Università degli Studi di Milano Bicocca

Dipartimento di Informatica, Sistemistica e Comunicazione

Corso di Laurea Magistrale in Informatica

A Fine-Tuned Playlist Recommender System Based on Emotions

Relatore: Prof.ssa Francesca Gasparini

Tesi di Laurea Magistrale di:

Mattia Marchi

Matricola 817587

Anno Accademico 2021-2022

Sommario

In questo progetto di tesi viene presentato un nuovo approccio per la creazione di un sistema di raccomandazione e generazione di playlist, utilizzando modelli di Deep Learning per riconoscere le emozioni degli utenti associate alla musica. L'ambito di ricerca è definito *Music Emotion Recognition* (MER), sottocategoria dell'*Affective Computing*, branca dell'intelligenza artificiale che riconosce ed utilizza le emozioni umane.

Il modello proposto è composto da diverse tipologie di reti neurali profonde. Inizialmente si utilizza una rete neurale convoluzionale (CNN), facendo uso della tecnica definita *transfer learning* su un'architettura allo stato dell'arte per estrarre le informazioni più rilevanti da spettrogrammi di Mel, quest'ultimi generati da file musicali. Successivamente si impiega una rete Long Short-Term Memory (LSTM) per classificare correttamente le feature estratte. Le classi emozionali esaminate corrispondono ai quattro quadranti del modello di Russell. I dataset, impiegati nella realizzazione del modello di Deep Learning, sono stati selezionati dalla letteratura di riferimento al fine di considerare multiple fonti per migliorare la generalizzazione. Il risultato di accuratezza raggiunto dal modello finale è pari al 91%.

È stata inoltre studiata la capacità del modello finale di adattarsi alla percezione del singolo individuo, tramite la tecnica di *fine-tuning* sulle emozioni soggettive di ogni utente. Questo secondo modello prende il nome di "user-tuned". L'obiettivo principale è quello di ottenere un sistema inizialmente capace di generalizzare, minimizzando l'errore nell'uso da parte di nuovi utenti, e successivamente adattarlo al singolo utente e alle sue percezioni soggettive. Tale metodologia consente di accentuare il focus sul singolo soggetto, migliorando l'esperienza di utilizzo e risolvendo uno dei problemi principali dei sistemi di raccomandazione: il *cold start*. Il sistema è stato testato su tre soggetti, utilizzando 50 canzoni di diverso genere, mostrando un'accuratezza nel predire correttamente le emozioni del 44.68% per il modello generale. Il test effettuato sul modello user-tuned, considerando 10 e 20 emozioni suggerite al sistema, ha raggiungo un valore medio di predizioni corrette del 62.66% e 76% rispettivamente, migliorando fortemente i risultati ottenuti dal modello generale. I risultati empirici hanno inoltre dimostrato che il modello user-tuned riesce a perfezionare le inferenze delle canzoni non suggerite direttamente al sistema, abbassando l'errore di predizione medio.

