Intelligenza Artificiale e diagnosi precoce del COVID-19

Artificial Intelligence and early diagnosis of COVID-19

Agostino Giorgio♦

♦ *Dipartimento di Ingegneria Elettrica e dell'Informazione – Politecnico di Bari*

Sommario

Con il dilagare della pandemia da COVID-19 la comunità scientifica si è attivata tempestivamente innanzitutto per sequenziare il virus e, quindi, per cercare cure adeguate e vaccini, ora finalmente disponibili, per la prevenzione della malattia. Infatti, la sequenza del nuovo virus fu pubblicata già dai cinesi ad inizio pandemia e per questo la diagnosi è apparsa subito di facile pronuncia con la tecnica dei tamponi naso-faringei, dei test sierologici e antigenici, delle radiografie toraciche, metodologie tutte già ben note alla comunità scientifica. In assenza di sequenziamento la diagnosi sarebbe stata più difficile. Infatti, nella gran parte delle pandemie del passato l'agente patogeno era sconosciuto mentre nel caso del COVID-19 il sequenziamento del virus ha consentito di partire da una identità genetica ben precisa. Ciò ha permesso poi il rapido sviluppo dei vaccini con nuove tecnologie come quella del mRNA, o RNA messaggero. Per questo motivo diagnosi e prevenzione vaccinale sono strettamente legate. Tuttavia, ciò che sembra meno sviluppato attualmente sono metodi per la diagnosi precoce della malattia che sarebbero utili soprattutto per prevenire o curare ai primi sintomi la polmonite interstiziale che è la causa principale dei ricoveri in terapia intensiva e dei decessi. Obiettivo di questo lavoro è mostrare come ci siano tecnologie tipicamente utilizzate in elettronica per l'acquisizione ed elaborazione di dati, specialmente di immagini, con particolare riferimento ad algoritmi di intelligenza artificiale (IA), nota anche come Deep Learning (DL) e Machine Learning (ML), che potrebbero permettere una diagnosi assai precoce dell'insorgere della polmonite interstiziale da COVID-19 e non solo. A tale scopo, almeno per un primo screening, potrebbe essere anche sufficiente l'utilizzo di smartphone di media capacità, senza necessità di ricorrere a costosa strumentazione medica ed esami diagnostici generalmente proibitivi per tempi di attesa e molto onerosi. Pur nella consapevolezza che sono sempre più gli scienziati ed i gruppi di ricerca che utilizzano strumenti di IA per la diagnosi medica del COVID-19 e non solo, l'obiettivo del presente lavoro non è quello di presentare un'analisi critica e/o descrittiva delle attività in corso di svolgimento da parte della comunità scientifica bensì quello di presentare gli strumenti ingegneristici alla base delle attività dei vari gruppi di ricerca con strumenti di IA. Pertanto, vedremo innanzitutto come sia possibile diagnosticare precocemente l'insorgere della polmonite interstiziale; successivamente faremo una panoramica comparativa, qualitativa e quantitativa, dei metodi algoritmici utili allo scopo; infine, vedremo come uno smartphone possa essere utile come strumento per la diagnosi precoce.

Abstract

With the spread of the COVID-19 pandemic, the scientific community took prompt action, first of all to sequence the virus and, therefore, to seek adequate treatments and vaccines, now finally available, for the prevention of the disease. In fact, the sequence of the new virus was already published by the Chinese at the beginning of the pandemic and for this reason the diagnosis immediately appeared to be easy to pronounce with the technique of nasopharyngeal swabs, serological and antigenic tests, chest radiographs, all methods already well known. to the scientific community. Without sequencing, diagnosis would have been more difficult. In fact, in most of the pandemics of the past the pathogen was unknown while in the case of COVID-19, the sequencing of the virus made it possible to start from a very specific genetic identity. This then allowed the rapid development of vaccines with new technologies such as mRNA, or messenger RNA. For this reason, vaccination diagnosis and prevention are closely linked. However, what seems less developed at present are methods for early diagnosis of the disease which would be useful especially when it is becoming more complicated towards interstitial pneumonia which is the main cause of ICU admissions and deaths. The aim of this work is to show how there are technologies typically used in electronics for the acquisition and processing of data, especially images, with particular reference to artificial intelligence (AI) algorithms, also known as Deep Learning (DL) and Machine Learning. (ML), which could allow a very early diagnosis of the onset of COVID-19 interstitial pneumonia and more. For this purpose, at least for a first screening, the use of medium-capacity smartphones may also be sufficient, without the need to resort to expensive medical equipment and diagnostic tests that are generally prohibitive for waiting times and very onerous. Despite the awareness that more and more scientists and research groups are using AI tools for the medical diagnosis of COVID-19 and beyond, the aim of this work is not to present a critical and / or descriptive analysis of the activities in progress by the scientific community but to present the engineering tools underlying the activities of the various research groups with AI tools. Therefore, we will first see how it is possible from a medical point of view to diagnose the onset of interstitial pneumonia early; subsequently we will make an overview of the algorithmic methods useful for this purpose; finally, we will see how a smartphone can be useful as a tool for early diagnosis.

1. Introduzione

I polmoni (fig. 1) sono i due organi preposti alla fornitura di ossigeno all'organismo e all'eliminazione dell'anidride carbonica dal sangue, ovvero agli scambi gassosi fra aria e sangue (processo noto con il nome di ematosi). Situati nella cavità toracica, sono avvolti da una membrana sierosa, la pleura, fondamentale per lo svolgimento delle loro funzioni. I polmoni sono separati da uno spazio compreso tra la colonna vertebrale e lo sterno, il mediastino, che comprende al suo interno il cuore, l'esofago, la trachea, i bronchi, il timo e i grossi vasi.

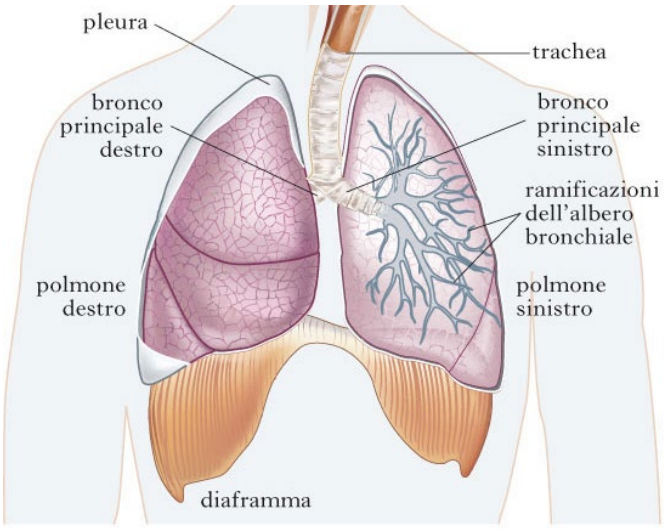

Figura 1 - *Struttura dell'apparato respiratorio: i polmoni*

Il loro compito principale è quello di ricevere il sangue carico di anidride carbonica e prodotti di scarto dalla circolazione periferica e di "ripulirlo" arricchendolo di ossigeno per poi inviarlo al cuore, da dove viene fatto circolare verso organi e tessuti.
La respirazione fisiologica è un processo automatico, controllato inconsciamente dal centro respiratorio posto alla base del cervello. La respirazione può essere controllata anche volontariamente, per esempio, quando si parla, canta o quando si trattiene volontariamente il respiro.
Gli organi sensoriali situati nel cervello, nell'aorta e nelle carotidi monitorano e rilevano i livelli ematici di ossigeno e di anidride carbonica interni all'organismo.
Nei soggetti sani, l'aumento della concentrazione di anidride carbonica rappresenta lo stimolo di maggiore importanza per una respirazione più profonda e rapida. Al contrario, quando la concentrazione di anidride carbonica nel sangue è bassa, il cervello riduce la frequenza e la profondità del respiro.
Durante la respirazione è possibile acquisire ad un segnale (fig. 2) la cui morfologia è assimilabile approssimativamente ad una sinusoide di periodo compreso tra 3.3 e 5 s avendo dunque una frequenza compresa tra 0.2 e 0.3 Hz, in un soggetto sano, la cui respirazione può variare tra 12 e 18 atti/min.

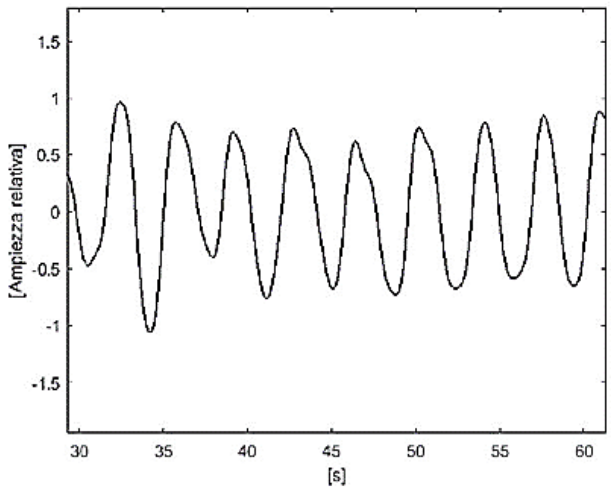

Figura 2 - *Segnale respiratorio con ampiezza normalizzata al suo valore massimo*

In questo segnale i tratti ascendenti rappresentano le inspirazioni e quelli discendenti le espirazioni, mentre la profondità del respiro dipende dal soggetto e dall'attività che sta svolgendo. Infatti, in base al tipo di respirazione ed allo stato dell'individuo, questo segnale può avere diversa ampiezza e i picchi possono essere più o meno ravvicinati nel tempo. Pertanto, dalla frequenza respiratoria e dalla morfologia del suo segnale è possibile risalire ad eventuali patologie che possono essere diagnosticate.

Le patologie dell'apparato respiratorio sono tra le più diffuse essendo legate a molteplici cause, dal banale virus del raffreddore a condizioni più gravi come tubercolosi, cancro ai polmoni e… COVID-19.

La tecnica dell'ascoltazione (o auscultazione) bronco-polmonare è il metodo più semplice e rapido per i medici per rilevare anomalie in fase di inspirazione ed espirazione rispetto al normale murmure vescicolare e diagnosticare o almeno ipotizzare la presenza di patologie.

Infatti, l'ascolto dei suoni respiratori permette ad un orecchio esperto di comprendere se il flusso d'aria all'interno dell'apparato respiratorio fluisce normalmente o è ostacolato da qualcosa. A seconda della sede di passaggio dell'aria si hanno diversi tipi di rumori fisiologici quali rumori vescicolari, bronco-vescicolari, bronchiali e tracheali [1, 2].

Il murmure vescicolare è il normale suono udibile a livello della maggior parte dei campi polmonari e che viene prodotto dalle piccole vibrazioni delle pareti degli alveoli in un soggetto sano. Tra le caratteristiche da analizzare dei suoni auscultati vi è il rapporto temporale tra inspirazione ed espirazione (I:E) normalmente di 1:2 ma se maggiore di 1:3 significa che vi sono limitazioni del flusso aereo, come nel caso di asma o broncopneumopatia cronica.

Tipici suoni anomali come i sibili (wheezes) durante l'espirazione e i crepitii (crackles) durante l'inspirazione sono ben noti sintomi di malattia [1, 2].

La presenza di qualunque patologia respiratoria si manifesta come un'alterazione dei suoni ascoltati in ampiezza, durata e frequenza. Di seguito viene riportata una tabella con le più

comuni patologie respiratorie e le relative caratteristiche della forma d'onda rilevabile tramite uno stetoscopio elettronico.

Tabella 1. Comuni patologie respiratorie e relative caratteristiche dei suoni rilevabili tramite stetoscopio

Tipologia di suono	Frequenza	Durata	Ampiezza/Profondità
Murmure vescicolare (normale)	< 200 Hz	4-5 sec	500 ml
Soffi	> 400 Hz	≥ 100 ms	
Stridore	400-800 Hz		
Ronchi	< 400 Hz	> 100 ms	
Rantoli	> 400 Hz	≥ 10 ms	
Crepitii	> 1000 Hz	< 7 ms	
Eupnea (normale)	14-20 atti/min	4-5 sec	500 ml
Apnea	nulla	15 sec	nulla
Tachipnea	> 20 fino a 40-60 atti/min		500 ml
Bradipnea	< 16, spiccata < 9 atti/min		500 ml
Respiro di Cheyne-Stokes	varia	45-180 sec	variabile
Respiro di Biot	variabile con alternanza di 4/5 atti/min e apnea	apnea di 10-30 sec	variabile
Respiro di Kussmaul	varia		variabile
Iperpnea	> 20 atti/min		> 500 ml

Oltre al murmure, c'è un particolare segnale, proveniente dall'apparato respiratorio, che può fornire importanti informazioni diagnostiche che è il suono proveniente dalla tosse.
Non tutti i suoni della tosse sono uguali, e questo è ben noto; tuttavia, forse meno noto è che ci sono delle caratteristiche che possono permettere una diagnosi precoce di patologie anche molto gravi, quali la polmonite.
Questo aspetto è stato studiato ed applicato dai ricercatori della Università di Lovanio al bestiame, ovvero per monitorare lo stato di salute delle mucche da latte e diagnosticare precocemente eventuale polmonite [3] ed è stato applicato sia dall'autore del presente articolo [4], che da altri ricercatori [5, 6] per la diagnosi precoce del COVID-19.
Per capire come ciò sia possibile, dobbiamo fare una breve panoramica sull'intelligenza artificiale (IA), applicata in maniera ormai sempre più diffusa ed affidabile per il riconoscimento dei segnali, degli oggetti e dei volti ed in particolare definire il concetto di caratteristiche (tecnicamente note come features) di un segnale, alla base dell'utilizzo di qualunque algoritmo di IA.

