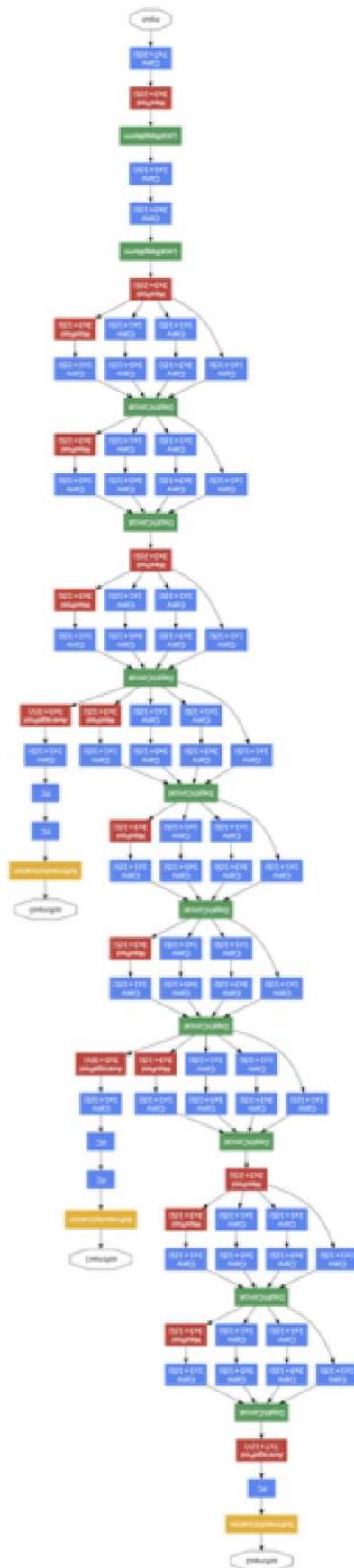


Fig. 3.8 Schema GoogLeNet
Fonte [20]

3.4.4 TRANSFER LEARNING

La Fig. 3.10, rappresenta uno schema della tecnica usata per l'addestramento di queste reti, questa è chiamata transfer learning e prevede il "riciclo" delle informazioni e competenze acquisite da un modello precedentemente addestrato, per farlo si sostituiscono il database in input e gli ultimi layer della rete (ovvero quelli che imparano le caratteristiche più complesse) [17].

Quindi, senza dover ripartire da zero, si può eseguire un ri-addestramento della rete neurale.

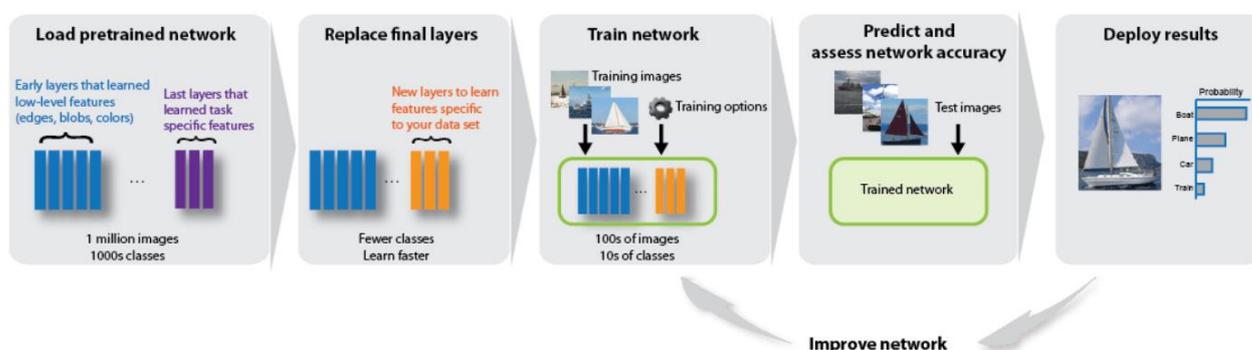


Fig. 3.10 Schema Transfer Learning

Fonte [21]

I vantaggi di questo tipo di applicazione sono:

- Maggiore accuratezza nella maggior parte dei casi
- Tempo di training ridotto
- È un estrattore di caratteristiche molto potente

3.5 DESCRIZIONE DEGLI STEP SVOLTI

Con le reti appena descritte sono stati eseguiti diversi test:

1. Per tutti e tre i database sono state svolte delle sessioni di training (usando il training set e il validation set) variando gli iperparametri: learning rate, momentum, max epoch (numero massimo di epoche) e mini batch size; per quanto riguarda i primi due iperparametri, dato il loro ruolo molto importante, è fornita una descrizione in Appendice 1. Le reti così addestrate sono state salvate per eseguire una valutazione successivamente sul test set.

- Solamente per il database principale sono state eseguite delle sessioni di addestramento sulla versione normalizzata del database originale. Anche in questo caso le reti addestrate sono state salvate per valutarne le prestazioni successivamente sul test set.

In Fig. 3.11 si può vedere un esempio di rete in fase di addestramento.

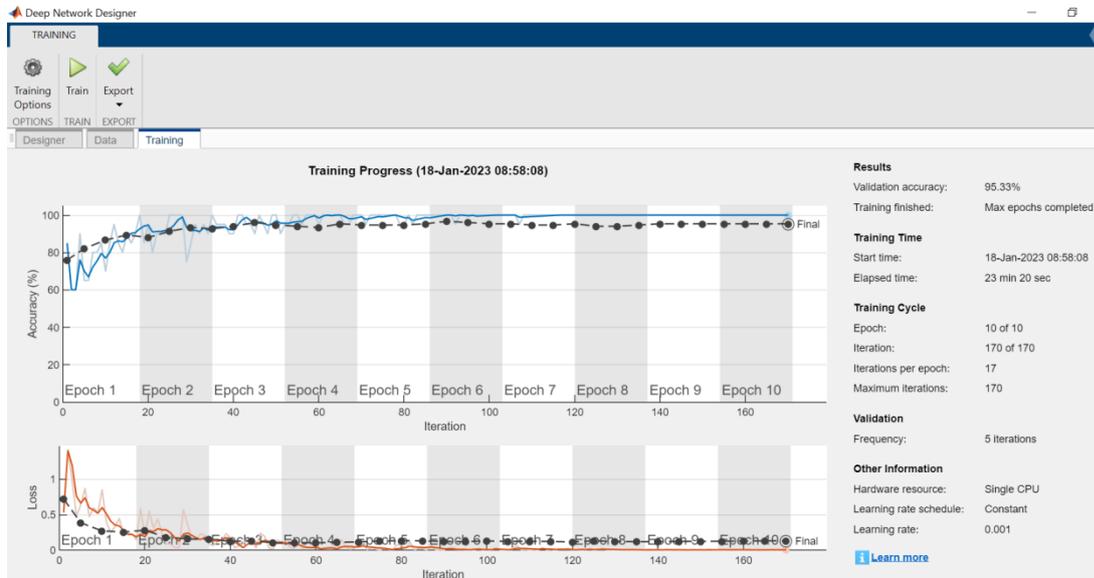


Fig. 3.11 Rete In Fase Di Addestramento

Alla fine di questa serie di test sono state valutate le prestazioni sui test set e prodotte le rispettive confusion matrix con valore di accuracy, come già discusso nel paragrafo 2.5.

Solamente per il database principale, sia nella versione originale che in quella normalizzata, sono poi state valutate le curve ROC e il valore di AUC-ROC. Questa scelta è dovuta al fatto che i due database secondari hanno dimostrato prestazioni ottime, non richiedendo quindi ulteriori approfondimenti, mentre il database principale necessitava di maggiori approfondimenti in quanto quello con il compito più complesso.

4 RISULTATI

4.1 RISULTATI OTTENUTI SUL DATABASE PER VALUTARE L'ACCETTABILITÀ DELLE IMMAGINI

Per quanto riguarda il primo step, ovvero la valutazione della qualità dell'immagine i risultati migliori sono stati ottenuti con la rete ResNet50 e con i parametri:

- Learning Rate = 0,001
- Momentum = 0.9
- Mini Batch Size = 20
- Max Epoch = 8

Nell'immagine seguente (Fig. 4.1) è possibile vedere un esempio di confusion matrix ottenuta per i modelli addestrati con ResNet50 sul database per la valutazione dell'accettabilità. Dalle due "tabelle" si può vedere come il valore di sensitività e specificity siano nettamente migliori nel modello con LR = 0,001 (e con tutti i parametri menzionati sopra), mentre in quella corrispondente al valore di LR = 0,01 il modello sbaglia con più frequenza a classificare le immagini.

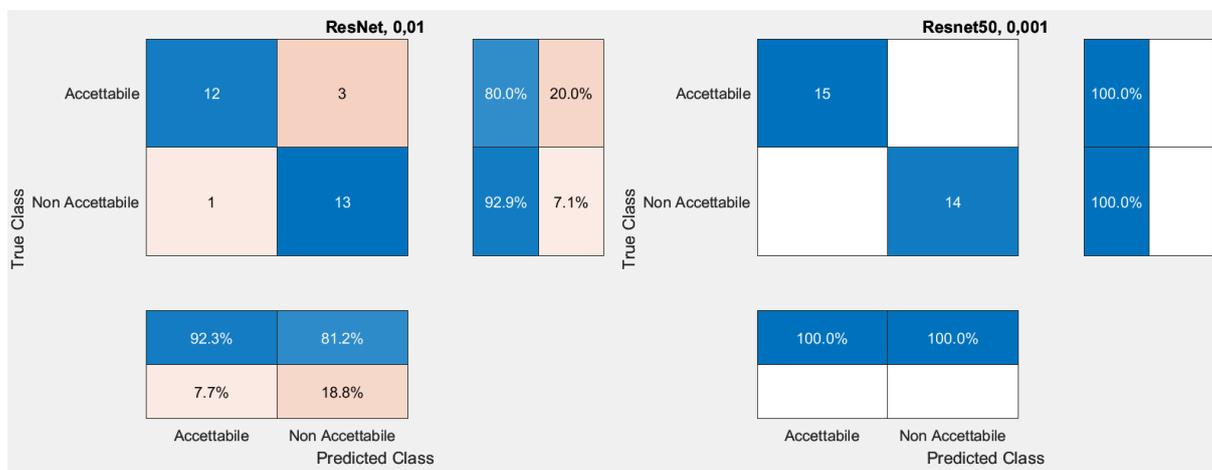


Fig. 4.1 Confronto Modelli Resnet50

Nelle due immagini in Fig. 4.2 si può vedere un esempio di classificazione (con percentuale relativa alla probabilità con cui le immagini sono state assegnate alla classe indicata) delle immagini contenute all'interno del database, la prima è nitida e la MT si vede in modo chiaro, mentre la seconda è sfocata e la MT è parzialmente nascosta, non adatta quindi per effettuare una diagnosi (nemmeno da un medico).