Infine, è stato creato un sistema di raccomandazione, il quale utilizza le informazioni ottenute dai modelli precedentemente descritti per generare una playlist personalizzata in base all'umore attuale dell'utente ed uno stato emozionale target da raggiungere. La generazione viene effettuata in modo tale da creare un graduale passaggio emozionale, utilizzando un numero a scelta di canzoni della propria libreria, con l'obiettivo di migliorare l'esperienza finale dell'utente che utilizza il sistema.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Negli ultimi decenni si è assistito ad un'esplosione della quantità di dati digitali condivisi nel web. Gli utenti avvertono sempre più la necessità di sistemi che possano filtrare in modo automatizzato i contenuti, così che gli vengano proposti solo quelli di loro potenziale interesse. Con questo obiettivo, nei primi anni '90 sono stati sviluppati i primi *Recommender System* (RS) che hanno rivoluzionato l'e-commerce e la fruizione di contenuti da parte delle persone [9]. Dal momento della loro nascita fino ai giorni d'oggi, sono stati diversi gli sviluppi ed i contesti in cui tali sistemi hanno operato e continuano tuttora a farlo. Uno di questi campi è quello musicale. Oggigiorno, le persone trascorrono ore intere ad ascoltare la propria musica preferita. Il rapporto sulle abitudini del consumo musicale, realizzato dall'IFPI, relativo all'anno 2021 ha rivelato che il tempo a cui le persone di tutto il mondo dedicano all'ascolto della musica è in continuo aumento. Secondo *International Federation of the Phonographic Industry*, a livello globale si è raggiunta una media di ascolto settimanale pari a 18.4 ore, mentre localizzando l'analisi solamente all'Italia questo dato cresce fino a 19.1 ore [10]. Differenti studi hanno mostrato come l'ascolto della musica abbia un impatto decisivo sul cambiamento delle emozioni che le persone provano nel breve periodo e sulla loro capacità di affrontare situazioni difficili. Per esempio, Lehmberg *et al.*[11] hanno scoperto evidenza di diversi benefici che la musica dal vivo può portare alle persone anziane: potenziale incremento dell'aspettativa di vita, minore stress, miglioramento della situazione medica e complessivo aumento della felicità. Secondo uno studio congiunto dell'università di Durham in Inghilterra e dell'università di Jyväskylä in Finlandia [12] anche la musica più triste può portare alle persone conforto e appagamento. Nonostante ciò, lo stesso studio ha anche evidenziato come questo tipo di musica possa causare in alcuni soggetti sofferenza. Un'ulteriore studio del 2013 [13] ha portato evidenza del fatto che l'ascolto della musica allegra può portare ad un miglioramento complessivo dell'umore delle persone in sole due settimane.

Diverse ricerche nell'ambito della psicologia, applicata al contesto musicale, hanno confermato che la musica può indurre nell'ascoltatore una risposta emozionale molto forte [14][15][16]. Le emozioni e i sentimenti hanno suscitato fin dai tempi più antichi interesse ed attenzione da parte di filosofi, psicologi, dottori e ricercatori che lavorano in questi ambiti [17]. Lo studio delle emozioni è nato svariati secoli fa e con il passare degli anni ne sono state fornite molteplici definizioni. Gran parte delle teorie moderne definiscono le emozioni come un processo multicomponenziale, ovvero formato da distinte componenti in continua evoluzione nel corso del tempo. Alcuni studiosi descrivono le emozioni come forti sensazioni che nascono in seguito al verificarsi di esperienze, positive o negative [18]. Altri ricercatori invece, le caratterizzano come stati psicologici provocati da cambiamenti neuro-fisici associati a pensieri, sentimenti e risposte comportamentali [19].

Negli ultimi anni è nato un crescente interesse relativo allo sviluppo di sistemi che integrano le emozioni, al fine di garantire all'utente un'esperienza personalizzata più gratificante. La branca dell'intelligenza artificiale che studia il riconoscimento e l'utilizzo delle emozioni viene chiamata Affective Computing[20]. La nascita di tali tecniche, integrate in applicazioni musicali, ha portato allo sviluppo di sistemi di riconoscimento delle emozioni e di generazione di playlist basate sul mood attuale dell'utente.

1.1 Obiettivi della Tesi

In questo lavoro viene esplorata la possibilità di creare un sistema di raccomandazione e generazione di playlist basato sull'unione di diversi modelli di Deep Learning. L'idea alla base del progetto è quella di sfruttare alcuni dei dataset più completi presenti nella letteratura di riferimento, nell'ambito del riconoscimento emozionale applicato alla musica, al fine di progettare un modello efficace nell'effettuare predizioni emozionali, basandosi solo sulle feature audio delle canzoni. La scelta dei modelli di Deep Learning - chiamati reti neurali profonde - è stata effettuata in base alla capacità delle reti di sfruttare grandi quantità di dati per creare sistemi capaci di riconoscere pattern e feature rilevanti, ed utilizzare tali informazioni in modo tale da effettuare task predittivi o di regressione.