Il nocciolo dell'algoritmo per la diagnosi precoce della polmonite da COVID-19 consiste nella trasformazione dei suoni respiratori (inclusa la tosse) in immagini e poi applicare l'IA per classificare tali immagini come normali ovvero patologiche. Questa classificazione è proprio la diagnosi cercata ed è possibile ottenerla con un semplice smartphone, come vedremo.

2. Intelligenza Artificiale: Machine Learning e Deep Learning

Per IA si intende un metodo per analizzare dati che si ispira al modo di funzionare del cervello umano e che si concretizza in diversi modelli o algoritmi di calcolo, noti come reti neurali [7], di enorme utilità in molteplici ambiti.
Le operazioni che svolge l'IA sono: analisi di dati, individuazione e quantificazione di elementi caratterizzanti tali dati (processo noto come estrazione di features) e classificazione dei dati ovvero assegnazione di una ben precisa categoria di appartenenza. Tutto ciò a seguito di un processo noto come addestramento del modello, ovvero di attribuzione da parte del modello di IA di un determinato tipo e valore di features ad una determinata categoria di appartenenza.
Questo metodo è ormai di uso comune nel riconoscimento del parlato (implementato nei ben noti assistenti vocali presenti negli smartphone e nei PC e in dispositivi "intelligenti" come Alexa, Siri, Google) [8, 9] e si applica molto bene alla classificazione delle immagini ovvero al riconoscimento automatico di oggetti perché ogni classe di oggetti ha delle caratteristiche ben precise: la classe delle penne, per quanto siano le penne una diversa dall'altra, presenta caratteristiche ben diverse dalla classe delle automobili, per esempio.
Nell'ambito, poi, di uno stesso tipo di oggetti (le automobili, per esempio) è chiaro che ognuno ha caratteristiche proprie che lo distinguono dagli altri per cui anche in questo caso è possibile estrarre features che permettono di stabilire con un certo grado di probabilità di quale oggetto specifico si tratti all'interno di una certa categoria.
È molto importante sottolineare che l'IA fornisce risposte non certe in assoluto ma certe con un determinato livello di "confidenza", cioè con una determinata probabilità.
Nelle neuroscienze, il termine "rete neurale" viene utilizzato con riferimento a una rete o a un circuito formato da neuroni. L'esistenza di reti neurali biologiche, naturali, ha ispirato nell'informatica le cosiddette reti neurali artificiali (ANN, Artificial Neural Network).
Una ANN è un modello di calcolo la cui struttura stratificata assomiglia alla struttura della rete di neuroni nel cervello, con strati di nodi connessi. Una ANN può apprendere dai dati, quindi può essere addestrata a riconoscere pattern, classificare i dati e prevedere ("calcolare") eventi futuri sulla base di quanto ha appreso.
Una ANN combina diversi livelli (layer) di elaborazione. È formata, infatti, da un layer di input, uno o più layer nascosti e un layer di output. I layer sono interconnessi tramite nodi o neuroni, ed ogni layer utilizza l'output del layer precedente come input, per una elaborazione in cascata dei dati in ingresso.
Le ANN che operano su due o tre layer di neuroni connessi sono conosciute come reti neurali superficiali. Al contrario, le reti profonde, note come reti di Deep Learning (DL), possono avere molti layer, anche centinaia. Entrambe sono tecniche di Machine Learning (ML) che imparano direttamente dai dati di input [10-12].

Il DL è particolarmente adatto per applicazioni di identificazione complesse come il riconoscimento facciale, la traduzione di testi e il riconoscimento vocale. È anche una tecnologia chiave utilizzata nei sistemi avanzati di assistenza alla guida tra cui la classificazione delle corsie, il riconoscimento della segnaletica stradale e dei pedoni [13, 14].
Il DL in realtà non è un concetto troppo recente. Infatti, è stato teorizzato per la prima volta negli anni '80, ma si è sviluppato soltanto di recente, con il potenziamento esponenziale dei sistemi di calcolo, per due ragioni principali:

- perché un addestramento efficace, che permetta di ottenere poi risultati corretti con una elevata probabilità, richiede una grande quantità di dati già classificati, etichettati. Per esempio, per lo sviluppo delle automobili a guida autonoma sono necessarie milioni di immagini e migliaia di ore di video;
- perché richiede una notevole potenza elaborativa. Le attuali Graphic Processing Unit (GPU) ad alte prestazioni sono dotate di un'architettura parallela molto efficiente per il DL [15]. In combinazione con i cluster o il cloud computing, i team di sviluppo sono in grado di ridurre i tempi di addestramento per una rete di DL da diverse settimane a poche ore.

Qual è la differenza tra ML e DL?
Il DL è una forma specifica di ML. Come schematizzato in figura 3, un flusso di lavoro di ML per classificare, per esempio, automobili, inizia con un processo di estrazione delle *features* dalle immagini (fotogrammi da un video in tempo reale, per esempio), processo da eseguire separatamente rispetto alla fase di ML vera e propria e che richiede che vengano esplicitamente indicate le *features* da estrarre. Queste vengono quindi organizzate in un *dataset* ed utilizzate per creare un modello che categorizza gli oggetti nell'immagine.
Con un flusso di lavoro di DL, invece, le *features* significative vengono individuate automaticamente ed estratte dalle immagini, con notevole vantaggio perché non è richiesta una fase preliminare di individuazione e validazione delle stesse da parte dell'operatore. Inoltre, il DL esegue un apprendimento *end-to-end*, in cui una rete apprende automaticamente come elaborare dati grezzi e svolgere un'attività di classificazione.
Un vantaggio fondamentale del DL è la possibilità di migliorare le prestazioni con l'aumentare della quantità di dati forniti per l'addestramento [16].

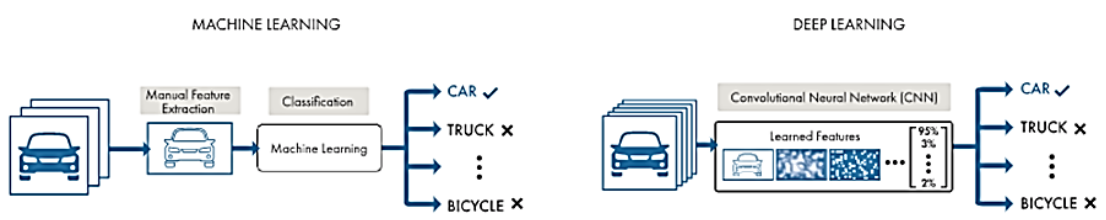

Figura 3 - *Confronto tra un approccio di ML (a sinistra) e DL (a destra) nella categorizzazione di veicoli. Il ML prevede la selezione manuale delle features e di un classificatore per l'ordinamento delle immagini. Con il DL l'estrazione delle features e la procedura di modellazione sono automatiche*

Al fine di produrre una diagnosi precoce di COVID-19 da tosse è più indicato il DL per via dell'automatismo con cui vengono individuate ed estratte le *features* ma si potrebbe anche usare il ML qualora non vi fossero dati a sufficienza a disposizione per addestrare la ANN.
Vediamo, quindi, come creare e addestrare un modello di DL.
I tre modi più comuni sono [16]:

- Addestramento della rete da zero e creazione del relativo modello di classificazione: per addestrare una rete di DL da zero, è necessario raccogliere un set di dati etichettati di grandi dimensioni e progettare un'architettura di rete dedicata. Ciò è utile per le nuove applicazioni o per le applicazioni che dispongono di un grande numero di categorie di output. Si tratta di un approccio poco comune poiché, considerata la grande quantità di dati e la critica velocità di apprendimento, il progetto e l'addestramento di queste reti richiede giorni o settimane.
- Transfer Learning: la maggior parte delle applicazioni di DL utilizza l'approccio denominato Transfer Learning (TL), un processo che consiste nella modifica e nel riaddestramento di un modello già esistente e precedentemente addestrato (rete preaddestrata) per riconoscere categorie di oggetti (classi) diversi da quelli di interesse [17, 18]. Si parte, dunque, da una rete neurale esistente (ve ne sono numerose: AlexNet, GooglNet, SquezeeNet, ResNet, CoffeeNet, ecc.) e la si modifica e riaddestra a riconoscere nuove classi di oggetti [19].
- Se la rete preaddestrata è in grado di riconoscere, ad esempio, 1000 classi di oggetti, con il TL, una volta modificata la rete, è possibile svolgere una nuova attività, per esempio il riconoscimento di sole 5 classi di oggetti anziché 1000. Questo approccio presenta il vantaggio di non dover progettare una ANN da zero e di richiedere molti meno dati per l'addestramento rispetto al caso in cui questo avvenga da zero, per cui i tempi di calcolo si riducono notevolmente.

In figura 4 vengono rappresentati gli step per modificare ed addestrare la rete tramite TL.
Successivamente verranno forniti alcuni dettagli in merito al tipo di ANN utilizzate per il DL sia alla specifica procedura di TL

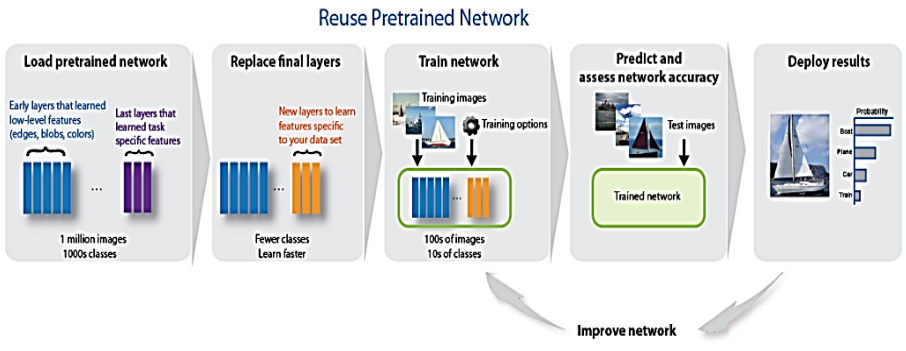

Figura 4 - *Algoritmo relativo all'addestramento di una rete neurale tramite il transfer learning. Tali steps devono essere eseguiti rigorosamente in cascata.*

2.1 Algoritmi per il Deep Learning: reti neurali convoluzionali

Nell'apprendimento automatico, una Rete Neurale Convoluzionale (CNN) è un tipo di ANN in cui il pattern di connettività tra i neuroni è ispirato dall'organizzazione della corteccia visiva animale [20]. Oltre alle CNN esistono anche altre reti come le reti LSTM, le quali a differenza delle prime acquisiscono i dati in ingresso come sequenza di vettori, contenenti tutte le informazioni caratterizzanti il dato da classificare.La CNN, come qualsiasi ANN, impara a eseguire attività di classificazione direttamente da immagini, video, testo o suono e non necessita di estrazione manuale delle *features*.
Le applicazioni che richiedono il riconoscimento degli oggetti e la visione artificiale, come i veicoli a guida autonoma e le applicazioni di riconoscimento facciale, si affidano in gran parte alle CNN in combinazione con l'utilizzo di GPU e del calcolo parallelo.
Come qualunque ANN, una CNN è composta da un *layer* di input, uno di output e molti *layer* nascosti nel mezzo.
Questi *layer* eseguono elaborazioni sui dati con l'intento di enfatizzare ed apprendere caratteristiche specifiche dei dati stessi. Tre dei layer più comuni che si replicano ripetutamente per costituire una CNN (fig. 5) sono: *convolution*, attivazione o ReLU e *pooling*.

- *Convolution*: questo layer inserisce le immagini in ingresso attraverso una serie di filtri convoluzionali, ognuno dei quali attiva determinate caratteristiche dalle immagini.
- L'unità lineare rettificata (ReLU), detto anche layer di attivazione, consente un addestramento più rapido ed efficace della rete mappando i valori negativi a zero e mantenendo i valori positivi. Questa operazione viene talvolta definita attivazione poiché solo le funzionalità attivate vengono trasferite al livello successivo.
- Il *pooling* semplifica l'output eseguendo il downsampling non lineare, riducendo il numero di parametri che la rete deve apprendere.

Queste operazioni vengono ripetute su decine o centinaia di livelli, con ogni strato che impara a identificare caratteristiche diverse dell'immagine in ingresso.