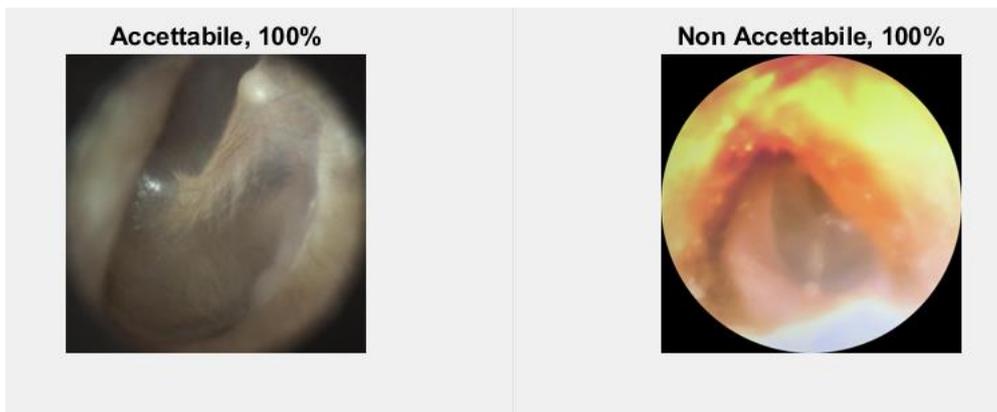


Fig. 4.2 Esempio Di Classificazione

4.2 RISULTATI OTTENUTI SUL DATABASE PER IL RICONOSCIMENTO DI MT DESTRA E SINISTRA

Nel secondo step che riguardava il riconoscimento delle MT tra destra e sinistra tutte le reti testate hanno dato risultati ottimi, riportiamo un esempio con ResNet50, ma le stesse prestazioni sono state raggiunte con tutte e tre le reti proposte. ResNet50 è quella che a parità di epoche (8) raggiungeva prima valori di accuracy sul validation test migliori; gli altri parametri impostati sono:

- LR = 0,001
- Momentum = 0,9
- Mini Batch Size = 10;
- Max Epochs = 8

Nella Fig. 4.4 è stata riportata solo la confusion matrix della rete appena citata come esempio, la maggior parte delle confusion matrix ottenute con le diverse reti addestrate su questo database (quindi anche MobileNetv2, che può avere dei vantaggi per l'utilizzo su dispositivi come gli smartphone), indipendentemente dai parametri scelti.

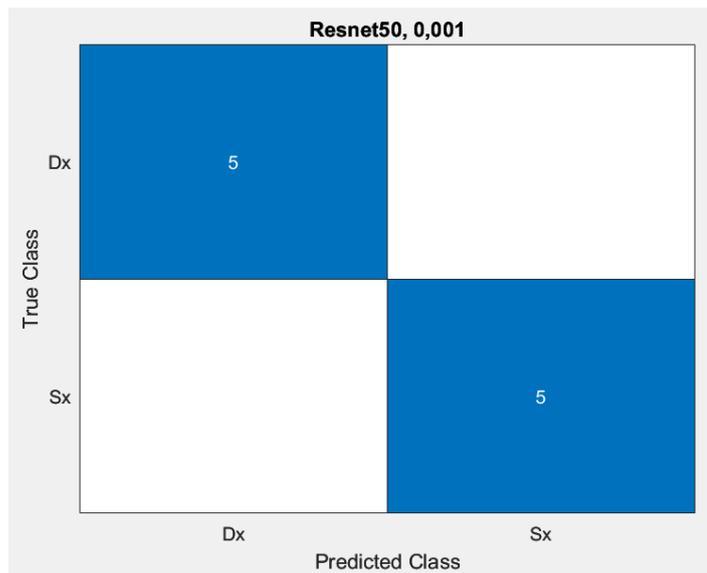


Fig. 4.3 confusion Matrix Per Riconoscimento Destra E Sinistra

Anche per questa rete sono stati riportati degli esempi di classificazione, visibili in Fig. 4.4 (anche in questo caso sono riportate, oltre alle classi scelte, le probabilità con e quali le immagini sono state assegnate a quelle classi).

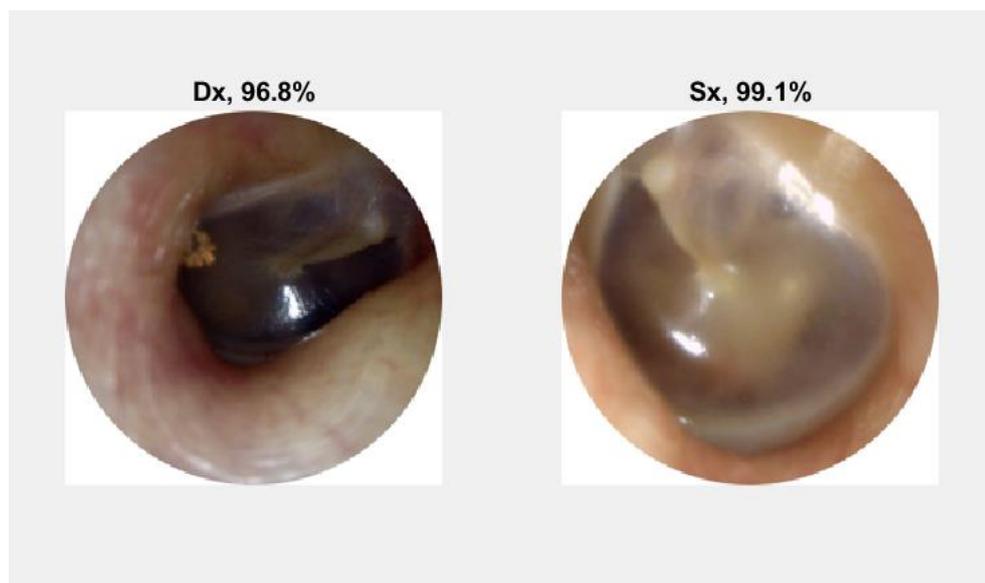


Fig. 4.4 Esempio Di Classificazione Destra E Sinistra

4.3 RISULTATI OTTENUTI SUL DATABASE PRINCIPALE

Questo è il database che ha richiesto più lavoro; cominciamo con l'analizzare le prestazioni nei diversi scenari rappresentati nella serie di confusion matrix di seguito, in Fig. 4.5, Fig. 4.6 e Fig. 4.7:

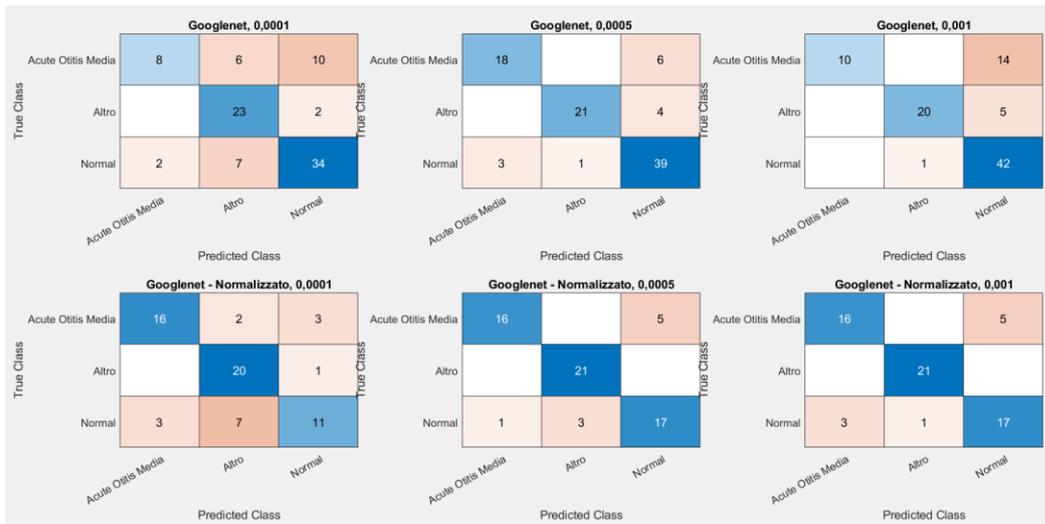


Fig. 4.5 Confronto Confusion Matrix Per GoogLeNet

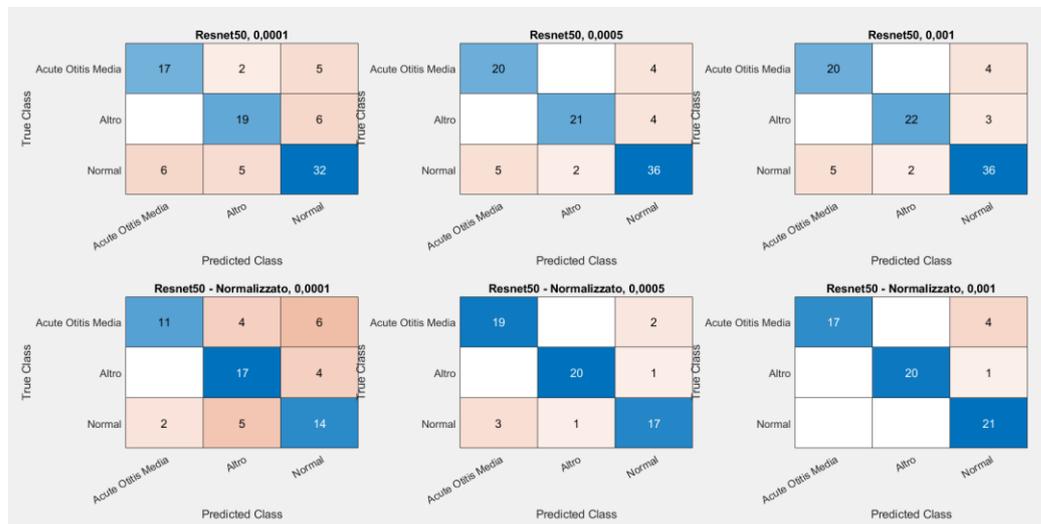


Fig. 4.6 Confronto Confusion Matrix ResNet50

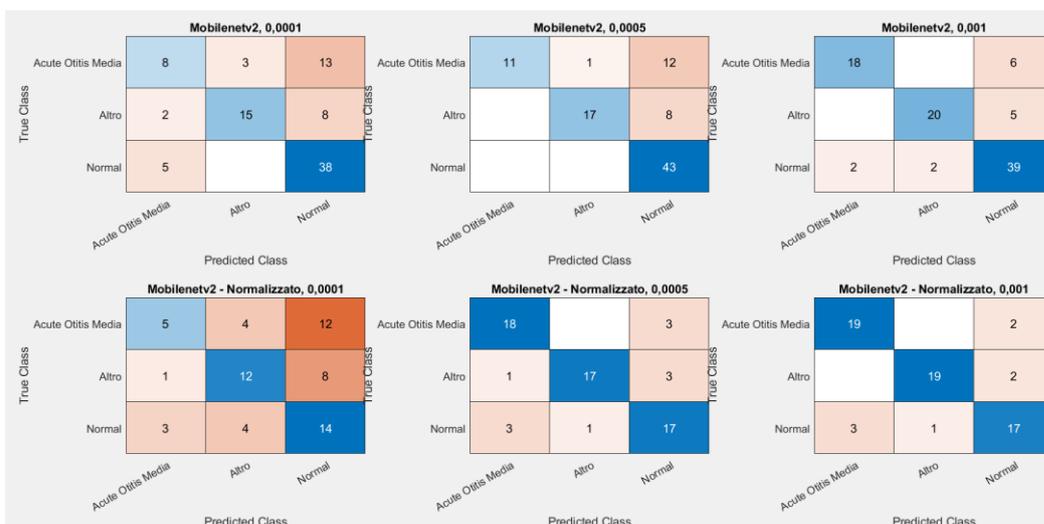


Fig. 4.7 Confronto Confusion Matrix Per MobileNetv2

L'accuracy migliore si ha con ResNet50 con iperparametri impostati a:

- LR = 0,001
- Momentum = 0,9
- Mini Batch Size = 20
- Max Epochs = 10

Questo vale sia per il database originale (84%) sia per quello normalizzato (92%); sempre dalle confusion matrix in Fig. 4.7 si può vedere come in generale le prestazioni migliori si hanno con LR = 0,001 con tutte le reti (GoogLeNet, ResNet50 e MobileNetv2) e, soprattutto per le confusion matrix relative alle prove eseguite con la rete MobileNetv2 la normalizzazione sia un fattore chiave per avere prestazioni migliori.

Eseguendo altri test con la rete normalizzata, ma variando il test set (che veniva scelto in modo casuale ogni volta) sono stati trovati anche valori di accuracy migliori, come riportato nella tabella seguente Tab. 1:

Trial	Net	LR	Accuracy	AUC-AOM	AUC-ALTRO	AUC-NORM
1	Resnet50	0,001	0,9783	0,9994	1,0000	0,9995
2	Resnet50	0,001	0,9239	0,9865	0,9994	0,9843
3	Resnet50	0,001	0,9348	0,9969	1,0000	0,9972
4	Resnet50	0,001	0,9457	0,9877	0,9994	0,9905
5	Resnet50	0,001	0,9783	0,9994	0,9988	0,9986

Tab. 1 Valori Accuracy E AUC-ROC Per ResNet50 Normalizzata

Dalla tabella è possibile vedere che, nonostante i cambiamenti dei test set le prestazioni restino molto alte, superiori al 92%.

Per quanto riguarda le metriche più significative per questo tipo di analisi vediamo subito che i risultati trovati precedentemente (visibili in Tab. 1) trovano conferma anche in questi grafici che riportano le curve ROC dei vari test, ma soprattutto per quanto riguarda i valori di AUC-ROC, come si può vedere nei grafici visibili nelle Fig. 4.8, Fig. 4.9 e Fig. 4.10.

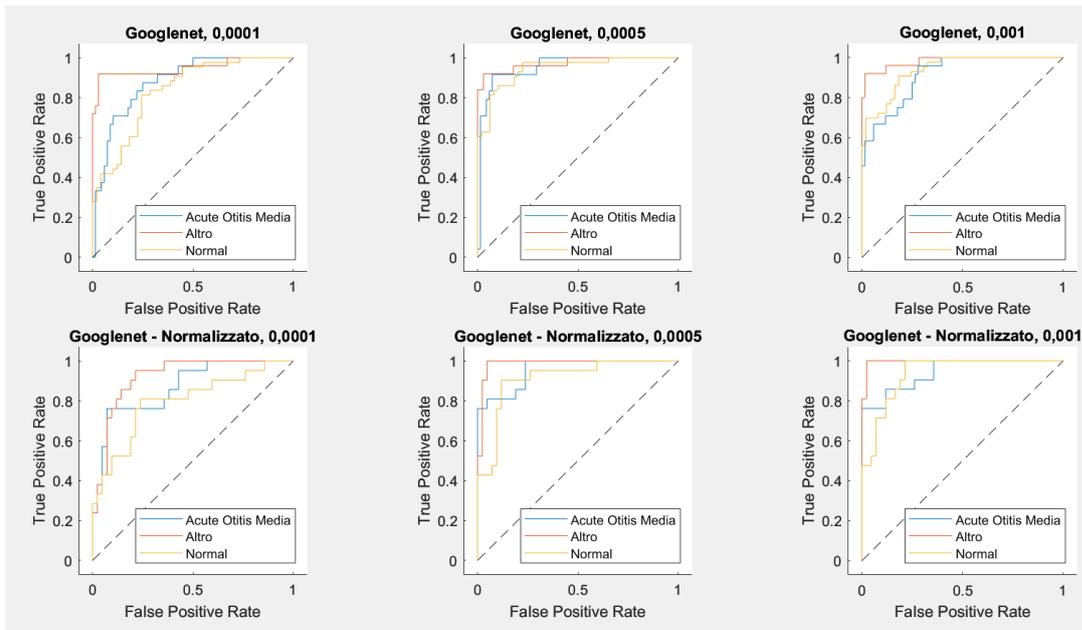


Fig. 4.8 Confronto ROC Per GoogLeNet

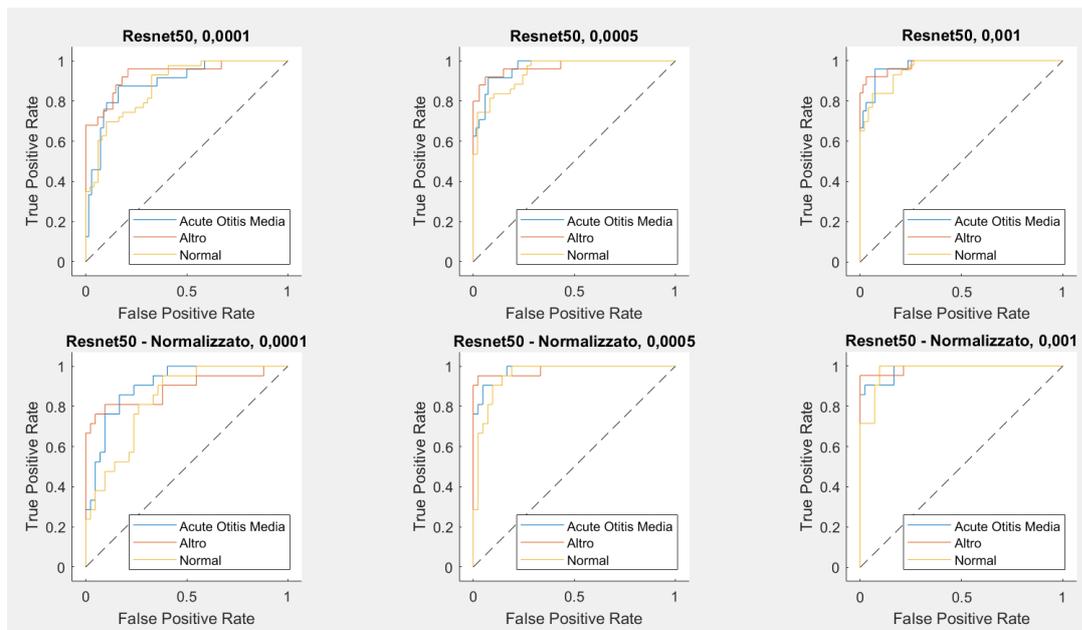


Fig. 4.9 Confronto ROC Per ResNet50

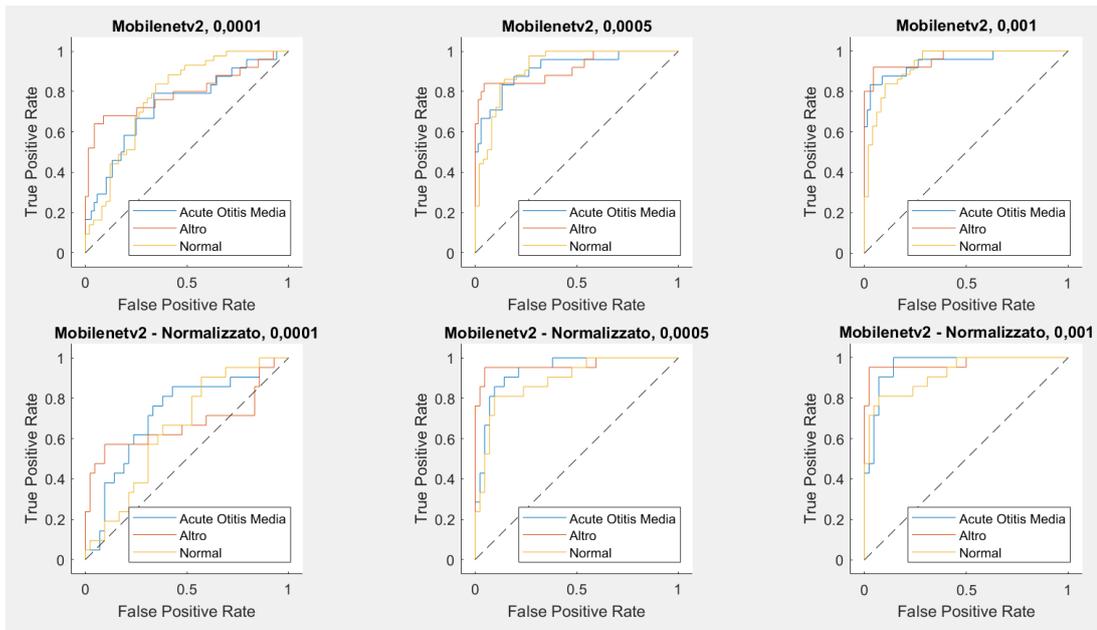


Fig. 4.10 Confronto ROC Per MobileNetv2

Secondo quanto visto nel paragrafo dedicato ai criteri di valutazione (2.5 Criteri di Valutazione) possiamo vedere subito come le curve che si avvicinano maggiormente al valore ideale (come mostrato nella Fig. 2.13) sono quelle dei modelli addestrati con valori di LR = 0,001, confermando quanto trovato osservando le confusion matrix (Fig. 4.5, Fig. 4.6 e Fig. 4.7); inoltre a parità di LR, il modello normalizzato sembra migliore in generale