Il sistema in questione sarà in grado di classificare ed etichettare le canzoni dell'utente che lo utilizzerà, sulla base delle diverse emozioni che l'ascolto può indurre nel soggetto, riconosciute dalla rete neurale. L'architettura è progettata per essere applicata ad immagini generate dai file audio, chiamati spettrogrammi di Mel. La classificazione si baserà su due diversi modelli, studiati per essere utilizzati in diverse casistiche reali. Il primo - chiamato modello generale - sarà formato da un insieme di reti neurali, capaci di stimare che tipo di umore può evocare l'ascolto di una canzone per la maggior parte degli utenti, minimizzando il rischio di errore nella predizione. Questo aspetto risulta fondamentale per far sì che anche nuovi utenti, approcciandosi al sistema per la prima volta, riescano a trovarsi d'accordo con l'emozione proposta. Il secondo modello - chiamato "user-tuned" - sarà invece un adattamento di quello generale in grado di imparare dal singolo utente, durante l'utilizzo del sistema, addestrando la rete a considerare la sua per-

cezione soggettiva. Tale considerazione risulta di vitale importanza in un lavoro di affective computing, proprio perché le emozioni indotte dalla musica e la loro percezione sono esperienze estremamente soggettive e dunque risulta significativo considerare tale aspetto. La tecnica usata per adattare il modello generale all'utente, creando quello che è stato definito user-tuned, è chiamata fine-tuning. Tale tecnica consentirà ad ogni utente di definire molteplici nuovi esempi al modello, basati sulla propria percezione, così da addestrarlo alla soggettività emotiva del soggetto, andando quindi ad affinare ulteriormente il sistema di raccomandazione. Il sistema, come obiettivo finale, dovrà generare automaticamente delle playlist, basate sull'umore attuale dell'utente ed utilizzando canzoni in suo possesso, consentendo a quest'ultimo di definire uno stato emozionale target da raggiungere alla fine dell'ascolto.

1.2 Outline

Questo documento è composto da 8 capitoli, di cui segue una breve descrizione:

- Questo capitolo introduce l'ambito di studio della tesi, il progetto preso in esame e ne descrive brevemente la struttura;
- Il secondo capitolo richiama alcune nozioni teoriche necessarie alla comprensione del testo;
- Il terzo capitolo descrive i principali approcci esistenti, applicati al problema di riconoscimento emozionale musicale, descritti in letteratura;
- Il quarto capitolo descrive i dataset di riferimento utilizzati nello sviluppo dei modelli impiegati nel progetto. Viene inoltre esposta la fase di preprocessing dei dati;
- Il quinto capitolo descrive i modelli di deep learning progettati, fornendo alcuni dettagli tecnici di realizzazione ed analizzando i risultati ottenuti;
- il sesto capitolo pone il focus sul processo definito di user-tuning, descrivendone la metodologia e mostrandone il funzionamento ed i risultati empirici ottenuti;
- Il settimo capitolo chiude la descrizione del progetto descrivendo come i modelli ideati vengano utilizzati dal sistema di raccomandazione e generazione di playlist;
- L'ultimo capitolo conclude la tesi andando a riassumere i risultati ottenuti ed i possibili sviluppi futuri.

Capitolo 2

Quadro Teorico

In questo capitolo vi è un'introduzione di tutti i concetti fondamentali necessari a comprendere ciò che verrà trattato in questo lavoro.

2.1 Deep Learning

Il Deep Learning (DL) rappresenta una branca dell'intelligenza artificiale che studia modelli, basati su reti neurali, che contengono molteplici strati nascosti, creando una struttura definita "profonda". Tali reti si utilizzano per effettuare processi di apprendimento automatico, chiamato anche Machine Learning (ML), imparando feature rilevanti relative ad uno o più dataset, osservando una grande quantità di esempi forniti dai dati. Ai fini della tesi verranno presentate le due principali tipologie di reti implementate nel progetto: Convolutional Neural Network (CNN) e Long Short-Term Memory (LSTM).