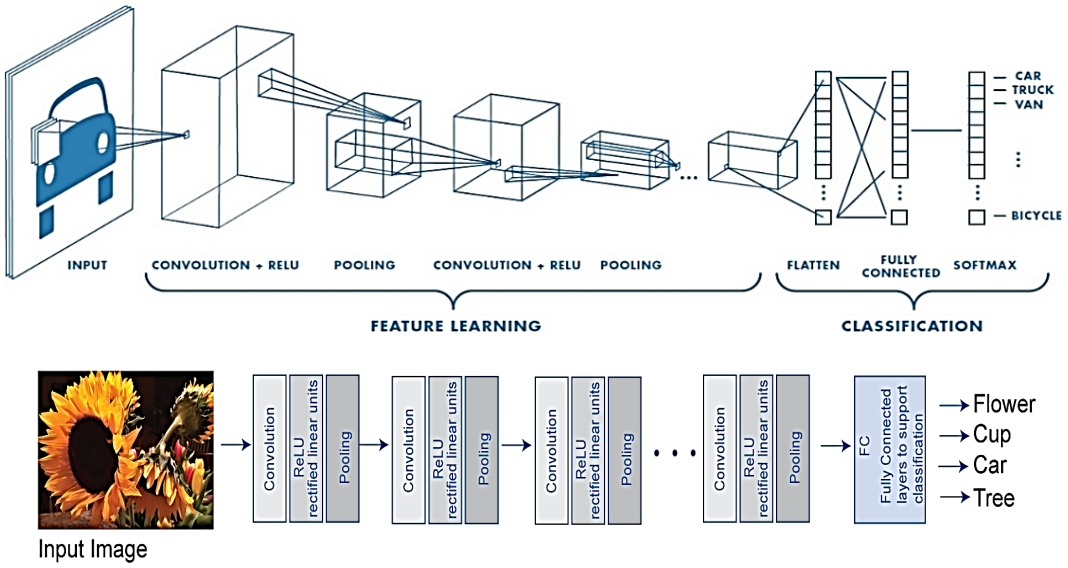

Figura 5 - *Esempio di una rete con molti strati convoluzionali. I filtri vengono applicati a ciascuna immagine di addestramento con risoluzioni diverse e l'output di ciascuna immagine convoluta viene utilizzato come input per il livello successivo. La CNN ha il compito di acquisire l'immagine e classificarla tra differenti categorie*

Il penultimo *layer*, denominato *fully connected layer*, fornisce un vettore di dimensioni K, dove K è il numero di classi che la rete sarà in grado di riconoscere. Il livello finale dell'architettura CNN utilizza un *layer* di classificazione per fornire *l'output* della classificazione cioè l'indicazione della classe cui appartiene l'oggetto rappresentato nella immagine di *input*.
Per la classificazione di oggetti da immagini la CNN riconosce la variazione di tonalità di colore in ogni pixel dell'immagine, così da capire sulla base della variazione della stessa di quale oggetto si tratti. L'algoritmo di riconoscimento quindi, si basa sul "percorrere" l'immagine in tutta la sua dimensione per intercettare variazioni di tonalità di colore, assegnando a ogni regione un valore numerico interpretato ed elaborato dai vari *layer* interconnessi.

2.2 Addestramento di una CNN

Per l'addestramento di una CNN il TL si rivela ancora una volta come un metodo molto efficiente perché richiede una minore quantità di dati rispetto all'addestramento da zero e tempi di calcolo ridotti.
Occorre, a tal fine, come schematizzato in figura 4, innanzitutto scegliere la rete preaddestrata che si ipotizza essere la più adeguata al tipo di attività che si vuole eseguire e, prima di avviare la procedura di riaddestramento, occorre modificarne opportunamente alcuni *layer* in modo che sia adatta al nuovo tipo di attività (ad esempio se dobbiamo analizzare immagini e classificare automaticamente la presenza di gatti, automobili, persone, penne le classi di output della rete saranno 4 mentre partiamo da una rete preaddestrata che sa classificare 1000 tipologie di oggetti diversi da quelli di nostro interesse).

Modificata la rete, occorre disporre di un *dataset* per l'addestramento, da suddividere in *mini-batch*: ogni iterazione del processo di addestramento utilizza differenti *mini-batch* di dati.
Ogni ciclo di addestramento è chiamato *epoch*. Il numero massimo di *epoch* e la dimensione delle *mini-batch* possono essere impostati come *training options*. In ogni *epoch* viene calcolata la perdita e la precisione del gradiente dell'addestramento eseguito.
Il gradiente permette di percorrere lungo una direzione specifica l'immagine in fase di analisi al fine di riconoscere le variazioni di tonalità dei colori, come visibile in figura 6.

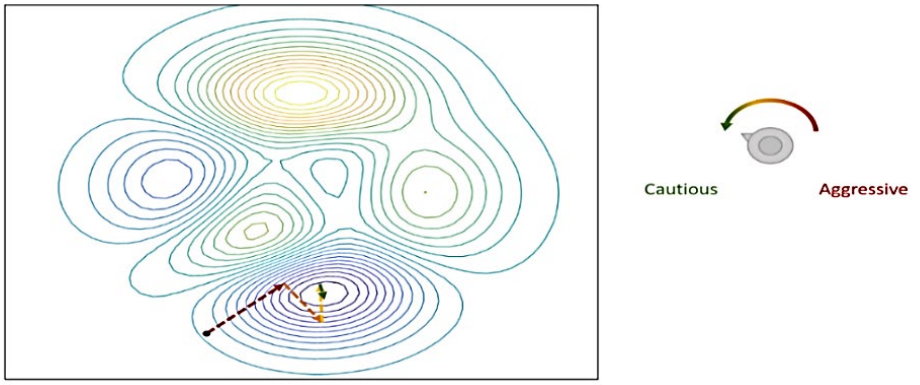

Figura 6 - *Percorso dell'operatore gradiente e della sua variazione all'interno dell'immagine.*

La dimensione dello *step* è chiamato *learning rate***,** il quale parte da un valore iniziale chiamato *initial learning rate*.
Durante l'addestramento viene prodotto un grafico (vedi fig. 7) in cui per ogni *epoch* è rappresentata la precisione e la perdita.

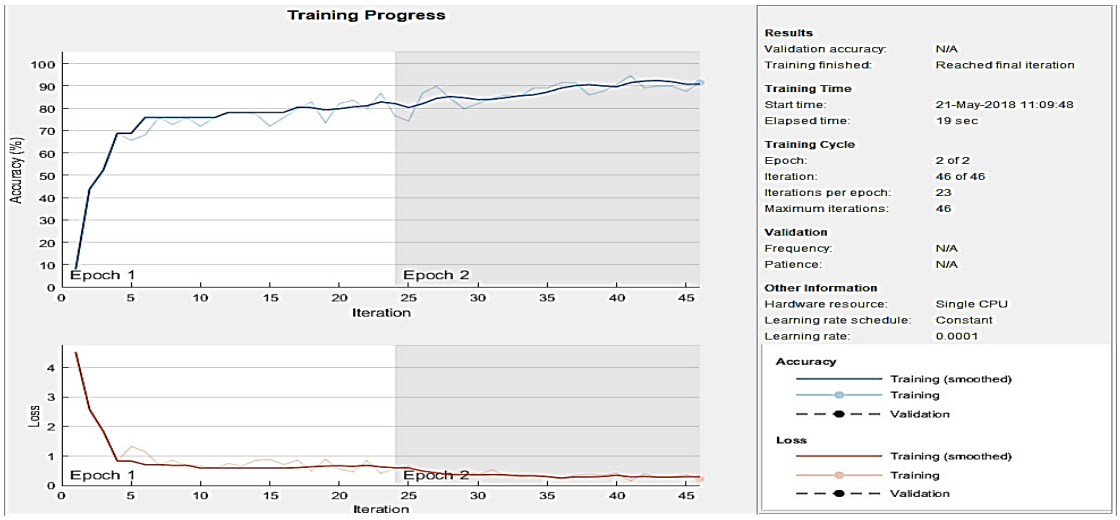

Figura 7 - Esempio di grafico di training progress in cui sono visibili l'andamento di accuratezza e perdite durante l'addestramento della rete.

2.3 Esportazione della rete addestrata

Al termine dell'addestramento è possibile esportare la rete addestrata inserendola successivamente come modello in algoritmi di IA. La esportazione nella forma di *compact model* permette anche, nell'ambito dello sviluppo di progetti in ambiente Matlab/Simulink, di implementare la rete in modelli di IA più complessi adatti alla traduzione di codice, per esempio alla creazione di app per sistemi operativi mobili (Android e iOS) funzionanti, quindi, su smartphone e tablet e al di fuori dell'ambiente Matlab/Simulink [21, 22].

3. Elaborazione di segnali audio per la classificazione con IA

Dalla panoramica sui metodi di IA concludiamo che è possibile effettuare riconoscimento di immagini tramite DL e/o ML effettuando in maniera automatica (con il DL) o manuale (con il ML) l'estrazione delle caratteristiche o *features*, per poi eseguire il riconoscimento ovvero la classificazione.
Affinché questo procedimento possa essere applicato ai dati di tipo audio occorre trasformarli in immagini. Si rende necessario, quindi, chiarire come possa avvenire in maniera efficace questa trasformazione.
Un segnale audio è un'onda di pressione che si propaga in un mezzo trasmissivo come l'aria. La frequenza di campionamento è tipicamente di 44,1 kHz ovvero 44.100 campioni al secondo. Un metodo di elaborazione di segnali audio, molto utile applicato congiuntamente ai modelli di IA, è la trasformata di Fourier (Fast Fourier Transform, FFT).
La FFT è uno strumento matematico attraverso il quale è possibile calcolare la trasformata di Fourirer discreta (DFT, Discrete Fourier Transform) o la sua inversa, ovvero convertire un segnale variabile nel tempo in una rappresentazione tempo-frequenza. Tale rappresentazione descrive il contenuto spettrale (cioè in frequenza) del segnale stesso e come esso si evolve al variare del tempo.
Di seguito verranno descritte diverse tecniche basate sulla rappresentazione tempo-frequenza che si sono rivelate particolarmente utili per la trasformazione in immagini di segnali audio e per la conseguente classificazione tramite DL:

- Spettrogramma Mel [23] e i suoi coefficienti (MFCC) [24];
- Spettrogramma Gammatone [25] e i suoi coefficienti (GTCC) [26];
- Scalogramma e coefficienti Wavelets (CWT) [27-29].

Oltre a queste tecniche ve ne sono altre che richiedono l'utilizzo di reti LSTM.
Tra le altre tecniche disponibili vanno citate la *Pitch* e la *Cepstral Descriptors*.
Le tipologie di immagini in cui possiamo convertire i dati di tipo audio sono tipicamente due: spettrogrammi e scalogrammi, e da queste immagini vengono estratte le *features* per il riconoscimento tramite DL. Tuttavia, le stesse *features*, a loro volta, possono essere rappresentate in forma di immagini. Si tratta in questo caso di immagini ottenute da *features* che sono i coefficienti MFCC, i coefficienti GTCC ed i coefficienti della CWT. Anche queste immagini possono poi essere analizzate tramite algoritmi di DL.
Esaminiamo, quindi, sia le immagini ottenute in forma di spettrogrammi e scalogrammi sia le immagini ottenute dalle *features* degli spettrogrammi e degli.scalogrammi.

3.1 Spettrogrammi e scalogrammi

Lo spettrogramma è la rappresentazione grafica dell'intensità di un suono in funzione del tempo e della frequenza. È dunque un grafico, nel quale sono riportate le frequenze che compongono l'onda sonora al passare del tempo. Lo spettrogramma contiene informazioni sull'ampiezza dell'onda (e quindi sull'intensità del suono), espresse mediante un codice di colori.
In figura 8 ad esempio è rappresentato lo spettrogramma di una nota "Do" emessa da una chitarra acustica.

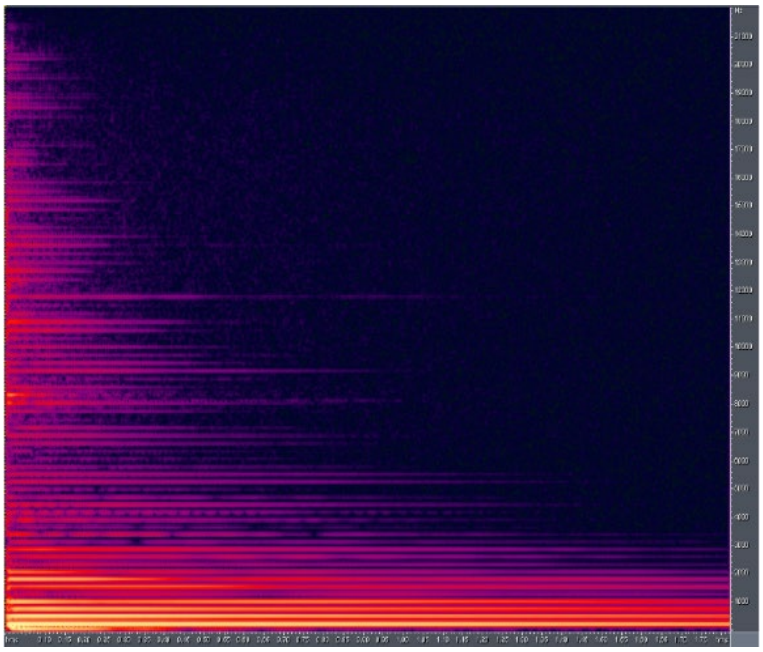

Figura 8 - Spettrogramma di un Do emesso da una chitarra acustica. I colori più chiari (giallo) indicano una maggiore intensità sonora rispetto a quelli più scuri (viola)

Sull'asse delle ordinate (verticale, asse y) vi sono le frequenze, su quello delle ascisse (orizzontale, asse x) il tempo. La presenza di più righe orizzontali, cioè di più frequenze, indica che non si tratta di un suono puro cioè monofrequenziale. Gli strumenti musicali, infatti, sono caratterizzati dai loro timbri, dovuti alla sovrapposizione di onde sonore di più frequenze accanto a quella "più importante", o dominante, che determina la nota. La frequenza dominante si chiama anche "altezza" (*pitch*) del suono. Lo spettrogramma della figura 8 mostra che la nota emessa contiene diverse frequenze, tuttavia i colori più chiari, associati a onde di maggiore ampiezza (e quindi a suoni più intensi), si concentrano intorno a determinate frequenze. Il suono è dunque piuttosto "pulito", cioè riconducibile quasi ad un'unica frequenza, poiché al suono di maggiore intensità corrisponde un intervallo di frequenze piuttosto "stretto".
Osserviamo inoltre che, al passare del tempo, il suono tende ad essere sempre più puro perché le frequenze lontane da quella dominante si attenuano maggiormente.
Diversamente accade quando la stessa identica nota è emessa da un banjo. Il Do prodotto dal banjo, il cui spettrogramma è mostrato in figura 9, contiene molte frequenze di elevata intensità

diverse tra loro, distribuite su un intervallo più ampio. Il suono è dunque meno puro, più "rumoroso", e inoltre ha una durata più limitata nel tempo.