Osserviamo adesso i valori di AUC-ROC, rappresentati nelle Fig. 4.11, Fig. 4.12 e Fig. 4.13:

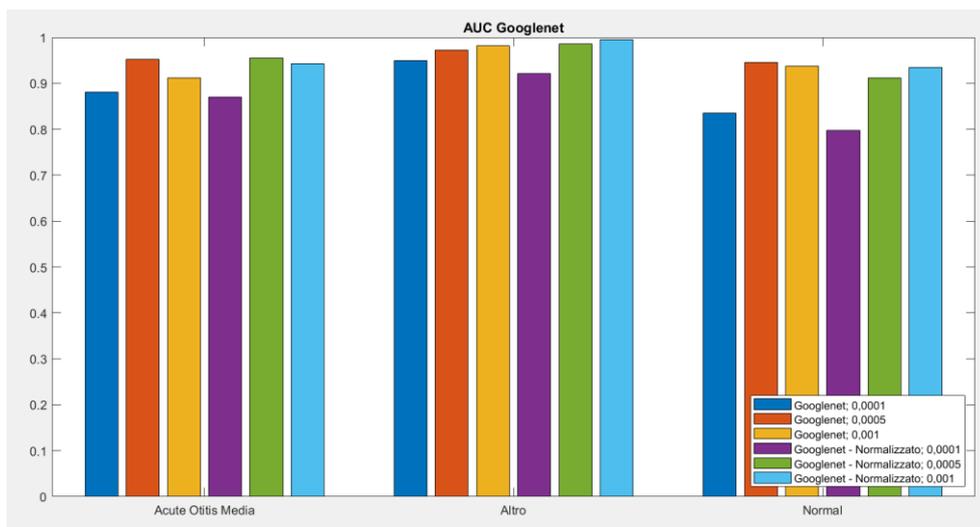


Fig. 4.11 Confronto AUC-ROC Per GoogLeNet

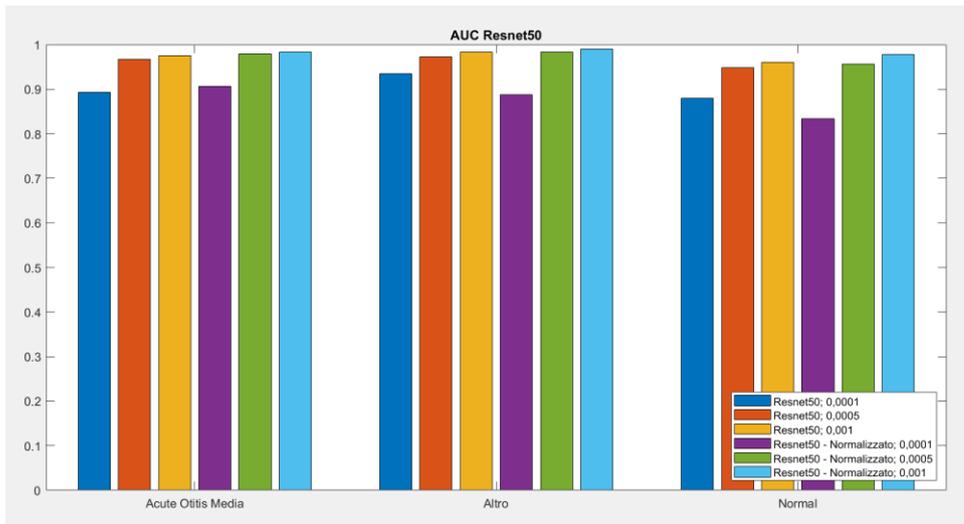


Fig. 4.12 Confronto AUC-ROC Per ResNet50

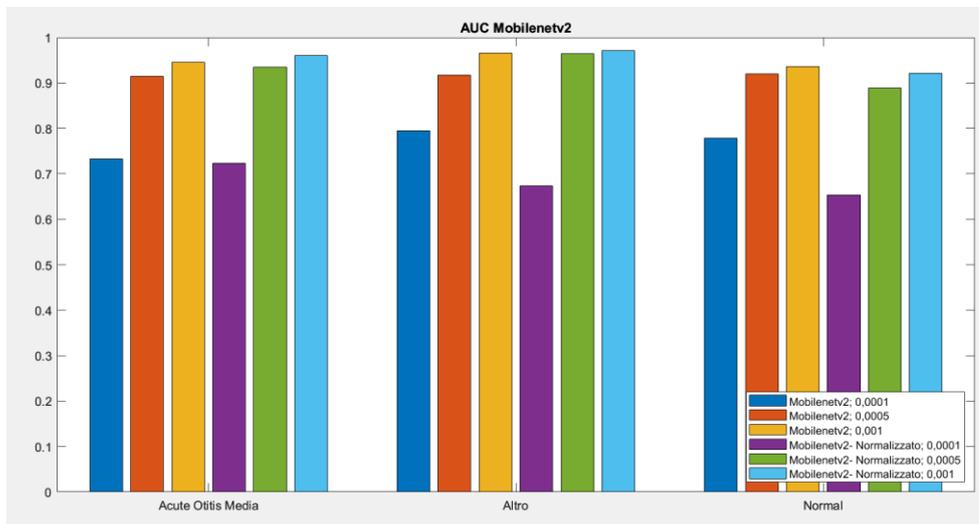


Fig. 4.13 Confronto AUC-ROC Per MobileNetv2

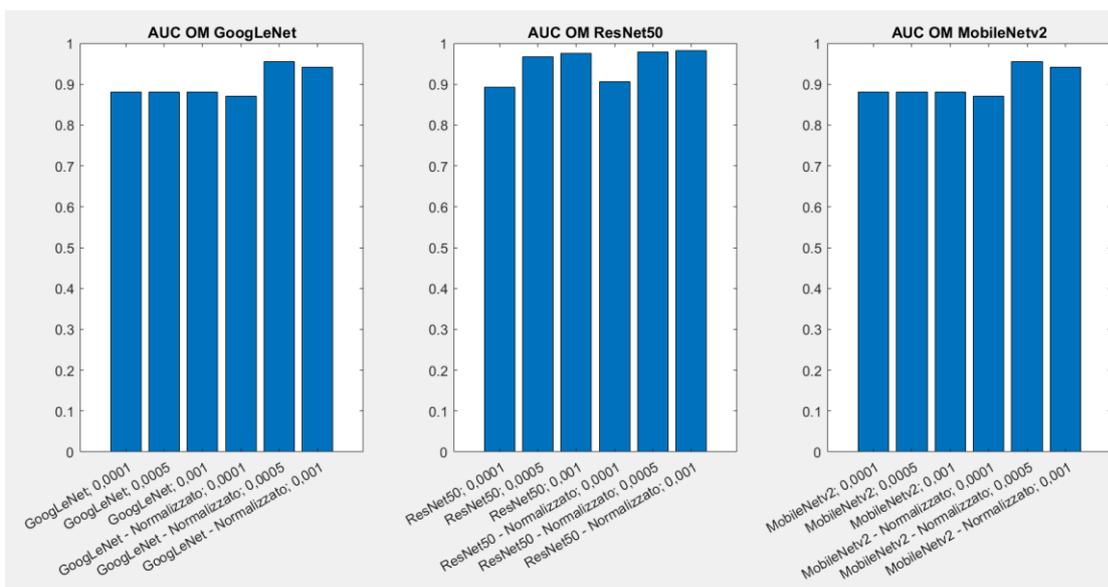


Fig. 4.14 Confronto AUC Otite Media Per Le Diverse Reti

Da questi grafici a barre possiamo vedere come in generale i modelli addestrati sui database normalizzati performino in modo migliore rispetto a quelli addestrati sul database non normalizzato, confermando quindi quanto trovato precedentemente.

In accordo con tutti i risultati trovati finora possiamo dire che il modello migliore è quello di ResNet50 con database normalizzato e con iperparametri:

- LR = 0,001
- Momentum = 0,9
- Mini Batch Size = 20
- Max Epochs = 10

Come menzionato prima, da altri test fatti con la stessa rete i valori trovati sono molto alti per quanto riguarda il valore di AUC della categoria otite media (visibile nella colonna AUC AOM nella Tab. 1).

4.4 PROBLEMA DELL'INTERPRETABILITÀ

Quello dell'interpretabilità è un problema ancora non risolto per le CNN, in quanto il processo decisionale è completamente nascosto all'utente nella fase di estrazione delle caratteristiche.

Per questo motivo è stata impiegata la funzione di Matlab gradCam, che permette di evidenziare con una heat map le caratteristiche che il network ha selezionato come più rilevanti; questo ci permette anche di valutare se le aree evidenziate sono le stesse che verrebbero usate dai medici durante la valutazione clinica.