2.1.1 Convolutional Neural Networks

Una Convolutional Neural Network rappresenta un'architettura di rete neurale profonda ideata per classificare immagini [21]. Tale architettura, ispirata alla biologia, consente di osservare porzioni locali di un'immagine, utilizzando dei filtri per imparare diverse *feature map*. Attraverso questo sistema la rete è capace di capire le relazioni tra i pixel ed a estrarre feature spaziali locali. Ogni strato convoluzionale è formato da dimensioni (altezza, larghezza, canali cromatici) ed un numero di filtri. Gli strati convoluzionali sono seguiti da una funzione di attivazione e un layer di pooling spaziale, usato per ridurre la dimensione dell'input. Negli anni, grazie alla ricerca e per merito di diversi contest come *ImageNet*, le reti convoluzionali hanno subito vasti miglioramenti alle performance, sviluppando reti sempre più complesse. Al fine di addestrare una CNN in un task di classificazione, viene aggiunta in coda una rete neurale Fully-Connected (FC) che, grazie alle feature ottenute dalla CNN, effettuerà la predizione. È possibile osservare quanto descritto in figura 2.1

[1]Source: `https://www.theclickreader.com/building-a-convolutional-neural-network/`

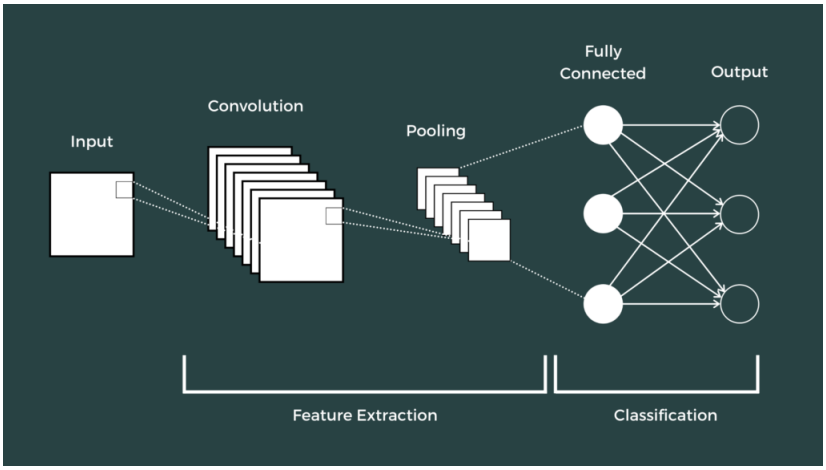


Figura 2.1: Struttura di una rete neurale convoluzionale.[1]

Per addestrare una rete CNN è necessario avere un dataset abbastanza numeroso ed una grande potenza di calcolo, pertanto è raro che si utilizzi una rete creata ad hoc. La scelta più popolare è quella di utilizzare il processo chiamato *tansfer learning*. Questa metodologia consente di usufruire di reti pre-addestrate e adattarle al task del dominio specifico. Esistono due diversi tipi di transfer learning:

- **Fine-Tuning**: i pesi vengono mantenuti bloccati o viene utilizzato un learning rate molto basso per non variare troppo i pesi originali. La rete fully connected finale viene invece sostituita per ottenere una nuova rappresentazione, così da poterla addestrare mantenendo i pesi della CNN.
- **Feature Extractor**: si taglia la rete all'ultimo layer convoluzionale e si ottiene un vettore di feature, chiamato *descriptor*, il quale verrà utilizzato in un nuovo classificatore.

2.1.2 Long Short-Term Memory

La famiglia di reti neurali specializzate nel processare dati sequenziali è chiamata Recurrent Neural Network (RNN) [22]. Queste reti vengono chiamate ricorrenti perché elaborano i dati come sequenze ed ogni output dipende dalla computazione dello stato precedente. La sotto-classe delle RNN più utilizzata al giorno d'oggi è quella chiamata Long Short-Term Memory (LSTM)[23]. Tale rete è formata da diversi stati interni e *gate*, i quali permettono di controllare il flusso delle informazioni. I gate inoltre consentono al modello di dimenticarsi delle informazioni di poco rilievo, mantenendo solo quelle rilevanti. Nel dettaglio, una rete LSTM è composta dai seguenti componenti:

- **Cell State**: prende in input l'output della sequenza precedente e, in base ai gate, mantiene o cancella tale informazione;

- **Forget Gate**: utilizza una funzione sigmoide, prende in input la concatenazione dell'output della sequenza precedente e l'input della sequenza attuale, determinando se l'informazione vecchia debba essere mantenuta o cancellata;
- **Input Gate**: determina se l'input della sequenza attuale verrà aggiunto al cell state;
- **Output Gate**: lo stato finale del cell state viene filtrato da una funzione tangente $(-1, 1)$, determinando il nuovo output.