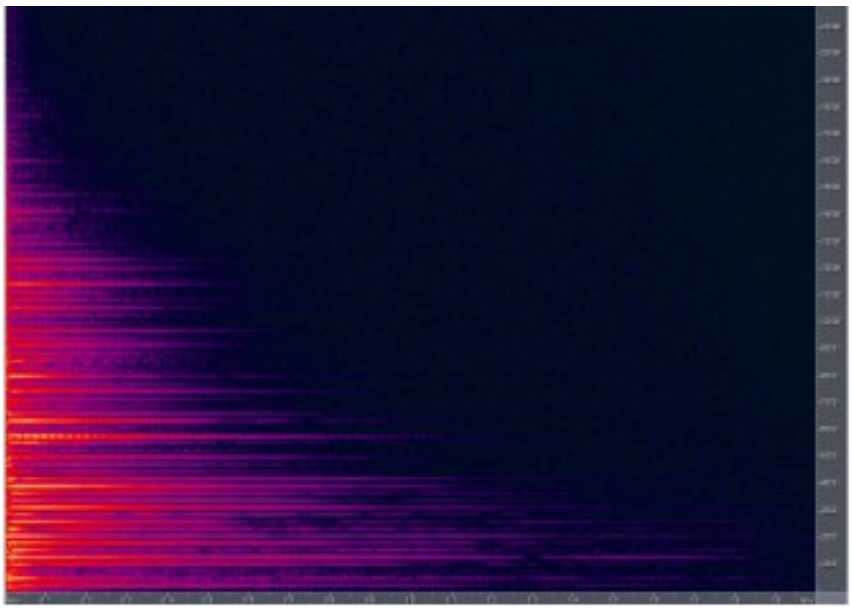

Figura 9 - *Spettrogramma di un Do emesso da un banjo. Un intervallo di frequenze più ampio, rispetto alla chitarra, è caratterizzato da una elevata ampiezza dell'onda sonora e quindi da un suono più intenso*

3.1.1 Generazione di uno spettrogramma e spettrogramma MEL

Uno spettrogramma si ottiene suddividendo la durata totale dell'intera forma d'onda da analizzare in sottointervalli uguali (detti finestre temporali) di durata tipica da 5 a 10 ms e calcolando la trasformata di Fourier della parte di forma d'onda contenuta in ciascuna finestra (solitamente si usa la FFT), che fornisce l'intensità del suono in funzione della frequenza. Le FFT relative alle diverse finestre temporali vengono poi assemblate a formare lo spettrogramma, come una sequenza di FFT impilate una sopra l'altra.
In uno spettrogramma generalmente l'asse delle ordinate viene generalmente convertito in una scala logaritmica e l'ampiezza viene convertita in decibel, dB, che è la scala logaritmica dell'ampiezza.
La scala di rappresentazione delle frequenze, tuttavia, non sempre è logaritmica e in base al modo in cui viene rappresentata otteniamo diversi tipi di spettrogrammi. Ad esempio, l'orecchio umano ha maggiore sensibilità alle frequenze più basse rispetto alle frequenze più alte per cui non percepisce le frequenze su una scala lineare. Questo significa che può facilmente distinguere un suono a 500 hz da uno a 1000 hz, ma difficilmente sarà in grado di distinguere un suono a 10.000 hz da uno a 10.500 hz.
Per questo motivo è definita la scala Mel, mostrata in figura 10. Il nome Mel deriva dalla parola melodia per indicare che la scala si basa sul confronto delle altezze dei suoni.

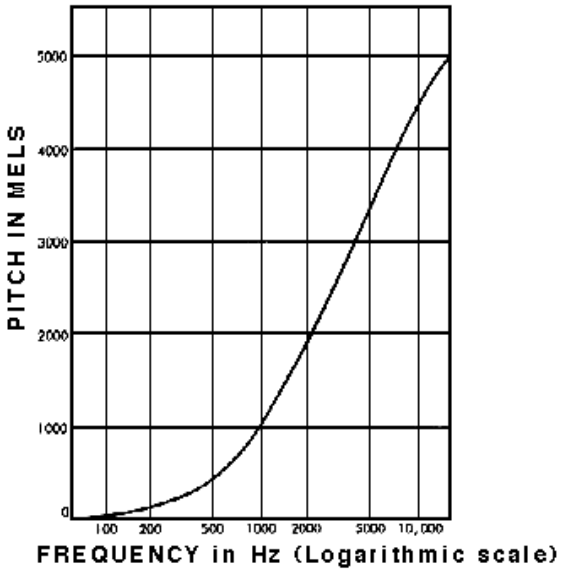

Figura 10 - *Scala Mel*

Infatti, la scala Mel è una scala percettiva dell'altezza di un suono. Il punto di riferimento tra questa scala e la normale misurazione della frequenza è definito assegnando un valore percettivo di 1000 Mels a un tono di 1000 Hz, con una pressione sonora di 60 dB sopra la soglia di ascolto della coclea umana. Al di sopra di circa 500 Hz, infatti, l'orecchio umano non riesce più a distinguere gli incrementi di frequenza. Di conseguenza, quattro ottave sulla scala hertz sopra 500 Hz comprendano circa due ottave sulla scala Mel.
Uno spettrogramma Mel è uno spettrogramma in cui le frequenze vengono convertite nella scala Mel. A scopo esemplificativo in figura 11 è mostrato lo spettrogramma MEL di un segnale audio generico e quello della registrazione di suoni cardiaci (toni cardiaci).

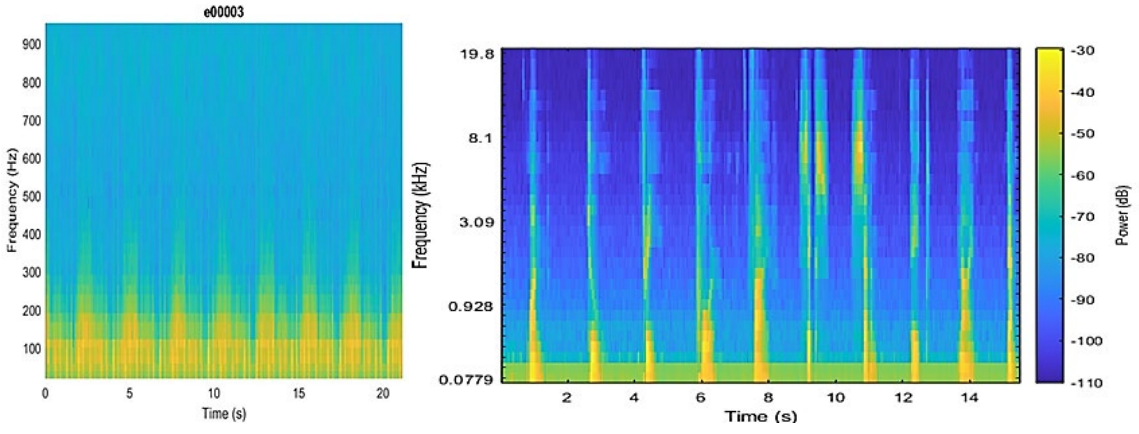

Figura 11 - *Spettrogramma Mel di un segnale audio cardiaco (toni cardiaci) a sinistra, e di un segnale audio generico a destra*

L'asse delle ordinate rappresenta la scala Mel e la variazione del colore rappresenta la variazione della densità di potenza (proporzionale all'ampiezza) del segnale espressa in dB, la quale viene rappresentata graficamente come di consueto con una differente tonalità di colore.

3.1.2 Spettrogramma Gammatone

Lo spettrogramma Gammatone è molto simile allo spettrogramma Mel, con la sola differenza che viene calcolato su una scala diversa, definita ERB (larghezza di banda rettangolare equivalente). La scala di frequenze ERB corrisponde all'incirca a posizionare un filtro ogni 0.9 mm nella coclea, come sarà meglio precisato successivamente.
In figura 12 viene mostrato un esempio di spettrogramma Gammatone di un segnale audio cardiaco e di uno generico.

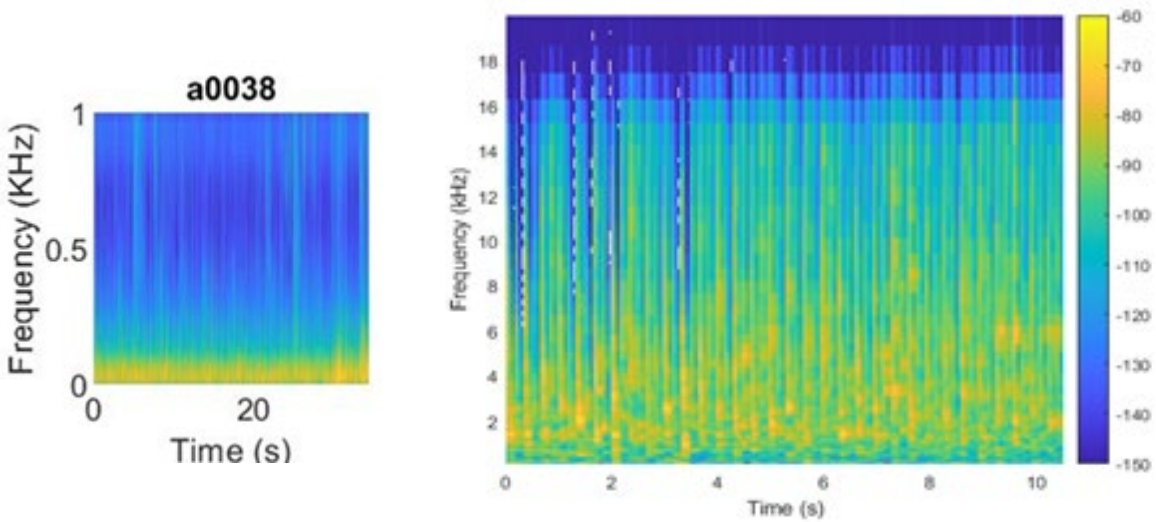

Figura 12 - *Spettrogramma Gammatone di un segnale audio cardiaco (a sinistra) e di uno generico (a destra).*

3.1.3 Trasformata Wavelet e scalogramma

La trasformata *wavelet* è la rappresentazione di un segnale mediante l'uso di una forma d'onda oscillante di lunghezza finita o a decadimento rapido (nota come *wavelet* madre). Questa forma d'onda è scalata e traslata per adattarsi al segnale in ingresso.
La trasformata *wavelet* è spesso paragonata alla trasformata di Fourier, dove i segnali sono rappresentati come somma di armoniche. La differenza principale è che le *wavelet* sono localizzate sia nel tempo sia nella frequenza mentre la trasformata di Fourier standard è localizzata solo in frequenza.
Anche la trasformata di Fourier a tempo breve (Short Time Fourier Transform, STFT) è localizzata in tempo e in frequenza, ma la trasformata *wavelet* offre generalmente una migliore rappresentazione del segnale grazie all'uso dell'analisi multirisoluzione.
La trasformata *wavelet* inoltre è anche meno complessa computazionalmente.

La trasformata *wavelet* di un segnale audio può essere rappresentata come immagine in forma di scalogramma che è il valore assoluto della trasformata *wavelet* continua (CWT) di un segnale, in funzione del tempo e della frequenza.
Esso si usa quando si desidera una migliore localizzazione temporale per eventi di breve durata e ad alta frequenza e una migliore localizzazione della frequenza per eventi a bassa frequenza e di lunga durata. Lo scalogramma può essere più utile dello spettrogramma, ad esempio, per analizzare segnali che variano lentamente punteggiati da transitori improvvisi.
In figura 13 è raffigurato un esempio di scalogramma di un segnale audio cardiaco.

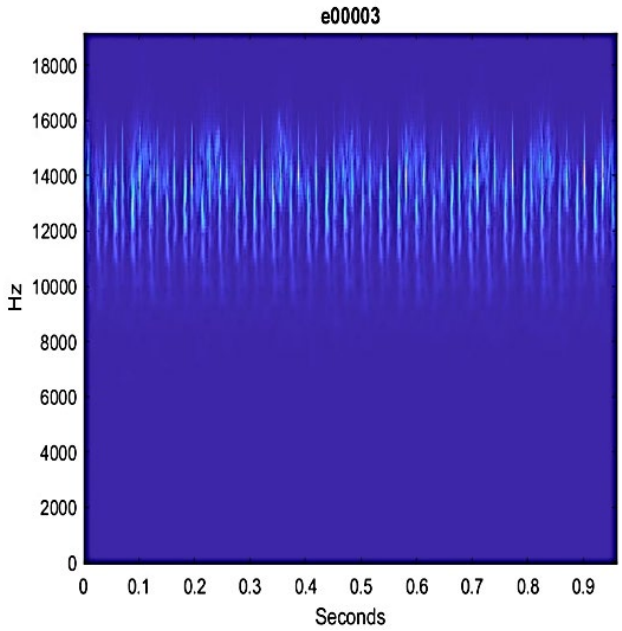

Figura 13 - *Spettrogramma Gammatone di un segnale audio cardiaco (a sinistra) e di uno generico (a destra).*

3.2 Immagini da features

3.2.1 Features dello spettrogramma MEL e immagine da MFCC

Lo spettrogramma MEL viene calcolato sottoponendo il segnale ad un filtraggio con un opportuno banco di filtri (fig. 14). Si tratta tipicamente di filtri triangolari sovrapposti a metà equidistanti tra loro sulla scala Mel, come visibile in figura 15.