In seguito è possibile vedere alcuni esempi di:

- Immagini con classificazione e rappresentazione grafica della probabilità di appartenere alla classe selezionata (Fig. 4.15 Fig. 4.18 Fig. 4.20)
- Le stesse immagini con a fianco la sovrapposizione della heat map che evidenzia le aree rilevanti

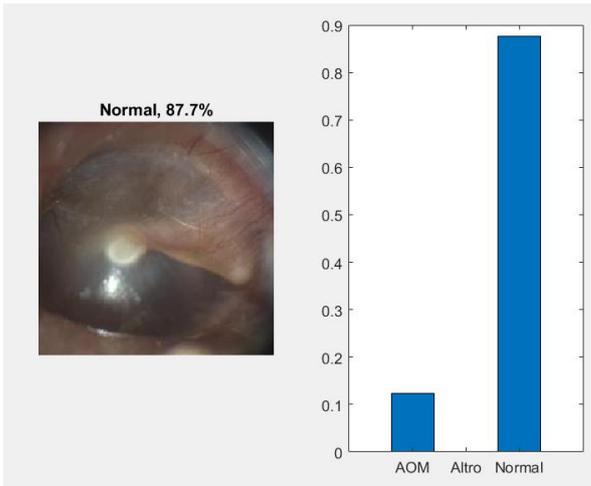


Fig. 4.15 Esempio Predizione Normale

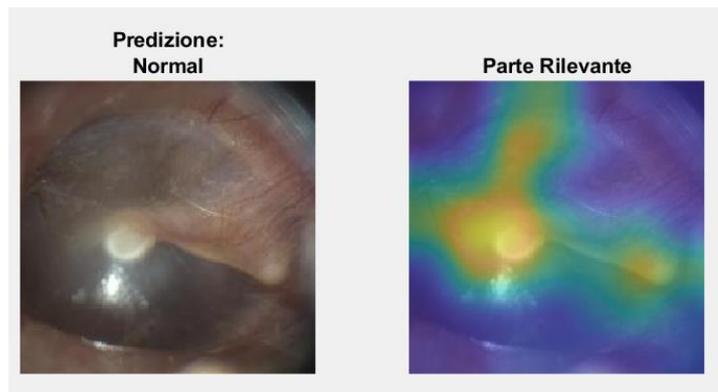


Fig. 4.16 Esempio GradCam Normale

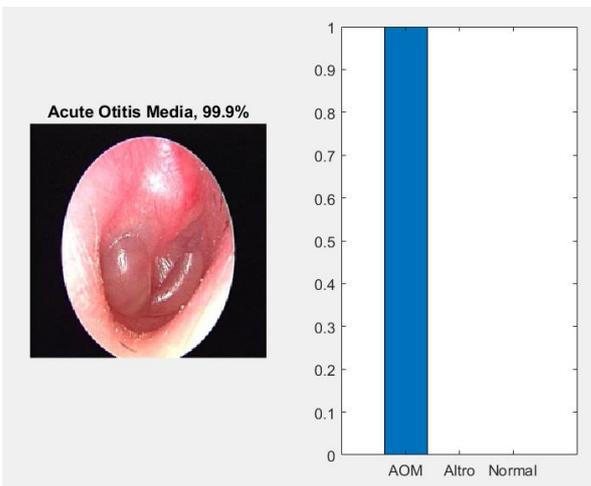


Fig. 4.18 Esempio Predizione Otite Media

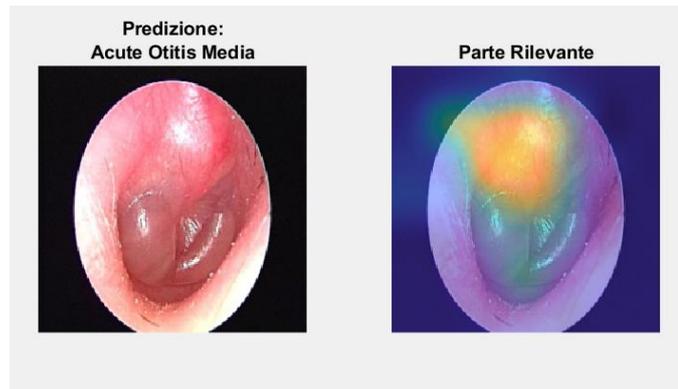


Fig. 4.17 Esempio GradCam Otite Media

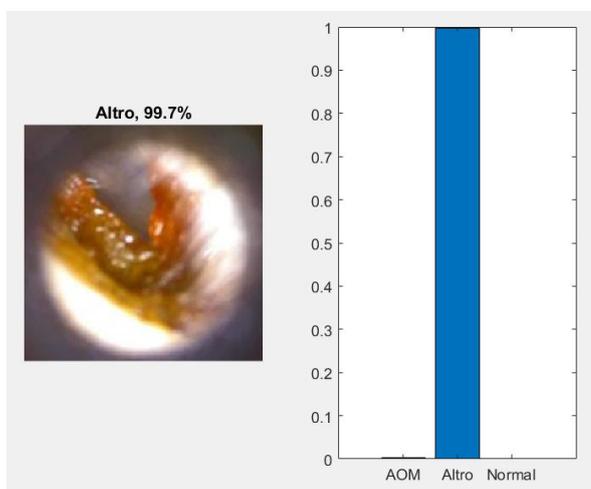


Fig. 4.20 Esempio Predizione Altro

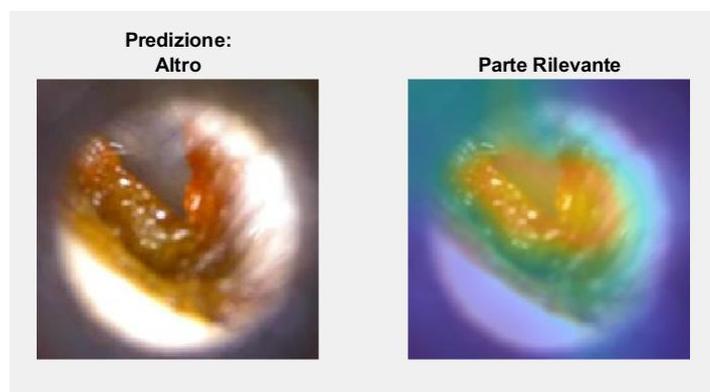


Fig. 4.19 Esempio GradCam Altro

Come è possibile vedere, non tutte le parti selezionate corrispondono effettivamente alle stesse caratteristiche che ricercerebbe il medico, tuttavia non bisogna dimenticare che la rete non ha ricevuto nessuna indicazione pratica sulle caratteristiche della patologia e su come questa si manifesti. Con un database sempre più ricco questo aspetto potrà sicuramente migliorare.

4.5 DISCUSSIONE

Questo studio dimostra la fattibilità di un approccio di machine learning per il riconoscimento dell'otite media tramite l'uso di uno strumento low cost, le cui immagini sono state usate sia per addestrare, sia per testare le diverse reti neurali.

I risultati raggiunti per quanto riguarda il riconoscimento dell'otite media sono in linea con quelli trovati nei precedenti studi presentati da Cha et al [18] [19], in generale possiamo vedere, come previsto dalla teoria, che il database normalizzato performa generalmente meglio rispetto a quello con le classi sbilanciate.

Rispetto ad altri studi precedenti [18], [19], [20], in cui sono state usate solo immagini raccolte con un solo strumento per ottenere i risultati migliori, noi abbiamo creato un database caratterizzato proprio da immagini raccolte con strumenti diversi e abbiamo comunque ottenuto ottimi parametri con valori di accuracy fino a 98% e valori di AUC per la classe dell'otite media fino a 99% (come visibile dai grafici e dalla tabella riportati nei capitoli di risultati); questo è utile anche perché rende il modello più robusto a eventuali variazioni nella scelta dello strumento qualora in futuro lo strumento usato da noi non fosse più disponibile.

Rispetto ad altri studi come quelli di Cha et al [18] e Livingstone [19] inoltre il numero di classi da riconoscere è molto più basso (il primo valuta 6 classi e il secondo 14), è possibile che rispetto ai valori di accuracy raggiunti in questi studi i nostri siano migliori perché il numero di classi da riconoscere è più piccolo.

Uno dei limiti maggiori riscontrati è legato al database: la dimensione del nostro database infatti è estremamente inferiore a quello degli studi già citati precedentemente, sicuramente aumentando il numero di immagini nel database l'accuracy generale potrebbe migliorare di molto.

Quindi possiamo dire che il più grande problema riscontrato è l'assenza di un database che fosse sufficientemente esteso per poter eseguire delle analisi complete e presentare

un'adeguata varietà di casi: in uno dei database utilizzati [8] c'erano centinaia di immagini, ma molte di esse erano semplicemente più foto dello stesso caso mentre altre sono state da noi definite inaccettabili: visione della MT molto ridotta o in alcuni casi addirittura MT non visibile [21]. Per questo motivo è stato scelto di eseguire una selezione molto restrittiva sui database già disponibili, in modo che questi non contenessero immagini ambigue che, di fatto, non verrebbero usate in ambito clinico per effettuare una diagnosi; il lato negativo di questa selezione è che generalmente non è mai stato possibile raggiungere un'accuratezza del 100%, ma andando a guardare le metriche più significative per quanto riguarda i modelli di classificazione, ovvero i valori di AUC-ROC, per la categoria OM non sono mai scesi sotto l'84% (nel caso peggiore, non riportato in tabella).

Il fatto di aver analizzato più CNN ci ha poi permesso di valutare in modo più completo le potenzialità di questo metodo, cosa che in altri studi precedentemente pubblicati non era mai stata fatta.

Nonostante i dati presentati lascino intendere la fattibilità di una tale applicazione, è necessario menzionare che ci sono dei limiti non trascurabili nei quali si è incorsi durante lo studio e che sarà necessario migliorare in futuro, prima di rendere questa idea un'applicazione effettivamente utilizzabile.

In primis è necessario menzionare la limitazione più grande al momento, ovvero la natura dello strumento, che è stato adattato per i nostri scopi e ha dato sicuramente buoni risultati, ma che tuttavia non è commercializzato per lo scopo per cui è stato utilizzato per questo studio. Questo significa che prima di poter pensare a un'applicazione sarà necessario valutare i dispositivi disponibili in commercio che rispettino le norme UE, oppure valutare la progettazione di uno strumento che possa essere approvato per l'utilizzo clinico.

Va inoltre considerato che lo studio è stato svolto con strumenti che si adattano alle orecchie degli adulti, ma ne va verificata l'applicabilità anche nei bambini, categoria per la quale questo tipo di applicazione potrebbe risultare più utile, come visto nei primi capitoli.

Un altro limite riscontrato è sicuramente l'assenza di una definizione chiara di Otite Media, purtroppo ogni fonte definisce questa patologia in modo leggermente diverso, rendendo la produzione di una diagnosi univoca molto difficile.

Nonostante i limiti presentati i vantaggi che questo approccio potrebbe portare in futuro sono evidenti, in quanto rende possibile, con una buona precisione, la diagnosi di una patologia estremamente comune e sarebbe applicabile nei luoghi in cui questa è più diffusa, come asili e

scuole, ma anche centri sportivi con piscine e nuclei familiari con bambini con particolari predisposizioni. Ad oggi non si può pensare che questo tipo di applicazione possa sostituire il medico professionista ma può fornire un supporto direttamente al paziente, ma anche al medico che può usarlo come confronto per prendere una decisione in caso di dubbio.

5 CONCLUSIONI E SVILUPPI FUTURI

Le diagnosi di otite media può risultare complicata per diversi motivi come la mancanza di medici specializzati (secondo i dati Istat in Italia c'è meno di uno specialista ORL ogni 10.000 abitanti [22]), decisioni e prescrizioni fatte da medici di base, necessità di strumenti costosi e grande variabilità delle definizioni della patologia stessa e database limitati.