La struttura di una rete LSTM è mostrata in figura 2.2.

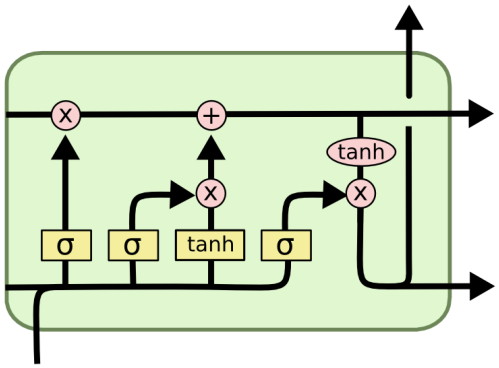


Figura 2.2: Struttura a gate di una rete LSTM.[2]

2.2 Le Emozioni

Diversi ricercatori e psicologi hanno definito alcuni modelli per classificare le emozioni, fondati su diverse teorie. Tuttavia, in letteratura esistono principalmente due famiglie di approcci con cui è possibile descrivere lo stato emotivo di una persona. Il primo di questi prevede di suddividere le emozioni in uno spazio discreto, formando alcuni gruppi di aggettivi, i quali rappresentano ciascuno un'emozione. Tale modello assume l'esistenza di un insieme limitato di categorie emozionali distinte. Uno dei modelli più rilevanti che appartiene a questa categoria è quello descritto da Plutchik [24], che divide le emozioni in 8 categorie rappresentate mediante la cosiddetta "*wheel of emotions*", mostrata in figura 2.3. L'intensità delle emozioni cresce dall'esterno verso il centro della ruota. Più è scuro il colore, maggiore è l'intensità dell'emozione.

[2]Source: `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`
[3]Source: `https://commons.wikimedia.org/wiki/File:Plutchik-wheel.svg`

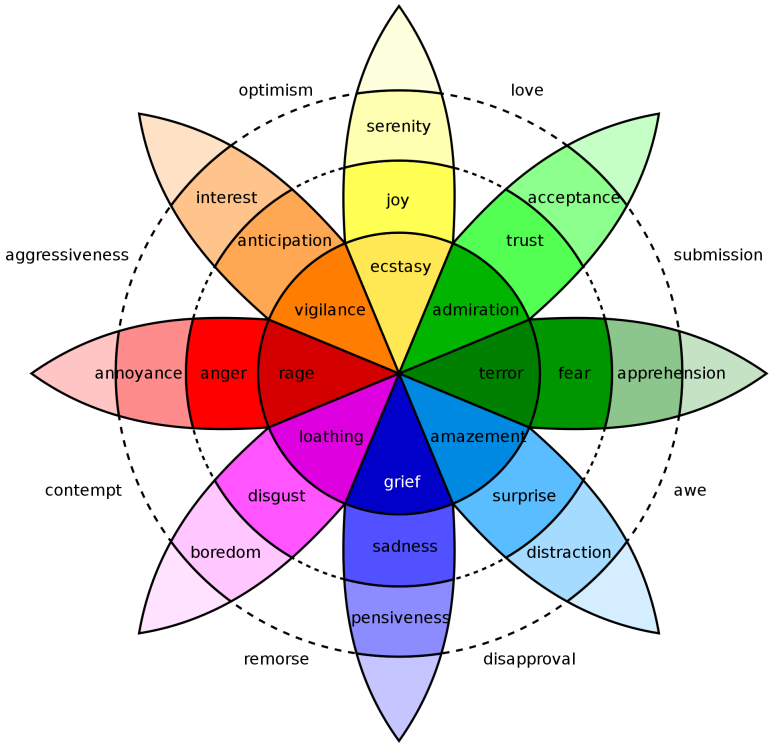


Figura 2.3: *Wheel of emotions* proposta da Plutchik.[3]