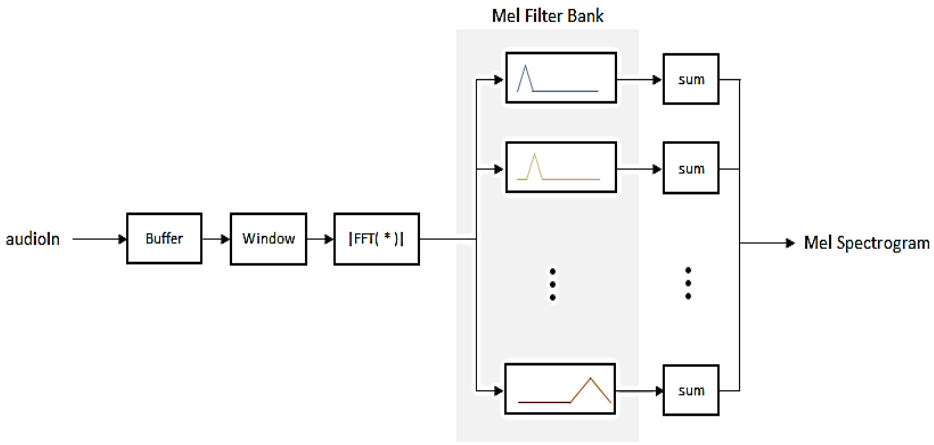

Figura 14 - *Algoritmo generatore dello spettrogramma Mel*

I seguenti tracciati mettono a confronto la degradazione provocata dalla singola presenza dei segnali LTE Uplink a 10 MHz centrati nelle frequenze di 708 MHz, 718 MHz e 728 MHz e la degradazione provocata dalla copresenza dei tre segnali 3x10 MHz, quest'ultimi a pari livello di potenza. La rappresentazione grafica avviene attraverso il rapporto I/C misurato all'ingresso del terminale di testa di un impianto TV impostato a guadagno massimo (≃ 37 dB) e con un livello di segnale utile (DVB-T2, Code Rate 2/3, Guard Interval 1/16, Modulazione 256 QAM ruotata) pari sia a -75 dBm che a -55 dBm. Attraverso la parte terminale del banco, composta da un attenuatore variabile e due matching pad 50/75 ohm, è stato fissato a -50 dBm il segnale utile all'ingresso RF del televisore.
Il primo filtro è molto stretto e dà un'indicazione di quanta energia è presente vicino alla frequenza continua (0 hz); man mano che le frequenze aumentano, i filtri si allargano in quanto è importante sapere solo approssimativamente quanta energia si trova in corrispondenza di frequenze sempre più alte.

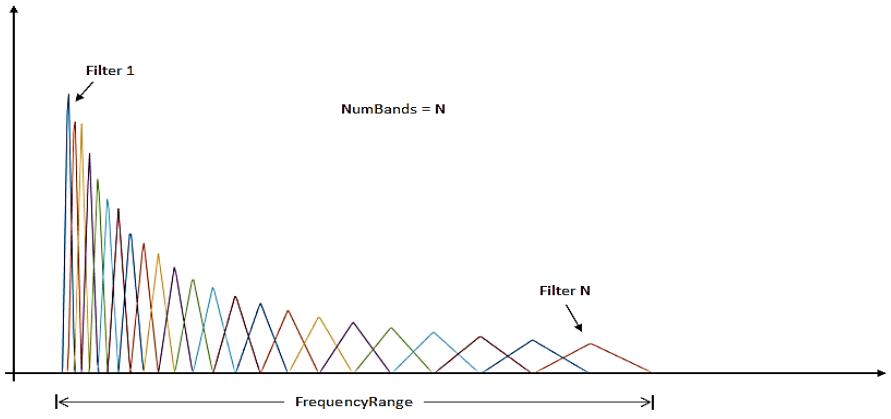

Figura 15 - *Mel filter bank*

Dallo spettrogramma Mel di un segnale audio si possono estrarre gli elementi caratteristici, o *features*, noti come *Mel-Frequency Cepstral Coefficients* (MFCC).
Gli MFCC permettono una rappresentazione del *cepstrum* reale di un segnale.
Il *cepstrum* di un segnale nasce come la trasformata di Fourier del logaritmo della trasformata di Fourier del segnale. A volte viene chiamato lo spettro dello spettro. In seguito, ha preso il sopravvento il calcolo del *cepstrum* come la trasformata di Fourier inversa applicata allo spettro del segnale espresso in scala logaritmica di Mel.
In figura 16 è rappresentato l'algoritmo relativo al calcolo dei coefficienti *cepstral*, il quale risulta essere uguale all'algoritmo visto nella generazione dello spettrogramma, con la sola differenza che in uscita al banco dei filtri viene inserito l'operatore DCT (Trasformata di coseno discreta), attraverso il quale è possibile ricavare proprio gli MFCC.

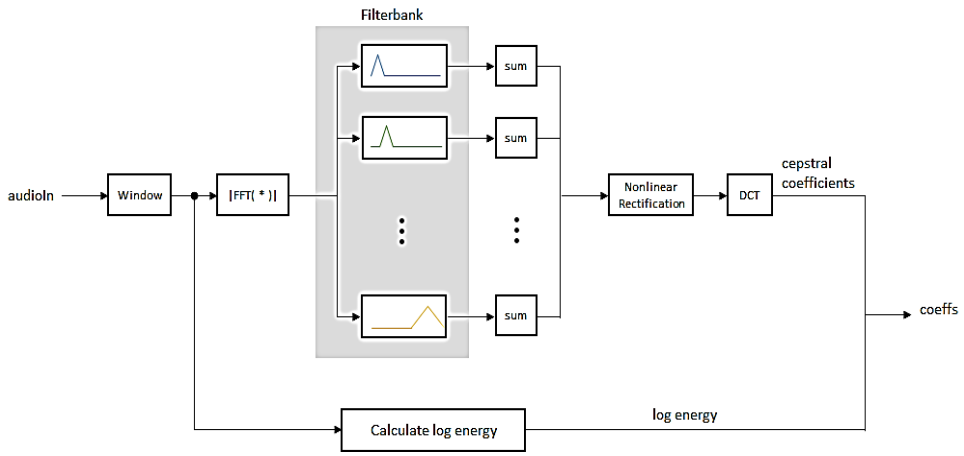

Figura 16 - *Algoritmo per l'estrazione di MFCC*

Gli MFCC a loro volta possono essere rappresentati come immagini che diventano, quindi, input per una CNN a scopo di classificazione con IA. Un esempio di immagine da MFCC è rappresentato in figura 17.

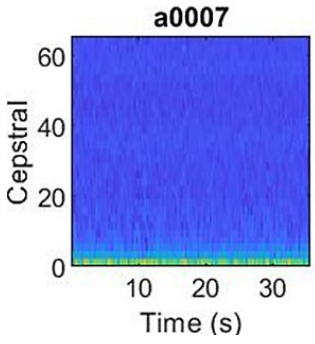

Figura 17 - *Esempio di immagine ottenuta rappresentando i coefficienti cepstral di uno spettrogramma di Mel in funzione del tempo*

La differente tonalità di colore è relativa alla diversa potenza del segnale al variare del tempo, fattore principale utilizzato dagli algoritmi di DL nella fase di addestramento della rete e classificazione dell'immagine per distinguere un segnale biologico normale da uno patologico.

3.2.2 Features dello spettrogramma GAMMATONE e immagine da GTCC

Lo spettrogramma Gammatone ed i relativi coefficienti *cepstral* alla frequenza Gammatone (GTCC), sono molto interessanti perchè risultano essere meno vulnerabili rispetto ad altri tipi di spettrogrammi a componenti di rumore eventualmente sovrapposti al segnale.
I coefficienti *cepstral* alla frequenza Gammatone (GTCC) vengono calcolati sottoponendo il segnale audio ad un filtraggio con banco di filtri composto da filtri Gammatone spaziati linearmente sulla scala ERB in un *range* di frequenze compreso tra 50 e 8000 Hz.
Le varie fasi di elaborazione per il calcolo dello spettrogramma Gammatone e delle sue *features* (i coefficienti GTCC) sono delineate in figura 18 e seguono la stessa logica dell'algoritmo visto nel calcolo dello spettrogramma Mel con le sue *features*, i coefficienti MFCC.

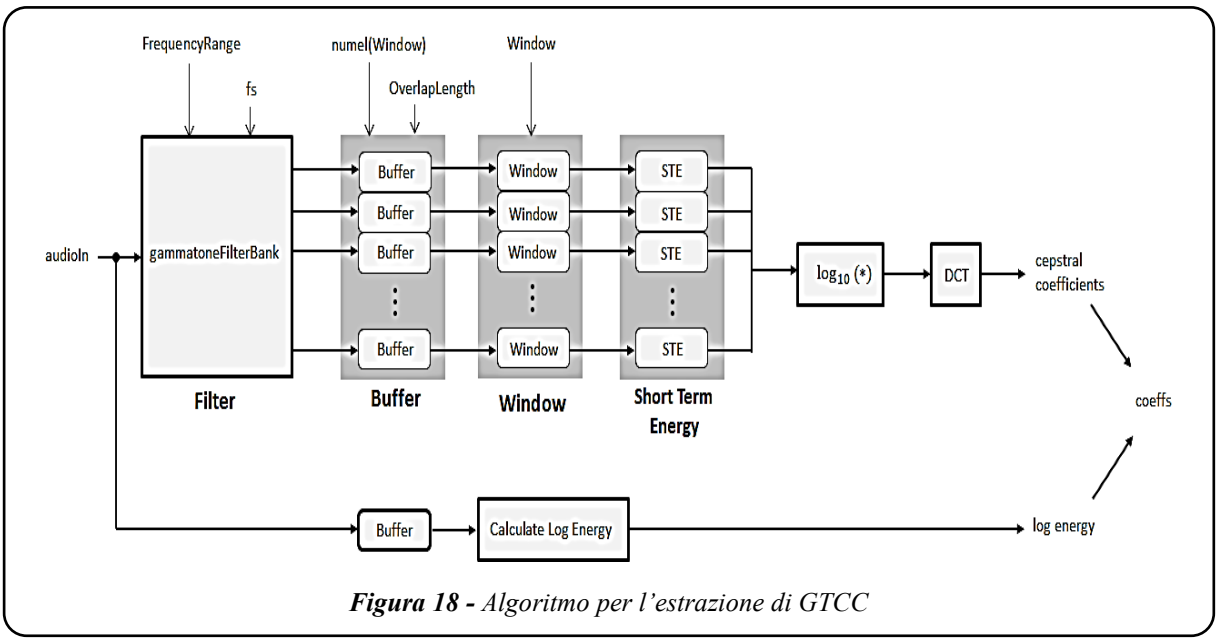

Figura 18 - *Algoritmo per l'estrazione di GTCC*

Un banco di filtri Gammatone viene spesso utilizzato come *front-end* di simulazione della coclea umana, trasformando suoni complessi in un modello di attività multicanale come quello osservato nel nervo uditivo, secondo la rappresentazione nel dominio della frequenza mostrata in figura 19.
La scala ERB definisce la spaziatura e la larghezza di banda dei filtri.

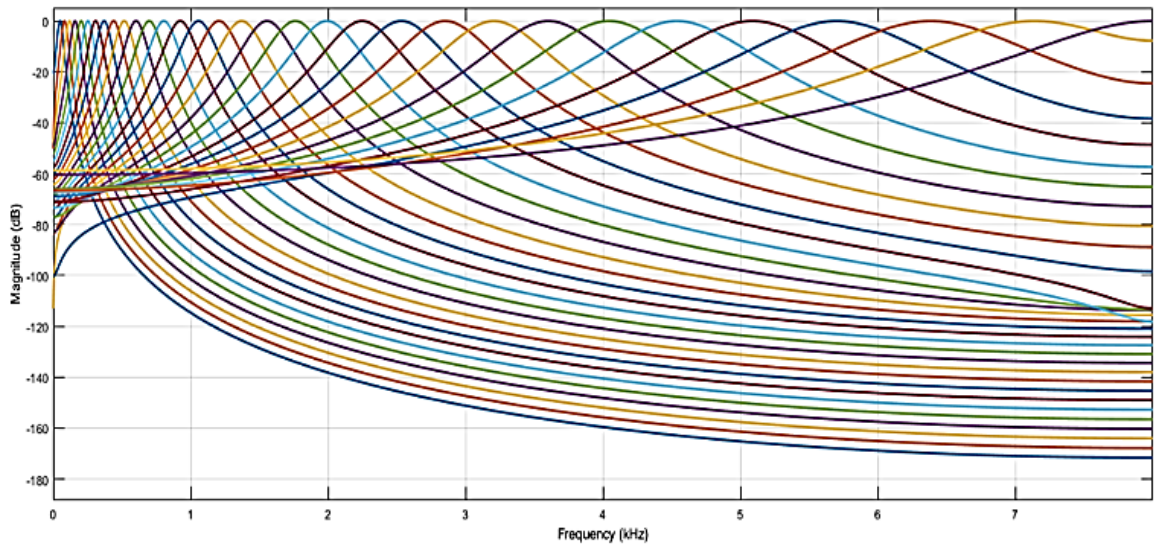

Figura 19 - *Gammatone filter bank*

L'uscita dal banco di filtri Gammatone è un segnale multicanale. Ogni canale di *output* dal banco di filtri viene bufferizzato in finestre di analisi sovrapposte e ne viene calcolata l'energia per ciascuna finestra di analisi.
Il segnale viene poi concatenato e fatto passare attraverso una funzione logaritmica e trasformata nel dominio *cepstral* utilizzando una trasformata discreta del coseno (DCT).
Anche per i coefficienti *cepstral* alla frequenza Gammatone (GTCC), *features* dello spettrogramma Gammatone, è possibile ottenere una rappresentazione in forma di immagine, nel dominio del tempo, come mostrato in figura 20, ed anche questa immagine può essere oggetto di classificazione tramite un modello di IA.