Sul mercato inoltre sono disponibili diversi strumenti professionali che tuttavia comportano spese ingenti e necessità di PC per gestirne l'interfaccia, mentre per quanto riguarda gli strumenti low cost la scelta è più limitata (inesistente se consideriamo che nessuno di questi è disponibile con marcatura CE e quindi non utilizzabili in ambito clinico) e le prestazioni ad ora sono nettamente inferiori rispetto a quelle degli otoscopi professionali.

Lo scopo della tesi era dimostrare come, unendo le performance dell'Intelligenza Artificiale con uno strumento che risulta essere molto più abbordabile dal punto di vista economico, si potesse ottenere una rete addestrata con delle prestazioni sufficientemente alte da poter essere presa in considerazione per una futura applicazione in ambito clinico e di screening.

Abbiamo considerato dati raccolti con entrambe le categorie di strumenti, questo perché ci sarebbe servito un numero di immagini raccolte solo con otoscopi low cost ancora non disponibile, ma era comunque necessario avere parte delle immagini che potessero confermare che la rete fosse in grado di riconoscere anche le immagini raccolte con il dispositivo low cost.

Abbiamo quindi testato vari approcci utilizzando diversi tipi di database (normalizzati e non) e diverse reti (GoogLeNet, ResNet50 e MobileNetv2) con caratteristiche diverse tra loro in modo da valutare quale dei diversi approcci funzionasse meglio per la nostra applicazione e infine abbiamo osservato i valori di accuracy, sensitivity, specificity, ROC e AUC-ROC per poter verificare la validità di questo tipo di applicazione.

I risultati sono positivi in quanto tutti portano verso una conclusione coerente per questo esperimento, ovvero l'impiego di database normalizzati e con la rete ResNet50 (LR = 0,001) forniscono le prestazioni migliori (accuracy = x, AUC-ROC per OM = y). I risultati inoltre sono coerenti anche con gli altri studi già proposti sullo stesso argomento, dando un'ulteriore conferma sulla potenzialità di questo tipo di approccio.

In sintesi si può vedere come molti dei problemi precedentemente menzionati potrebbero essere evitati utilizzando un modello di riconoscimento automatico con CNN e con strumenti

dal costo estremamente contenuto come quello presentato in questa tesi; la combinazione di queste due caratteristiche è sicuramente promettente per il futuro e potrebbe portare a un miglioramento nella cura del paziente con modelli ancora più efficaci nella predizione di patologie dell'orecchio, non solo in ambito clinico, ma dando la possibilità anche a persone esterne all'ambito clinico di eseguire uno screening, come gli asili (in cui patologie come l'OM sono molto diffuse), ma anche per l'utilizzo in paesi in cui trovare personale medico specializzato può risultare molto complesso.

Gli sviluppi futuri di questo lavoro potrebbero riguardare un lavoro di raccolta di immagini solo con dispositivi low cost, questo renderebbe le reti migliori nel riconoscimento di patologie dell'orecchio su immagini con risoluzione inferiore a quelle raccolte con strumenti professionali. Una volta migliorato il database sarebbe necessario valutare le prestazioni utilizzando reti più adatte all'utilizzo su smartphone (come MobileNetv2, che ha dimostrato comunque buone prestazioni per i due compiti più semplici di questa tesi), per essere coerenti con una delle idee alla base di questa tesi, ovvero mantenere i costi contenuti (uno smartphone, generalmente, costa meno di un PC).

Ulteriori step successivi potrebbero essere il test di un prototipo di app sul campo e lo sviluppo di uno strumento con costo contenuto ma con prestazioni migliori per quanto riguarda la qualità dell'immagine.

6 APPENDICE 1: CONCETTI BASE DELLE RETI NEURALI

6.1 NEURONI “REALI” E NEURONI ARTIFICIALI

In biologia, il termine “rete neurale” indica un modello matematico che identifica la rete formata dalle cellule nervose, dette neuroni, che sono interconnesse tra di loro e che formano il sistema nervoso centrale, questo tipo di struttura è visibile in Fig. 6.1. Ciascuna di queste cellule esegue un’operazione che consiste nell’acquisire i segnali dei neuroni vicini attraverso i dendriti che li collegano (canali di input) ed elaborare la risposta nel soma; se la somma pesata dei contributi supera una certa soglia allora si produce un “potenziale d’azione” che viene trasportato dall’assone (canale di output) ai neuroni collegati, altrimenti il neurone resta in uno stato di riposo.

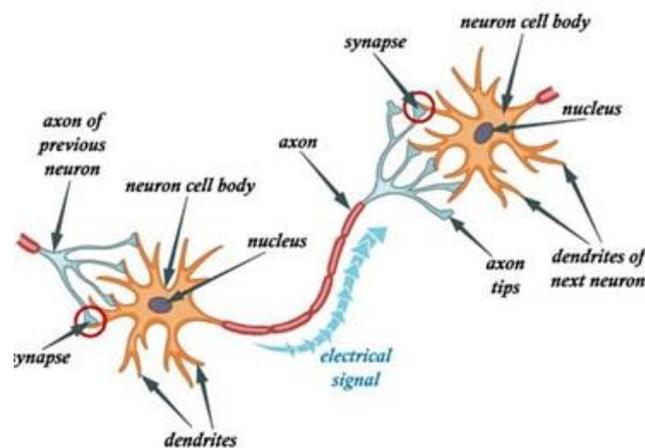


Fig. 6.1 Struttura Dei Neuroni

Fonte [22]

Le caratteristiche principali che rendono queste strutture estremamente interessanti sono le capacità di apprendimento e la memoria: riusciamo non solo a ricordare le informazioni, ma anche a valutare come si evolverà una determinata situazione con l’esperienza di casi analoghi. Questo è possibile perché alcune caratteristiche dei neuroni possono essere modificate mediante due fenomeni:

- La permeabilità della membrana della cellula, che influenza la soglia di reazione
- La plasticità della sinapsi, che è in grado di modulare la connessione tra i neuroni (un collegamento debole farà sì che l’attività del neurone ricevente sia poco influenzata dall’attività del neurone trasmettitore)

6.1.1 PERCEPTRON

I primi ad aver ideato un modello matematico per i neuroni, visibile in figura Fig. 6.2 sono stati McCulloch e Pitt (nel 1943) [24], la loro idea era molto semplice e consisteva in una prima parte con il ruolo di raccogliere gli input e calcolare la somma pesata, in base al valore ottenuto la seconda parte genererà un output.

Pochi anni dopo (nel 1958) [25] questa semplice idea venne approfondita da Frank Rosenblatt che, nel suo articolo menzionò per la prima volta il “Perceptron”, un algoritmo di classificazione binaria che ha come obiettivo quello di calcolare l’iperpiano che sia in grado di separare le due classi. Per farlo verrà usato un neurone composto da una serie di input, come nella versione precedente, ma in questo caso viene aggiunta una funzione di attivazione non lineare e un bias (solitamente legato a condizioni sulla funzione di attivazione, ad esempio il livello base di attivazione).

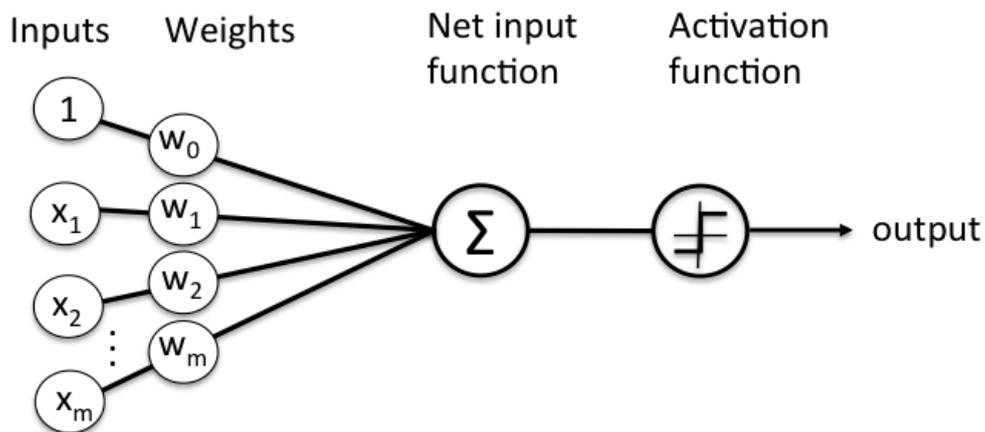


Fig. 6.2 Schema Del Perceptron

Fonte [23]

ACTIVATION FUNCTIONS

Le funzioni di attivazione sono funzioni che ci permettono di trovare l’output di neurone i-esimo

$$o_i = f(a)$$

La funzione f riceve in input la somma pesata degli x_j

$$A = \sum_{j=1}^n w_{ij} x_j$$

dove j è l'indice relativo agli input che entrano nel neurone i -esimo, mentre w_{ji} sono i valori dei pesi di ogni singola interconnessione tra l'input j -esimo e il neurone i -esimo che riceve l'input

Le funzioni di attivazione decidono se un neurone si attiverà o meno per un dato input producendo quindi l'output corrispondente; nelle CNN sono usate dopo ogni convolutional layer. Una caratteristica importante di queste funzioni è che siano differenziabili per consentire l'error back-propagation (che vedremo in seguito) per addestrare il modello.

Alcuni esempi di funzione di attivazione sono:

- Binary Threshold (soglia binaria): quella vista nell'esempio del perceptrone

$$f(t) = \begin{cases} 0 & \text{se } x < 0 \\ 1 & \text{se } x \geq 0 \end{cases}$$

Il problema di questa funzione è che in zero la funzione non è derivabile, quindi spesso è conveniente usare delle funzioni che sono derivabili sempre.