La seconda famiglia di modelli segue un approccio dimensionale. Tale approccio descrive ogni emozione come un punto in uno spazio continuo ed ogni dimensione rappresenta una aggettivo che caratterizza l'emozione stessa. Il modello più frequentemente utilizzato appartenente a questa categoria è quello proposto da James Russell [25], chiamato *Circumplex Model*. Russell suggerisce che le emozioni siano distribuite in uno spazio circolare bidimensionale, il quale è composto da due dimensioni: attivazione - definito come *arousal* o *activation* - che indica l'intensità dell'emozione percepita e la valenza - o *valence* - che esprime la polarità dell'emozione come positiva o negativa. L'arousal rappresenta l'asse verticale, mentre la valence rappresenta l'asse orizzontale. Ogni stato emozionale può essere rappresentato come un punto in questo spazio bidimensionale. Questa classificazione ha il vantaggio, tramite l'utilizzo di uno spazio continuo, di poter utilizzare valori compresi tra $[0, 1]$ per esprimere ogni possibile tipologia di emozione come una coordinata sul piano. È inoltre possibile inserire emozioni categoriche in questo spazio bidimensionale, così da poter effettuare una conversione tra i due modelli descritti. Una rappresentazione del modello proposto da Russell è visibile in figura 2.4.

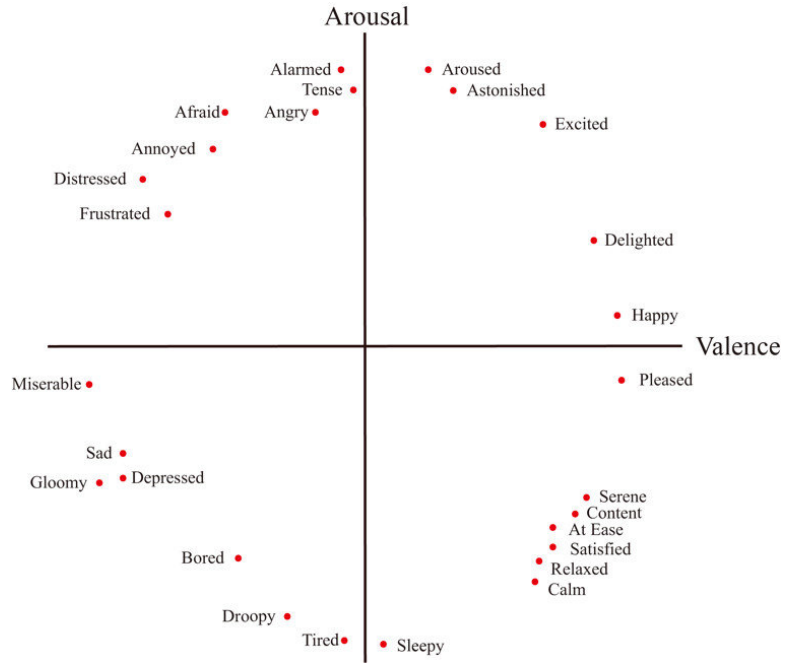


Figura 2.4: Il quadrante delle emozioni proposto da Russell. [1]

2.3 Sistemi di Raccomandazione

Nell'ambito dell'intelligenza artificiale, con sistema di raccomandazione - o *recommender system* - si intende una classe di algoritmi di machine learning usati dagli sviluppatori software per effettuare predizioni sulle scelte dell'utente ed offrire ad esso consigli rilevanti [26]. Esistono principalmente quattro tipologie di recommender system: *collaborative filtering*, *content based filtering*, *demographic filtering* ed i sistemi *ibridi* [27]. Il primo di questi approcci ricerca similarità e relazioni tra gli utenti che usano il sistema. In particolare, questi modelli consigliano ad un utente α i contenuti che sono stati apprezzati da un utente β simile ad α stesso. Viceversa, i modelli appartenenti alla seconda categoria effettuano delle raccomandazioni esclusivamente analizzando le feature e le descrizioni degli oggetti che sono stati valutati positivamente dagli utenti, al fine di consigliagliene altri simili. I modelli demografici invece sfruttano il principio per cui individui con attributi simili (ad esempio sesso, età, paese di provenienza ecc.) avranno preferenze comuni. Infine, l'approccio ibrido prevede di combinare le tecniche utilizzate dagli altri approcci per ottenere risultati superiori.