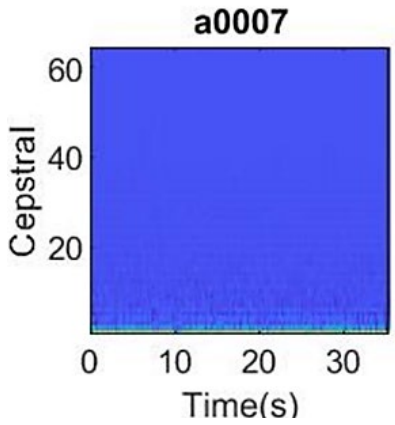

Figura 20 - *Esempio di rappresentazione nel tempo di coefficienti GTCC di un tono cardiaco*

3.2.3 Features dello scalogramma e immagini da CWT

Lo scalogramma, come già detto, è un grafico raffigurante il valore assoluto della CWT di un segnale audio, tracciato in funzione del tempo e della frequenza ed è particolarmente utile quando si desidera analizzare segnali con eventi di breve durata e ad alta frequenza e/o a bassa frequenza e di lunga durata.
Lo scalogramma si ottiene campionando il segnale con una *Window Length* di durata costante che viene spostata nel tempo e nella frequenza, a differenza dello spettrogramma in cui è fissa. Poiché lo spettrogramma utilizza una finestra costante, la risoluzione tempo-frequenza dello spettrogramma è fissa.
Per calcolare uno scalogramma è necessario innanzitutto campionare il segnale in segmenti sovrapposti e per ognuno di essi calcolare i coefficienti *Wavelet* costanti. In figura 21 è rappresentato un tipico scalogramma di segnale audio cardiaco con patologie.

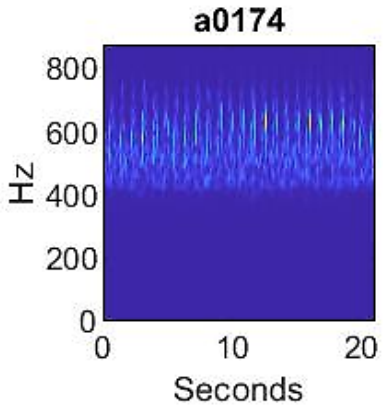

Figura 21 - *Scalogramma di segnale audio cardiaco con patologie valvolari. I coefficienti wavelet vengono calcolati processando il segnale con un banco di filtri*

In figura 22 è rappresentato il diagramma in frequenza relativo al banco di filtri utilizzati per calcolare lo scalogramma in ambiente Matlab, circa 10 filtri passa-banda per ottava.

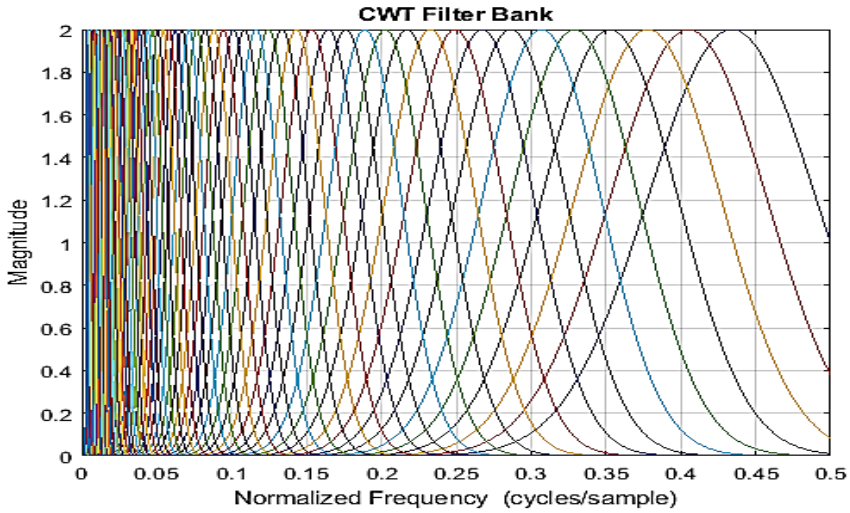

Figura 22 - *CWT filter bank*

Anche i coefficienti della CWT possono essere rappresentati come immagine, utile per essere classificata con strumenti di IA. In figura 23 è rappresentata l'immagine ottenuta dai coefficienti della CWT di un segnale audio cardiaco di un soggetto sano.

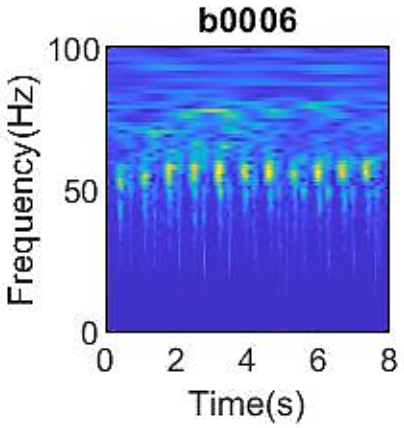

Figura 23 - *Immagine ottenuta rappresentando nel dominio tempo-frequenza i coefficienti della CWT di un soggetto sano*

3.3 Tecniche a confronto

Le tecniche presentate per la trasformazione di segnali audio in immagini risultano per certi aspetti simili tra loro, in quanto indicano diverse modalità di rappresentazione tempo-frequenza di un segnale audio; tuttavia, presentano delle differenze che ne consigliano per ciascuno uno specifico campo di applicazione.
Lo spettrogramma Mel e i suoi coefficienti MFCC sono tra le tecniche più utilizzate perché dimensionate sulla base della sensibilità uditiva umana. Tuttavia, presentano dei limiti legati alla bassa efficienza dei filtri Mel nell'eliminare il rumore additivo, soprattutto quello presente nei segnali audio del parlato e quindi in applicazioni di *speech recognition*; il problema è meno importante se si tratta di segnali biologici come i toni cardiaci ed il segnale respiratorio.
Il problema del rumore in applicazioni di *speech recognition* può essere risolto ricorrendo allo spettrogramma Gammatone e i suoi coefficienti GTCC, i quali riproducono meglio il comportamento della membrana della coclea umana, compreso il filtraggio delle frequenze in cui è collocato principalmente il rumore additivo.
I GTCC e il relativo spettrogramma Gammatone risultano quindi più adatti nell'identificazione e riconoscimento vocale, gli MFCC e il suo spettrogramma Mel invece, sono adatti per classificare segnali audio generici ma poco rumorosi ovvero battiti cardiaci e segnali biologici in generale, all'interno dei quali il comportamento al variare del tempo è ben definito e il campionamento risulta essere più facile, considerando anche che il rumore additivo risulta essere molto più attenuato rispetto agli altri perché si presume che siano acquisiti con hardware dedicato e con basso rumore.
L'altra tecnica presa in esame è quella relativa allo scalogramma e i suoi coefficienti *Wavelets*, attraverso la quale è possibile avere un buon compromesso tra risoluzione nel tempo e nella frequenza. L'analisi *Wavelet*, infatti, permette di elaborare informazioni con migliore risoluzione rispetto ad altre tecniche in quanto la finestra temporale di analisi non è fissa e permette di captare al meglio segnali audio con lunghi intervalli temporali a basse frequenze e intervalli temporali molto brevi ma ad alte frequenze. Tale tecnica sembra perciò particolarmente indicata per la classificazione di segnali respiratori dove si rilevano, soprattutto se patologici, suoni che si sovrappongono al murmure vescicolare, di durata breve e a frequenze elevate o prolungati e a frequenze più basse, come i *whezees* e i crepitii.
Le immagini ottenute con ciascuno di questi 6 metodi possono essere classificate tramite l'impiego di CNN ovvero con algoritmi di DL.
Pertanto, sono stati implementati e confrontati tutti i 6 metodi descritti, al fine di stabilire quale possa essere il più indicato per la classificazione del segnale prodotto dalla tosse da COVID-19 distinguendolo dalla tosse causata da altre patologie, al fine di poter formulare una diagnosi precoce attendibile di focolaio di polmonite da COVID-19.
Le immagini ottenute dai 6 algoritmi sono state classificate utilizzando più CNN al fine di determinare non solo il metodo di trasformazione del segnale audio in immagini maggiormente efficace ai fini di una accurata classificazione, ma anche il tipo di rete preaddestrata in grado di fornire una migliore accuratezza con l'utilizzo della tecnica del TL.

4. Classificazione di segnali audio biologici con modelli di IA

Riassumendo, i metodi di trasformazione dei segnali audio in immagini implementati e confrontati tra loro sono:

- Spettrogramma Mel
- Spettrogramma Gammatone
- calogramma (CWT)
- Immagine dei coefficienti MFCC
- Immagine dei coefficienti GTCC
- Immagine dei coefficienti CWT

I relativi algoritmi di elaborazione possono essere sintetizzati nello schema in figura 24

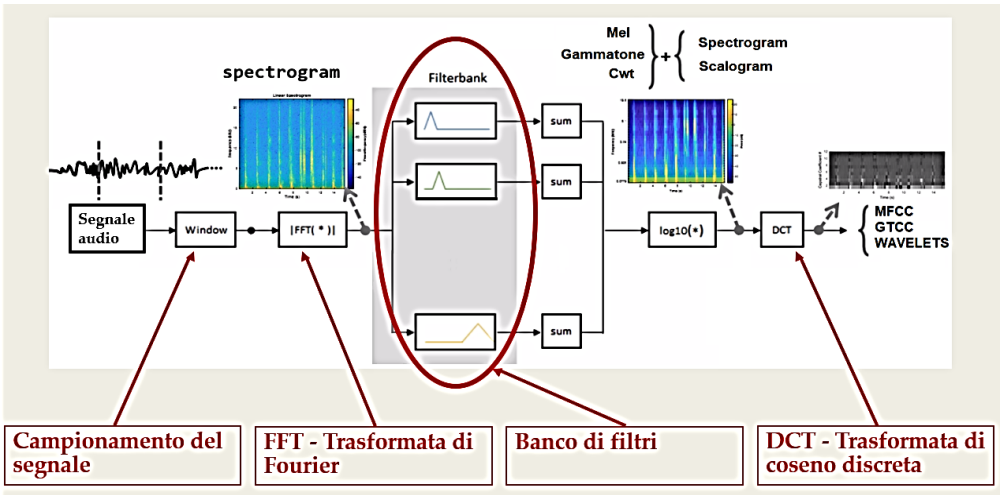

Figura 24 - *Rappresentazione sintetica degli algoritmi per la trasformazione di segnali audio in immagini. In base al tipo di banco di filtri utilizzati si ottiene spettrogramma Mel, spettrogramma Gammatone, CWT con scalogramma. Tramite il blocco DCT, in base al tipo di filtraggio eseguito a monte, si ottengono rispettivamente MFCC, GTCC, coefficienti della CWT*

Il primo database elaborato è di 3240 file audio di suoni cardiaci (i cosiddetti toni cardiaci), acquisiti tramite la tecnica della fonocardiografia (PCG) convertiti in immagini e successivamente dati come *input* alla rete neurale sia per l'addestramento che per il test di classificazione. Di questi file audio, 2574 sono relativi a toni cardiaci normali mentre 666 a toni cardiaci patologici. Il relativo database è disponibile online [30].
Successivamente, utilizzando un database [31] sia di toni cardiaci (817 normali e 183 patologici) che di suoni respiratori (35 normali e 885 patologici) è stato effettuato il confronto delle prestazioni tra varie reti preaddestrate, personalizzate con la tecnologia del TL, trasformando i file audio in spettrogrammi e scalogrammi.

4.1 Risultati e confronti

La prima rete preaddestrata utilizzata e riaddestrata secondo la tecnica del TL in ambiente Matlab, è GoogLeNet di cui sono stati modificati alcuni *layer* attraverso il *Deep Network Designer* del Matlab prima di procedere con l'addestramento.
È stato modificato il primo *layer* in quanto la dimensione dell'immagine dell'*imageInputLayer* di default della rete ha dimensione 224x224x3, invece le immagini da analizzare hanno dimensione 227x227x3; dopo di che è stato modificato il *fullyConnectedLayer* impostandolo a 2, in quanto le classi di uscita sono due, ovvero *normal* e *abnormal* (patologico), per lo stesso motivo è stato modificato anche l'ultimo *layer*, relativo al *classificationLayer*.
Infine, la rete è stata addestrata accedendo alle *training Options*, impostando un valore di *learning Rate* pari a 0.0001 e numero di *epoch* pari a 6.
Tale procedura è stata applicata per tutti i 6 metodi di trasformazione del segnale audio in immagini precedentemente descritti.
Dai risultati ottenuti un primo confronto può essere effettuato tra le tecniche relative alle rappresentazioni tempo-frequenza: spettrogramma Mel, spettrogramma Gammatone e scalogramma. Di seguito, in figura 25, sono mostrate le immagini relative allo stesso file audio derivante dall'acquisizione di toni cardiaci, preso come esempio e trasformato secondo ciascuno di questi tre algoritmi.