- Funzione logistica (sigmoide), range (0, 1), definita come:

$$f(t) = \frac{1}{1 + e^{-t}}$$

- Funzione tangente iperbolica, range (-1, 1), definita come:

$$f(t) = \tanh(t)$$

- ReLU (Rectifier Linear Unit), range (0,1), definita come:

$$f(z) = \begin{cases} 0 & \text{se } z < 0 \\ z & \text{se } z \geq 0 \end{cases}$$

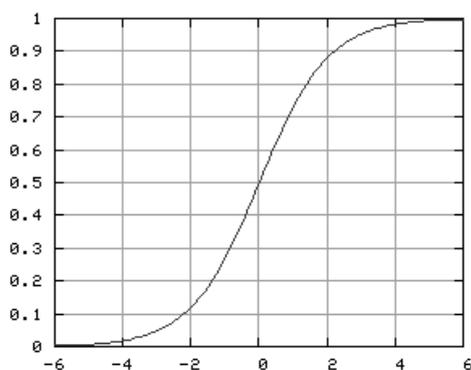


Fig. 6.3 Grafico Della Funzione Logistica

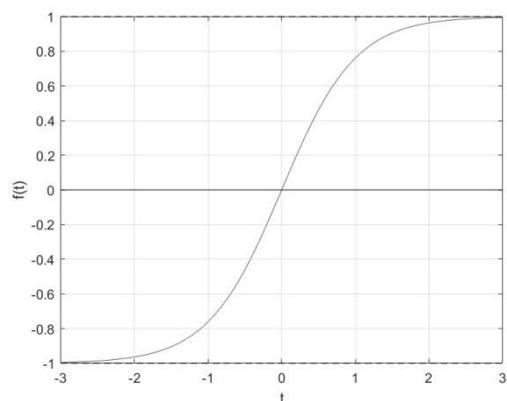


Fig. 6.4 Grafico Funzione Tangente Iperbolica



Fig. 6.5 Grafico Funzione ReLU

La funzione ReLU è la più usata nelle CNN in quanto converte tutti i valori di input in numeri positivi e inoltre richiede un costo computazionale inferiore rispetto alle altre funzioni.

6.1.2 FUNZIONE LOGISTICA COME FUNZIONE DI ATTIVAZIONE

La funzione logistica (o sigmoide) ci permette di definire in output un valore che indica la probabilità di appartenere a una determinata classe, è descritta dalla funzione riportata nel paragrafo Activation Functions del capitolo 1.1.1.

L'algoritmo per la back-propagation diventa:

1. Inizializzare i pesi con valori piccoli e random
2. Introdurre nella rete il primo input, lasciarlo propagare per tutta la rete e calcolare l'output
3. Per ogni output neurone calcolo: $\delta_j = (t_j - o_j)f'(net_j) = (t_j - o_j)o_j(1 - o_j)$
 Per ogni hidden neurone calcolo: $\delta_j = f'(net_j) \sum_s \delta_s w_{js} = o_j(1 - o_j) \sum_s \delta_s w_{js}$
 Dove s è il numero di neuroni nascosti

4. Per ogni layer aggiornare i pesi, per ogni unità avremo:

$$\Delta w_{ij}(n + 1) = \eta \delta_j o_j + \alpha \Delta w_{ij}(n)$$

5. Ripetere gli step da 5 a 4 per tutti gli input del training
6. Calcolare l'errore E su tutto il training set, se $E \leq \text{toll} \Rightarrow$ l'algoritmo finisce; altrimenti ripresentiamo il training set e ri-aggiorniamo tutti i pesi.

6.1.3 IL PROBLEMA DEL XOR

Nel 1969 M. Minsky e S. Papert [26] pubblicarono un'analisi che portava a galla una serie di punti deboli del perceptrone, ad esempio il problema del XOR.

XOR è un'operazione logica che il perceptrone per com'era stato costruito non poteva risolvere; supponiamo di avere due classi: x_1 e x_2 e di avere come output $x_1 \text{ XOR } x_2$

La tabella sarebbe questa:

Input 1	Input 2	Output
0	0	0
0	1	1
1	1	0
1	0	1

Tab. 2 Problema Del XOR

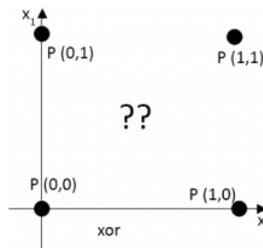


Fig. 6.6 Problema Del XOR

Fonte [24]

Come si può intuire non esiste un iperpiano, in questo caso una retta, che separi le due classi, le soluzioni possono essere due: creare una soluzione ad hoc (poco conveniente in quanto specifica per il singolo problema e a volte inesistente) oppure aggiungere degli hidden layers (ci consente di generalizzare).

Un hidden layer è un livello aggiuntivo inserito tra l'input e l'output, che presenta anch'esso dei neuroni, l'idea è che il problema di partenza venga trasferito in un nuovo spazio all'interno del quale il problema risulti linearmente separabile. Il vantaggio di aggiungere un hidden layer è che mi permette di tracciare tante rette quanti sono i neuroni nascosti nel layer, permettendo quindi di creare regioni arbitrariamente complesse, mentre il numero di hidden layer mi permette di creare tante regioni quanti sono i layer nascosti (all'aumentare del numero di layers aumenta notevolmente anche la complessità).

DELTA RULE

Consideriamo una rete composta da n neuroni di input e p neuroni di output che producono gli output o_j e un training set con valori attesi t_k , la funzione errore per ogni pattern di input k sarà:

$$E_k = \frac{1}{2} \sum_{j=1}^p (t_j - o_j)^2$$

Mentre l'errore su tutto il training set sarà

$$E = \sum_{k=1}^m E_k$$

L'obiettivo è trovare le componenti w che coincidono al minimo della funzione errore, per questo l'algoritmo che si usa è detto a discesa del gradiente, quindi la variazione del peso Δw_{ij} sarà pari a $-G = -\frac{\partial E}{\partial w_{ij}}$ finchè non raggiungo il minimo.

La δ -rule mi dice che, una volta definita delta come $t-y$, la variazione del peso

$$\Delta w_i = \eta \delta x_i$$

η è detta learning rate e sarà discussa nella sezione successiva

Il peso w_i all'epoca (presentazione di tutto il training set) $n+1$ sarà dato dalla seguente formula:

$$w_i(n+1) = w_i(n) + \Delta w_i(n)$$

è possibile vedere, con una serie di passaggi algebrici omissi, che la δ -rule e la regola del gradiente decrescente coincidono.

LEARNING RATE E MOMENTUM

Il learning rate η compare nella formula di aggiornamento dei pesi:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}$$

che indica quanto grande deve essere il passo che viene fatto ad ogni iterazione verso il minimo della funzione di errore.

Se η è un valore troppo grande la convergenza è più rapida ma può succedere che, se ci troviamo in prossimità di un minimo, questo venga superato per rispettare la condizione e quindi diventa difficile convergere verso il minimo della funzione. D'altra parte se i passi che facciamo sono troppo piccoli è più facile arrivare a convergenza, ma il numero di iterazioni necessarie diventa troppo grande e quindi il tempo di apprendimento della rete intera diventa molto lungo.

Per questo motivo spesso il learning rate si sceglie o in maniera sperimentale, quindi in base alla singola applicazione, oppure viene adattato nel tempo, ovvero nelle prime fasi dell'apprendimento il learning rate è più grande e poi man mano che avviciniamo il minimo diminuisce.

Un'altra difficoltà che si incontra è quella dei minimi locali, ovvero la funzione di errore è molto complessa e contiene dei minimi locali oltre ai minimi assoluti, questo fa sì che l'algoritmo possa fermarsi in uno di questi minimi locali.

Una tecnica per superare questo problema è quella di aggiungere un nuovo parametro globale α ($0 < \alpha < 1$) che viene determinato con un approccio trial & error, ovvero non esiste un valore giusto per ogni applicazione ma va adattato a ogni caso. Quello che succede è che la variazione del peso diventa:

$$\Delta w_{ij}(n+1) = \eta \delta x_j + \alpha \Delta w_{ij}(n)$$

La variazione di peso quindi viene data dalla δ -rule più una certa componente che tiene conto della variazione di peso del passaggio precedente, quando il gradiente continua a puntare sempre nella stessa direzione, il momento fa prendere velocità all'algoritmo aumentando la dimensione dello step fatto in quella direzione, questo fa sì che se ci troviamo in un minimo locale il momento ci farà fare uno step abbastanza grande da superarlo.

Per fare questo è necessario che il learning rate sia particolarmente piccolo quando il learning rate è grande, altrimenti gli step potrebbero essere così grandi da farci superare anche il minimo.

Ora che conosciamo tutti gli elementi possiamo vedere come funziona il perceptrone nel caso più semplice, supponiamo che gli ingressi siano dati da:

$$z = w_0 x_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n, \quad \text{con } w_0 = -b \text{ e } x_0 = 1$$

dove x è il vettore degli input, w è il vettore dei pesi legati a ciascun valore di input e b è il valore del bias costante.

I valori di output saranno dati dalla funzione di attivazione segno

$$f(z) = \begin{cases} 1 & \text{se } z \geq 0 \\ -1 & \text{altrimenti} \end{cases}$$

Per addestrare il perceptrone quindi bisogna seguire la regola:

$$\text{sign}(\mathbf{w}^T \cdot \mathbf{x}) = t$$

dove t è il valore target (+1 o -1) a seconda della classe a cui appartiene il punto

Bisogna controllare che il segno del prodotto scalare calcolato dal perceptrone sia lo stesso del target; per farlo l'algoritmo prevede di inizializzare il vettore dei pesi in maniera casuale, di

fissare il valore del learning rate η nell'intervallo $(0, 1]$ e poi cominciare a eseguire la classificazione, aggiornando i valori dei pesi fino a quando w “non cambia più” (decidiamo noi come definire la stopping condition, la più usata prevede di valutare se la differenza tra il nuovo vettore dei pesi e il vecchio vettore dei pesi in valore assoluto è minore rispetto a un valore che riteniamo sufficientemente piccolo).

Quindi all'interno di ogni ciclo, per ogni (\mathbf{x}, t) nel training set calcolo

$$o = f(\mathbf{w} \cdot \mathbf{x}) \Rightarrow \begin{cases} \text{se } o = t \rightarrow \mathbf{w} \text{ non cambia} \\ \text{se } o \neq t \rightarrow \mathbf{w}(\text{old}) + \eta(t - y)\mathbf{x} \end{cases}$$

Questo tipo di perceptrone ha come grande limite quello di convergere in un numero finito di epoche solo in presenza di input linearmente separabili, ovvero solo se esiste un iperpiano che separa le due classi.

Le informazioni per questo capitolo sono state tratte da [27] [28].

7 RINGRAZIAMENTI

A conclusione di questo elaborato, desidero menzionare tutte le persone, senza le quali questo lavoro di tesi non potrebbe esistere.

Un sentito grazie al mio relatore Giovanni Sparacino per la sua fiducia nell'avermi lasciato la possibilità di scegliere personalmente l'argomento di questa tesi e per l'infinita disponibilità e tempestività ad ogni mia richiesta.

Grazie anche al mio correlatore Pietro Scimemi per i suoi preziosi consigli e per avermi sempre supportata mettendo a disposizione le sue conoscenze e i materiali dell'ospedale.