Un ulteriore divisione nella tassonomia dei recommender system è data dal metodo con cui si effettua la raccomandazione. Esistono infatti due principali categorie: *memory-based* e *model-based*[27]. Il primo metodo utilizza lo storico delle valutazioni dell'utente sui prodotti per calcolare la distanza tra due diversi utenti o *items*. Il metodo model-based invece utilizza modelli di machine learning per effettuare predizioni sulle preferenze dell'utente ed effettuare raccomandazioni. I

modelli più utilizzati sono matrix factorization, classificatori Bayesiani, reti neurali, algoritmi genetici e logica fuzzy. Si utilizzano inoltre tecniche di dimensionality reduction, al fine di ridurre la sparsità dei dati.

Un problema noto dei sistemi di raccomandazione è chiamato *cold start* [27]. Tale evento si può verificare in vari modi e porta ad avere difficoltà nella raccomandazione. Un primo caso si verifica quando non si hanno abbastanza informazioni sull'utente, appena iscritto alla piattaforma utilizzante il recommender system, per effettuare raccomandazioni accurate. In questo caso si parla di *new user problem*. Un secondo caso è quando un nuovo oggetto viene aggiunto al sistema: esso non avrà abbastanza recensioni per essere adeguatamente raccomandato al giusto cluster di utenti, prendendo il nome di *new item problem*. La soluzione proposta in questa tesi è stata ideata con il fine di risolvere tali problemi, come si potrà riscontrare nei capitoli successivi.

2.3.1 Sistemi di Raccomandazione in Ambito Musicale

Il lavoro su cui si basa questa tesi riguarda un dominio specifico, ovvero quello musicale. È importante sottolineare questo aspetto poiché esistono sostanziali differenze rispetto ad altri sistemi di raccomandazione in differenti ambiti. Batmaz *et al.* [2] hanno effettuato uno studio comparativo dei diversi sistemi di raccomandazione, in dominio musicale, presenti in letteratura. Tale studio ha evidenziato le tipologie differenti di dati utilizzati e le ha classificate in tre diverse categorie:

- **Segnali Audio**: si utilizzano feature estratte dai segnali audio per addestrare il modello su cui si basa il sistema di raccomandazione;
- **Informazione Content-Based**: si utilizzano le informazioni relative agli utenti e/o items, utilizzando dataset eterogenei;
- **Valutazioni ed Utilizzo**: si utilizzano le informazioni relative all'ascolto musicale e le valutazioni per selezionare i brani da raccomandare.

In figura 2.5 è possibile osservare uno schema rappresentante la scelta delle feature utilizzate in 9 studi, presenti in letteratura, di recommender system in ambito musicale. Si osserva come quasi la totalità degli studi utilizzano le valutazioni e gli ascolti per effettuare la raccomandazione e solo in alcuni casi si associano questi dati ai segnali audio o alle informazioni content-based. Un limite di questi studi risulta essere quello di non considerare le emozioni nell'effettuare raccomandazioni agli utenti, ignorando i relativi vantaggi.