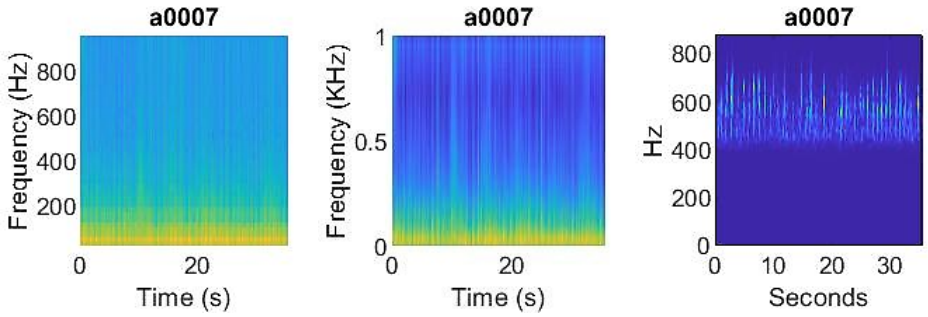

Figura 25 - *Spettrogramma Mel, Gammatone e scalogramma del tono cardiaco normale*

E' possibile notare che lo spettrogramma Mel (a sinistra in figura) rappresenta meglio questo tipo di segnale audio, qual è il tono del battito cardiaco, sin dalle basse frequenze, che sono le più importanti ai fini della corretta classificazione perché posseggono il maggiore contenuto informativo. Questo è dovuto principalmente al banco di filtri Mel che esalta queste componenti di segnale.
Un ulteriore confronto può essere effettuato estraendo le *features* dalle precedenti immagini, ovvero gli MFCC, GTCC ed i coefficienti Wavelets, e rappresentandole a loro volta in forma di immagini, come mostrato in figura 26.

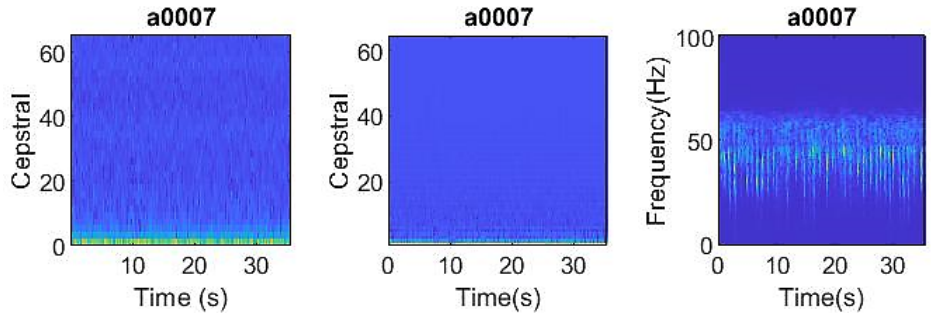

Figura 26 - *MFCC, GTCC e coefficienti della CWT del tono di un battito cardiaco normale*

È possibile notare che l'immagine ottenuta dagli MFCC (a sinistra) mostra variazioni di tonalità più marcate prestandosi, quindi, meglio alla classificazione tramite DL. Anche l'immagine ottenuta dai coefficienti *wavelets* (a destra) presenta una buona variazione della tonalità dei colori mentre le variazioni meno accentuate sono quelle presenti nella immagine ottenuta dai GTCC.
In tabella II sono confrontate le prestazioni, in termini di accuratezza e perdita per ogni *epoch*, della CNN basata sull'utilizzo della GoogleNet ed addestrata con il metodo del TL, relativamente ad ognuna delle tecniche di trasformazione di suoni in immagini sopra descritte.

Tabella 2. relativa al confronto tra le tecniche implementate con rete GoogleNet riaddestrata con TL

TECNICA UTILIZZATA	ACCURATEZZA	PERDITA
SPETTROGRAMMA MEL	**93 %**	**1.1**
SPETTROGRAMMA GAMMATONE	86.63 %	1.5
SCALOGRAMMA	89 %	3.5
MFCC	**93 %**	**1.2**
GTCC	91 %	1.3
WAVELETS	90 %	1.7

Come visibile in tabella e già preannunciato tramite analisi qualitativa, nella classificazione dei toni cardiaci la tecnica relativa allo spettrogramma Mel presenta un valore dell'accuratezza maggiore delle altre due tecniche ed un valore della perdita più basso, quindi tra le tre tecniche relative alle rappresentazioni tempo-frequenza questa risulta la più accurata e affidabile. Per lo stesso motivo tra le tecniche di estrazioni *features*, i coefficienti *cepstral* alla frequenza Mel (MFCC) risultano offrire un'accuratezza maggiore degli altri due (GTCC e coefficienti della CWT).
Successivamente, utilizzando un database [31] sia di toni cardiaci (817 normali e 183 patologici) che di suoni respiratori (35 normali e 885 patologici) è stato effettuato il confronto

delle prestazioni tra varie reti preaddestrate, personalizzate con la tecnologia del TL, trasformando i file audio in spettrogrammi e scalogrammi.
Le reti confrontate in ambiente Matlab sono state: GoogLeNet, SqueezeNet, AlexNet e ResNet50.
Il confronto delle prestazioni è mostrato in Tabella III
L'accuratezza maggiore (99.5%), per quanto riguarda il dataset dei toni cardiaci, è stata riscontrata nella rete GoogLeNet addestrata con *l'imagedatastore* degli spettrogrammi.

Tabella 3. Confronto delle prestazioni tra diverse reti preaddestrate e riaddestrate con il metodo del TL per classificare due tipi di suoni biologici: toni cardiaci e suoni polmonari

CNN (Convolutional Neural Network)	Validation accuracy spettrogrammi mel battiti cardiaci	Validation accuracy scalogrammi battiti cardiaci	Validation accuracy spettrogrammi mel respiri	Validation accuracy scalogrammi respiri
GoogLeNet	99.5%	97%	95.65%	100%
SqueezeNet	98.5%	98.59%	96.2%	100%
ResNet50	95%	99%	96.74%	100%
AlexNet	96.33%	97.67%	96.36%	100%

In figura 27 e 28 rispettivamente sono riportati i grafici dell'accuratezza e delle perdite rilevati durante l'addestramento della GooGleNet tramite TL con gli scalogrammi dei suoni respiratori (figura 27) e con gli spettrogrammi Mel dei toni cardiaci (figura 28), rispettivamente.

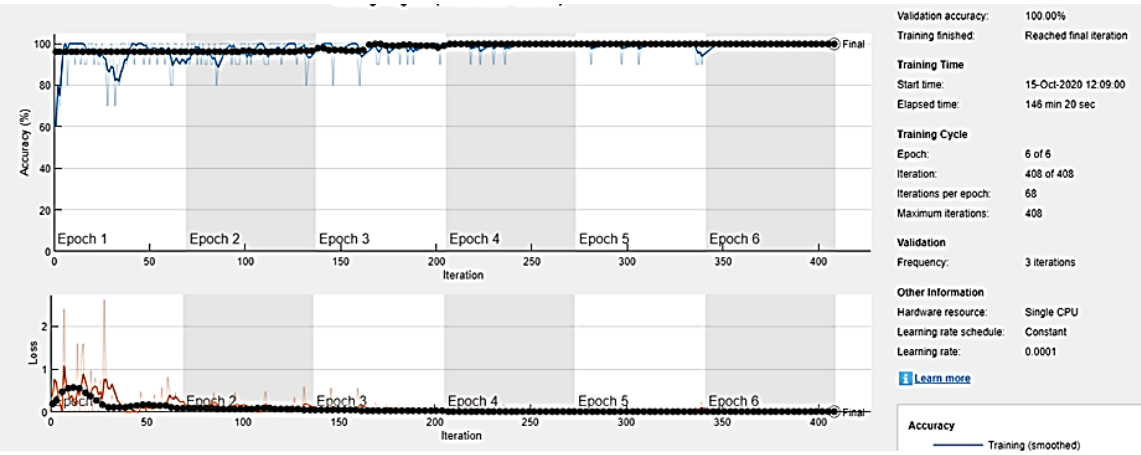

Figura 27 - *Accuratezza e perdite della CNN GoogLeNet in fase di addestramento con TL con gli scalogrammi dei suoni respiratori*

Per quanto riguarda invece il dataset dei suoni respiratori, l'accuratezza maggiore (100%) è stata riscontrata in tutte e 4 le reti addestrate con l'imagedatastore degli scalogrammi.

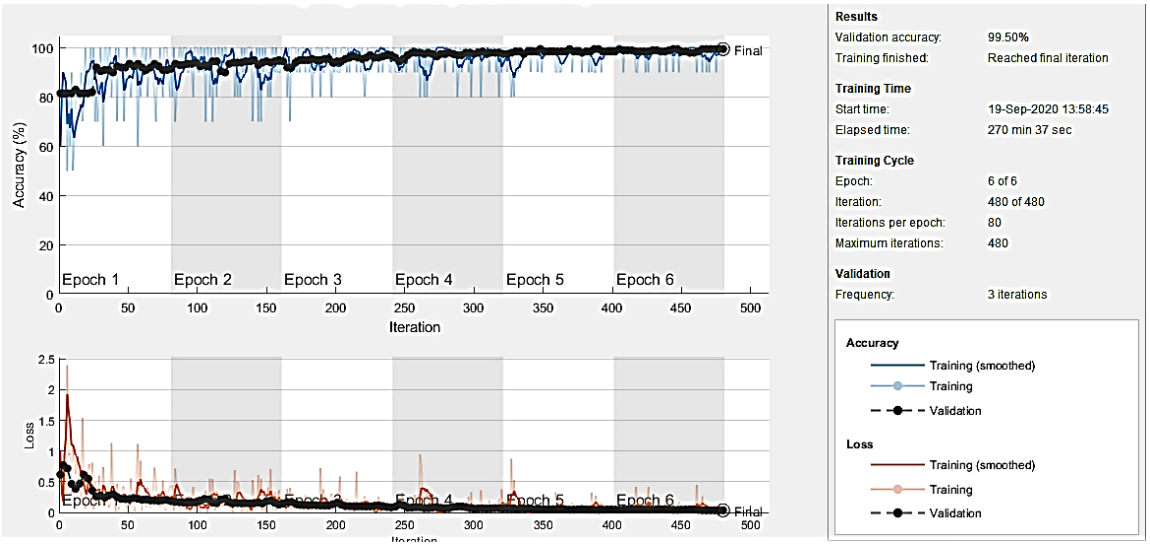

Figura 28 - *Accuratezza e perdite della rete CNN GoogLeNet in fase di addestramento con TL con gli spettrogrammi dei toni cardiaci*

Dai risultati ottenuti si evince molto chiaramente che lo spettrogramma Mel, i coefficienti MFCC, lo scalogramma ed i coefficienti *wavelets* sono i metodi di elaborazione dei suoni che meglio si prestano agli algoritmi di IA per riconoscere con elevato grado di confidenza i suoni biologici provenienti dall'apparato respiratorio classificandoli come normali o patologici, e che la rete GoogLeNet è la CNN preaddestrata, tra quelle prese in considerazione, che offre il più alto grado di confidenza nella classificazione.
Affinchè questa procedura possa essere applicata alla diagnosi precoce della polmonite da COVID-19 è necessario che si raccolga una quantità di file audio che sia la più grande possibile per applicare il metodo del TL per un addestramento fine di una delle CNN – presumibilmente la GoogleNet – al riconoscimento della tosse da COVID-19.
Studi preliminari [32] su un limitato numero di campioni dimostrano un'accuratezza non inferiore all'80% che, alla luce dello studio presentato in questo lavoro, ha potenziali ampi margini di incremento almeno sino al 90% ed oltre.

5. Sviluppo delle app per smartphone

La CNN addestrata tramite TL può essere implementata in modelli di IA eseguibili sia su PC che su smartphone permettendo così di sfruttare al meglio per scopi diagnostici di rilevante importanza le potenzialità computazionali e grafiche degli smartphone.
Per sviluppare app eseguibili sia in ambiente Android che in ambiente iOS occorre ovviamente essere esperti programmatori; tuttavia, almeno per uno sviluppo finalizzato alla verifica dell'idea ed al debug dell'algoritmo, ci sono comodi strumenti di traduzione di codice che permettono una programmazione di alto livello, spesso visuale, dunque abbastanza semplice, che non richiede conoscenze particolarmente approfondite dei linguaggi di programmazione. Questo è il caso dell'ambiente Matlab/Simulink [21, 22]. In particolare, il Simulink offre un tool di sviluppo di app per smartphone molto interessante ed utile, a partire dal *design* di un

modello a blocchi funzionali (dunque non direttamente scritto in linguaggio di programmazione) che poi viene automaticamente tradotto e convertito in app eseguibile su smartphone, come descritto di seguito.