Ringrazio infinitamente i miei genitori, Carlo e Antonella, che mi hanno sempre sostenuta, appoggiando ogni mia decisione, fin dalla scelta del mio percorso di studi, senza il loro supporto non sarei mai potuta arrivare fin qui. Grazie per esserci sempre stati soprattutto nei (numerosi) momenti di sconforto.

A mia sorella Marta dedico un grazie particolare, perché nonostante sia spesso silenziosa e sulle sue, mi ha sempre consolata e fatta ridere quando ne avevo bisogno, è stata per me una fonte di ispirazione per la sua determinazione e bravura.

Uno speciale grazie va anche ai miei nonni Paolo e Franca, che con il loro interesse per i miei studi mi hanno sempre spronata a proseguire, grazie per avermi messo da parte tutti i ritagli di giornale che potevano interessarmi e per avermi ospitata a pranzo ogni volta che tornavo indietro dall'ospedale.

Un grazie anche ai miei nonni Bruno e Lina, che non ci sono più, ma che ho sempre sentito vicini.

Grazie ai miei zii, Luisa, Matteo, Mauri e Anna, e ai miei cugini Emma e Nicolò, che hanno saputo regalarmi molti momenti di spensieratezza e allegria nei numerosi pranzi e cene.

Voglio ringraziare anche le mie carissime amiche, che conosco da sempre e da sempre sono per me un punto fisso: Caterina, Linda, Matilde e Sofia; che nonostante i chilometri che ci separano (chi più chi meno) non hanno mai mancato di esserci per me. Un ringraziamento speciale va anche al mio amico Carlo, che essendoci passato prima di me, ha saputo darmi preziosi consigli per sopravvivere a ingegneria.

Ringrazio anche tutti i miei colleghi (ormai ufficialmente amici) per le giornate di studio, gli infiniti aperitivi e le numerose feste per aver reso ogni momento più leggero ma anche pieno di ricordi, ognuno di voi ha contribuito a rendere questi anni indimenticabili.

Ultimo ma non per importanza, voglio ringraziare il mio ragazzo, Matteo, che con la sua infinita pazienza ha saputo essere un supporto, un punto di riferimento e un esempio. Ha sempre trovato le parole giuste per motivarmi e spronarmi anche quando non credevo di potercela fare spingendomi sempre a fare di meglio.

Senza nessuno di voi tutto questo sarebbe stato possibile.

8 BIBLIOGRAFIA

- [1] A. W. Cripps, D. C. Otczyk, and J. M. Kyd, “Bacterial otitis media: a vaccine preventable disease?,” *Vaccine*, vol. 23, no. 17, pp. 2304–2310, Mar. 2005, doi: 10.1016/j.vaccine.2005.01.023.
- [2] M. Balbi, *Atlante Orecchi Medio*.
- [3] L. Monasta *et al.*, “Burden of Disease Caused by Otitis Media: Systematic Review and Global Estimates,” *PLOS ONE*, vol. 7, no. 4, p. e36226, Apr. 2012, doi: 10.1371/journal.pone.0036226.
- [4] J. O. Klein, “The burden of otitis media,” *Vaccine*, vol. 19, pp. S2–S8, Dec. 2000, doi: 10.1016/S0264-410X(00)00271-1.
- [5] C. J. Williams and A. M. Jacobs, “The impact of otitis media on cognitive and educational outcomes,” *Medical Journal of Australia*, vol. 191, no. S9, Nov. 2009, doi: 10.5694/j.1326-5377.2009.tb02931.x.
- [6] W. Mulwafu, R. Ensink, H. Kuper, and J. Fagan, “Survey of ENT services in sub-Saharan Africa: little progress between 2009 and 2015,” *Global Health Action*, vol. 10, no. 1, p. 1289736, Jan. 2017, doi: 10.1080/16549716.2017.1289736.
- [7] S. Camalan *et al.*, “OtoMatch: Content-based Eardrum Image Retrieval using Deep Learning.” Zenodo, Dec. 30, 2019. doi: 10.5281/zenodo.3595567.
- [8] “Ear imagery database.” Feb. 21, 2020. doi: 10.6084/m9.figshare.11886630.v1.
- [9] “Aurical Otocam 300,” *Natus*. <https://natus.com/it-it/prodotti/aurical-otocam-300> (accessed Mar. 30, 2023).
- [10] “QIMIC Otoscopio, Rimozione di Cerume con Telecamera, 1080P 3,5mm Impermeabile Endoscopio con 6 Luci LED, Pulisci Orecchie per iPhone, iPad e Android : Amazon.it: Salute e cura della persona.” https://www.amazon.it/Otoscopio-Rimozione-Telecamera-Impermeabile-Endoscopio/dp/B0999FQLRT/ref=sr_1_1?__mk_it_IT=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crd=1Q2GAHT57ZUL1&keywords=qmic%2Botoscope&qid=1670841292&srefix=qmic%2Boto%2Caps%2C331&sr=8-1&th=1 (accessed Mar. 30, 2023).
- [11] Y. LeCun *et al.*, “Handwritten Digit Recognition with a Back-Propagation Network,” in *Advances in Neural Information Processing Systems*, Morgan-Kaufmann, 1989. Accessed: Feb. 21, 2023. [Online]. Available: <https://proceedings.neurips.cc/paper/1989/hash/53c3bce66e43be4f209556518c2fcb54-Abstract.html>
- [12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.
- [13]

- C. Szegedy *et al.*, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9. doi: 10.1109/CVPR.2015.7298594. [14]
- K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90. [15]
- S. Serte, A. Serener, and F. Al-Turjman, “Deep learning in medical imaging: A brief review,” *Trans Emerging Tel Tech*, vol. 33, no. 10, Oct. 2022, doi: 10.1002/ett.4080. [16]
- M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks.” arXiv, Mar. 21, 2019. doi: 10.48550/arXiv.1801.04381. [17]
- J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, in NIPS’14. Cambridge, MA, USA: MIT Press, Dec. 2014, pp. 3320–3328. [18]
- D. Cha, C. Pae, S.-B. Seong, J. Y. Choi, and H.-J. Park, “Automated diagnosis of ear disease using ensemble deep learning with a big otoendoscopy image database,” *EBioMedicine*, vol. 45, pp. 606–614, Jul. 2019, doi: 10.1016/j.ebiom.2019.06.050. [19]
- D. Livingstone and J. Chau, “Otosopic diagnosis using computer vision: An automated machine learning approach,” *Laryngoscope*, vol. 130, no. 6, pp. 1408–1413, Jun. 2020, doi: 10.1002/lary.28292. [20]
- J. Sandström, H. Myburgh, C. Laurent, D. W. Swanepoel, and T. Lundberg, “A Machine Learning Approach to Screen for Otitis Media Using Digital Otoscope Images Labelled by an Expert Panel,” *Diagnostics (Basel)*, vol. 12, no. 6, p. 1318, May 2022, doi: 10.3390/diagnostics12061318. [21]
- “Tympanic membrane / eardrum dataset / otitis media.” <https://www.kaggle.com/datasets/erdalbasaran/eardrum-dataset-otitis-media> (accessed Mar. 30, 2023). [22]
- “Personale sanitario.” http://dati.istat.it/Index.aspx?DataSetCode=DCIS_PERS_SANIT# (accessed Mar. 30, 2023). [23]
- G. Anastasi, *Trattato di Anatomia Umana*, 4th ed. edi-ermes, 2006. [24]
- W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259. [25]
- “APA PsycNet.” <https://psycnet.apa.org/record/1959-09865-001> (accessed Feb. 21, 2023). [26]
- M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry*. 1969. [27]
- A. Zhang, C. Lipton, M. Li, and J. Smola, *Dive into Deep Learning*.

[28]

S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Hamilton, Ontario, Canada: Pearson, 2008.

[29]

H. C. Myburgh, W. H. van Zijl, D. Swanepoel, S. Hellström, and C. Laurent, “Otitis Media Diagnosis for Developing Countries Using Tympanic Membrane Image-Analysis,” *EBioMedicine*, vol. 5, pp. 156–160, Mar. 2016, doi: 10.1016/j.ebiom.2016.02.017.

9 BIBLIOGRAFIA IMMAGINI

- [1] <https://www.sordita.it/sordita/sordita-trasmissiva/>
- [2] <https://fisiologicamente.altervista.org/anatomia-orecchio-esterno/>
- [3] <https://www.tomatismodena.it/anatomiaorecchio/>
- [4] https://it.wikipedia.org/wiki/Timpano_%28anatomia%29
- [5] <https://www.fisioterapiarubiera.com/anatomia-dell-orecchio/>
- [6] <https://images.app.goo.gl/kih1Lm1a93zFc56H6>
- [7] <http://www.specialistaotorino.it/blog/otorinolaringoiatria-pediatria/otite-media-acuta/>
- [8] <https://www.medispo.com/shop/diagnostica/otoscopi/dh1120-otoscopio-a-fibre-ottiche/>
- [9] <https://www.medicaexpo.it/prod/globalmed/product-82214-529457.html>
- [10] https://www.amazon.it/Otoscopio-Rimozione-Telecamera-Impermeabile-Endoscopio/dp/B0999FQLRT/ref=sr_1_1?__mk_it_IT=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=1Q2GAHT57ZUL1&keywords=qmic%2Botoscope&qid=1670841292&srefix=qmic%2Boto%2Caps%2C331&sr=8-1&th=1
- [11] <https://www.nbshare.io/>
- [12] <https://www.mariakhalusova.com/posts/2019-04-17-ml-model-evaluation-metrics-p2/>
- [13] Wikipedia
- [14] <https://www.nxp.com/docs/en/training-reference-material/TP-NXP-EBV-AI-Webinar-Day1-20200707.pdf>
- [15] <https://it.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html>
- [16] <https://www.mdpi.com/2072-4292/12/7/1194>
- [17] <https://www.mdpi.com/2072-4292/12/7/1194>
- [18] https://en.wikipedia.org/wiki/Residual_neural_network#/media/File:ResNets.svg
- [19] <https://cv-tricks.com/keras/understand-implement-resnets/>

[20] <https://paperswithcode.com/method/googlenet>

[21] <https://it.mathworks.com/help/deeplearning/gs/get-started-with-transfer-learning.html>

[22] <https://www.alzheimer-riese.it/contributi-dal-mondo/ricerche/4267-attivita-dei-dendriti-e-cruciale-per-la-memoria>

[23] <https://medium.com/@lucaspereira0612/solving-xor-with-a-single-perceptron-34539f395182>

[24] <https://www.reyesandres.com/2015/12/31/i-limiti-del-perceptron-il-problema-della-xor/>