A. Importazione della rete addestrata

Ci sono due modi in ambiente Simulink per implementare algoritmi di IA: uno è l'utilizzo del blocco funzionale *Image Classifier*, in cui si importa una CNN addestrata con TL dal tool di progetto delle reti neurali, ed esportata come modello; l'altro modo è l'utilizzo di un blocco funzionale Matlab personalizzato in cui viene inserito il codice per l'esecuzione di una CNN precedentemente addestrata ed esportata come *Compact Model*.
In ingresso al blocco funzionale, che sia l'uno o l'altro, viene fornita l'immagine da classificare (spettrogramma, scalogramma, immagine da *features*), in uscita si ottiene la classificazione (prediction) del dato fornito in ingresso ovvero segnale audio sano/patologico nel nostro caso.
Ovviamente per poter classificare i suoni occorre innanzitutto acquisirli.

B. Acquisizione audio

L'operazione di acquisizione dell'audio dei segnali biologici può avvenire tramite microfono dello smartphone o, qualora questo non risultasse di qualità adeguata, si può utilizzare un microfono esterno collegato allo smartphone. Il modello Simulink utilizzato allo scopo e direttamente traducibile in app per Android è mostrato in figura 29. Analogo modello potrebbe essere realizzato per traduzione in app per iOS.

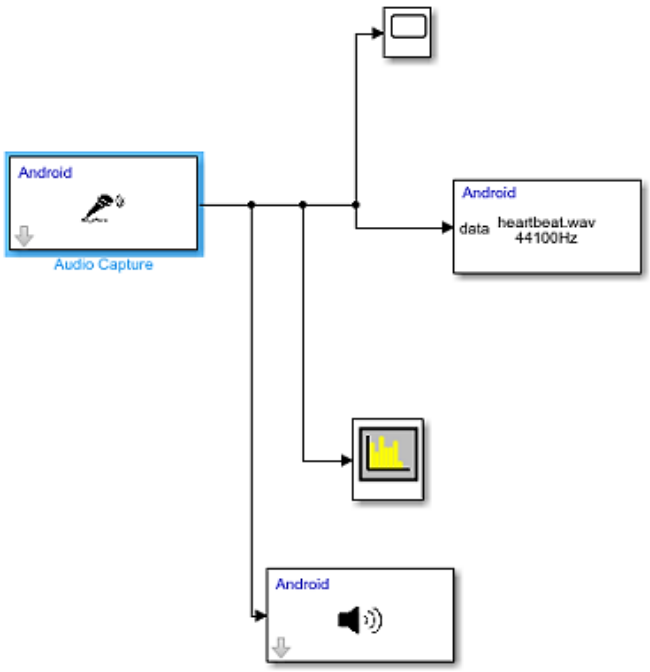

Figura 29 - *Modello Simulink per l'acquisizione audio*

Esso viene disegnato utilizzando il blocco funzionale *Audio Capture*, disponibile nella libreria del *Simulink Support Package for Android/iOS Devices*, che ci permette di acquisire l'audio dal microfono dello smartphone. La frequenza di campionamento scelta è di 44.1 kHz. In parallelo vengono inseriti i blocchi *Audio Playback* (per riprodurre sullo smartphone l'audio che si sta acquisendo), *Spectrum Analyzer* e *Time Scope*. Questi ultimi ci permettono di analizzare il segnale rispettivamente nel dominio della frequenza e del tempo simultaneamente all'acquisizione sul display dello smartphone. Infine, il blocco *Audio File Write* ci permette di salvare l'audio in formato *wav* sullo smartphone per eventuale, successivo, *post-processing*.
Il funzionamento di questo modello Simulink quando viene tradotto in app ed eseguito su smartphone Android è mostrato in figura 30.

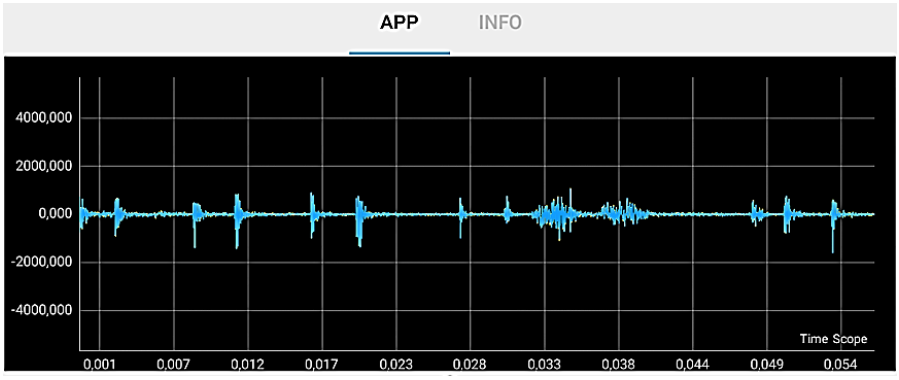

Figura 30 - *Screenshot dell'app progettata tramite Simulink, in acquisizione*

C. Trasformazione del file audio in immagine

Ogni file audio da classificare viene convertito in immagini tramite un ulteriore blocco funzionale appositamente sviluppato in Matlab ed inserito nel modello Simulink: in base alle considerazioni precedenti risulta più adatto l'uso dello spettrogramma di Mel e l'estrazione dei coefficienti MFCC per i toni cardiaci e l'uso dello scalogramma con i coefficienti della CWT per i suoni respiratori.

D. Classificazione con rete CNN

Si procede, infine, con l'inclusione della CNN addestrata con il TL ed esportata, in un blocco *Image Classifier* ovvero in un blocco funzionale *custom* di tipo *Matlab Function*, a completamento del modello, che viene poi tradotto in app installata ed eseguita su smartphone. Il modello risultante è mostrato in figura 31.

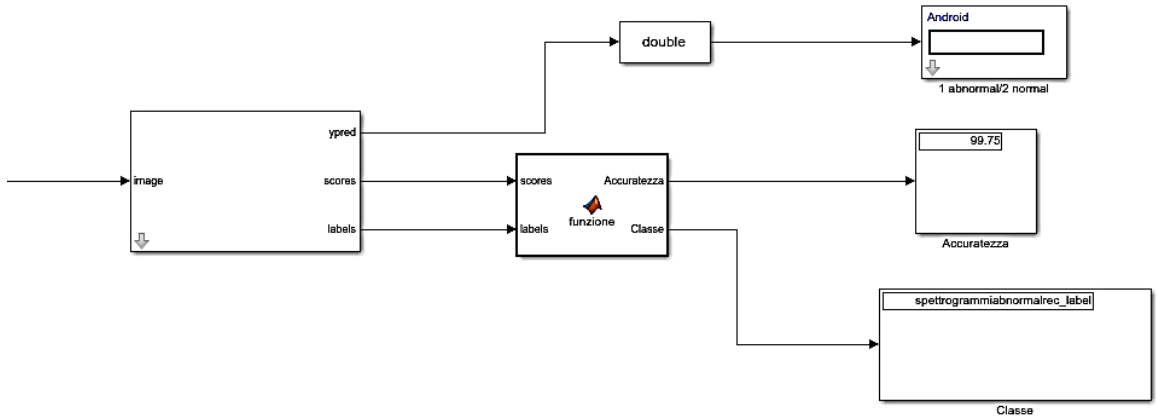

Figura 31 - *Modello simulink per la classificazione dei suoni biologici, tradotto in app ed eseguito su smartphone*

In questo modo si dispone di un'app in grado di monitorare automaticamente e continuamente lo stato di salute e di favorire una diagnosi precoce del COVID-19 a partire dai primi colpi di tosse.

6. Conclusioni e sviluppi futuri

Lo studio proposto in questo lavoro ha lo scopo di dimostrare come strumenti di IA possono essere applicati per uno degli obiettivi più importanti del trattamento del COVID-19 ovvero la diagnosi precoce della polmonite interstiziale. Tale diagnosi permetterebbe una riduzione dei ricoveri, un aumento della probabilità di sopravvivenza potendo intraprendere tempestivamente terapie adeguate ed una riduzione del rischio di contagio perché sarebbero individuati precocemente anche soggetti pauci-sintomatici.
La possibilità in termini di strumenti ingegneristici disponibili allo scopo è stata dimostrata, trattandosi peraltro di strumenti facilmente implementabili anche su smartphone tramite app dedicate che avrebbero, quindi, anche un costo assai accessibile.
Il passo ulteriore per completare questo studio con il progetto di un algoritmo finito di classificazione altamente affidabile è quello di disporre di un nutrito database di registrazioni della tosse di pazienti con di diverse patologie, incluso il COVID-19 ovviamente, perché indispensabile per un addestramento della rete neurale che consenta di ottenere livelli di confidenza della classificazione che siano il più possibile elevati, tendenti al 100%.
Per questo sarà necessario organizzare progetti coordinati tra gruppi di lavoro di medici e di ingegneri che speriamo possa essere attuata nel più breve tempo possibile.

Riferimenti bibliografici

[1] Grotberg, J. B. (2019). Crackles and Wheezes: Agents of Injury? Annals of the American Thoracic Society https://doi.org/10.1513/AnnalsATS.201901-022IP.
[2] https://www.healthline.com/health/breath-sounds
[3]http://www.scienzaegoverno.org/book/export/html/2144?fbclid=IwAR2kSmHDaxVVqkIjaKFtsGdTJ7Gtvo_5CYsZu5LaJ1krKTPJb2fqIjKRaFo
[4] https://www.facebook.com/100000562225270/videos/3255442694484439/
[5] https://global.techradar.com/it-it/news/covid-19-individuare-gli-asintomatici-con-smartphone-e-deep-learning-e-possibile?fbclid=IwAR3TvMaJb0-azObUUCbp_FhiAVgsiEb0URdmTMabZZoKRJNSUaor3VZGTRs
http://www.scienzaegoverno.org/book/export/html/2144?fbclid=IwAR1e-1JYZ4HiwJHRqsX41ycY8_1QKGUKFA1eG-9ew4IInpy-Wxhxhbdg3L0
[6] https://biomedicalcue.it/app-riconoscere-tosse-covid-19/22717/?fbclid=IwAR2SjT_iGlGa-7mpaaKpltVdu4ETdsmGeJKhvYPPjANJLHK_e7GjeBhvVZA
[7] https://it.mathworks.com/discovery/neural-network.html
[8] Speech Command Recognition using Deep Learning, Mathworks: https://www.mathworks.com/help/deeplearning/ug/deep-learning-speech-recognition.html
[9] Introduction to Deep Learning for Audio and Speech Applications, Webinar by Gabriele Bunkheila, MathWorks: https://www.mathworks.com/videos/introduction-to-deep-learning-for-audio-and-speech-applications-1560448385032.html
[10] Machine Learning and Deep learning for Audio, MathWorks: https://www.mathworks.com/help/audio/feature-extraction-and-deep-learning.html
[11] https://it.mathworks.com/discovery/deep-learning.html
[12] https://it.mathworks.com/discovery/machine-learning.html
[13] Deep Learning Onramp and Deep Learning with Matlab,
https://it.mathworks.com/learn/tutorials/deep-learning-onramp.html .
[14] Deep Learning for Signals and Sound, Webinar by Johanna Pingel and Emelie Andersson, MathWorks: https://www.mathworks.com/videos/deep-learning-for-signals-and-sound-1544467789023.html
[15] Deep Learning for Speech and Audio Processing with NVIDIA GPUs, Webinar by Gabriele Bunkheila, MathWorks: https://www.mathworks.com/videos/deep-learning-for-speech-and-audio-processing-with-nvidia-gpus-1586524417560.html
[16] Deep Learning Toolbox, Mathworks: https://www.mathworks.com/products/deep-learning.html
[17] Get Started with Transfer Learning, Mathworks: https://www.mathworks.com/help/deeplearning/gs/get-started-with-transfer-learning.html
[18] Transfer Learning with Deep Network Designer, Mathworks: https://www.mathworks.com/help/deeplearning/ug/transfer-learning-with-deep-network-designer.html

[19] https://it.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html#mw_45a8c0b2-26fa-48e9-905a-a7ed7b87bfc8
[20] M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K.Wilson. 2017. CNN architectures for large-scale audio classification. In Proc.IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).
131–135.
[21] Deep Learning in Simulink using Deep Neural Networks library, Mathworks: https://www.mathworks.com/help/gpucoder/ug/deep-learning-in-simulink-using-deep-neural-networks-library.html
[22] Getting Started with Android Devices, Mathwork: https://www.mathworks.com/help/supportpkg/android/examples/getting-started-with-android-devices.html
[23] Mel Spectrogram, Mathworks: https://www.mathworks.com/help/audio/ref/melspectrogram.html
[24] MFCC, Mathworks: https://www.mathworks.com/help/audio/ref/mfcc.html
[25] Gammatone filter bank, Mathworks: https://www.mathworks.com/help/audio/ref/gammatonefilterbank-system-object.html
[26] GTCC, Mathworks: https://www.mathworks.com/help/audio/ref/gtcc.html
[27] CWT,Mathworks: https://www.mathworks.com/help/wavelet/ref/cwt.html
[28] Wavelets, Mathworks: https://www.mathworks.com/help/wavelet/ref/cwtfilterbank.wavelets.html?searchHighlight=wavelet&s_tid=srchtitle
[29] Wavelet Scattering, Mathworks: https://www.mathworks.com/help/wavelet/ref/waveletscattering.html
[30] https://physionet.org/content/challenge-2016/1.0.0/
[31] https://www.kaggle.com/vbookshelf/respiratory-sound-database
[32] Chloë Brown, Jagmohan Chauhan, Andreas Grammenos, Jing Han, Apinan Hasthanasombat, Dimitris Spathis, Tong Xia, Pietro Cicuta, and Cecilia Mascolo. "Exploring Automatic Diagnosis of COVID-19 from Crowdsourced Respiratory Sound Data." Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2020)