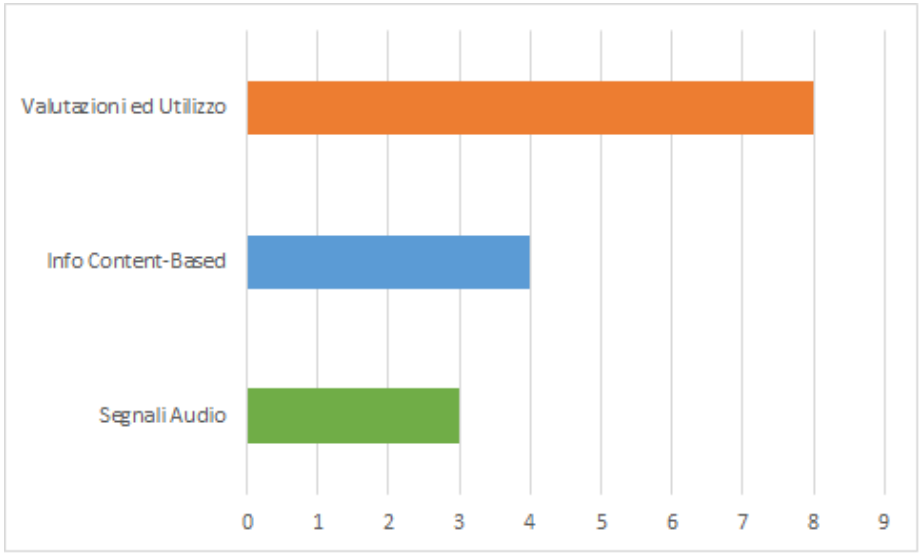


Figura 2.5: Distribuzione tipologie di dati utilizzati come feature su 9 studi riguardanti RS in ambito musicale.[2]

Capitolo 3

Stato dell'Arte

In questo capitolo vengono illustrate diverse tipologie di modelli presenti in letteratura nell'ambito di sistemi di riconoscimento delle emozioni. Al fine di utilizzare le emozioni in un sistema di raccomandazione musicale è necessario creare un modello capace di riconoscere i possibili stimoli emozionali suscitati nell'ascolto. I sistemi capaci di riconoscere emozioni in ambito musicale vengono definiti di *Music Emotion Recognition* (MER). I sistemi MER sfruttano tecniche per elaborare diversi tipi di informazioni, al fine di riconoscere quale tipo di emozione sarà associata ad una canzone.

Kim *et al.* [28] hanno effettuato uno studio di diverse tipologie di approcci utilizzati nell'ambito dei sistemi MER, suddividendoli in due macro categorie: *contextual text information* e *content-based audio analysis*.

3.1 Contextual Text Information

In questi sistemi si utilizzano informazioni testuali per effettuare la predizione dell'emozione suscitata nell'utente. Le principali tecniche utilizzate in questo approccio sono tecniche di *Information Retrieval* (IR), *Text Mining* e *Natural Language Processing* (NLP). Esistono diverse tipologie di sistemi che utilizzano tali informazioni:

- **Riconoscimento emozionale dal testo**: questa tecnica utilizza il testo della canzone - o *lyrics* - per riconoscere l'emozione che essa vuole trasmettere. Malheiro *et al.* [29] hanno effettuato uno studio comparativo di questa metodologia rispetto alla controparte content-based. Per l'estrapolazione delle feature dal testo la metodologia risultata più promettente è stata la *Bag-Of-Word* (BOW) [30]. Tale tecnica, applicata al NLP, consente di creare una rappresentazione vettoriale del testo, composta dal numero di occorrenze delle parole, ed estrarre feature da esso. Il vantaggio di questa tecnica è quello di poter applicare modelli di machine learning sulle feature estratte. Tuttavia, i risultati ottenuti, applicando un classificatore *Support Vector Machine* (SVM), hanno dimostrato sperimentalmente che l'utilizzo di solo

feature testuali (F-score 44.2%) peggiori le performance rispetto all'utilizzo di feature audio (F-score 62.4%).

- **Documenti Web**: questa metodologia analizza i dati documentali ottenibili da svariati sistemi di IR musicali, ad esempio informazioni riguardanti gli artisti, gli album e le recensioni. Uno dei principali problemi di questo approccio è dato dalla scarsa rilevanza delle svariate informazioni che possono essere recuperate da tali sistemi. Le informazioni sono ottenute da pagine web o forum, ossia fonti spesso non autorevoli. È quindi estremamente difficile classificare l'umore associato ad una canzone utilizzando tali feature, a meno di un sistema di filtering capace di selezionare in maniera efficace le informazioni ricercate, rendendo la maggior parte dei dati ottenuti inutili [31].

3.2 Content-Based Audio Analysis

I sistemi precedentemente descritti spesso risultano incompleti a causa della possibile mancanza di informazioni associate ad una canzone o all'inconsistenza dei dati. Al fine di sopperire a tali problematiche nascono i sistemi content-based che, analizzando l'audio esattamente come farebbe un essere umano ascoltando una canzone, riescono ad effettuare raccomandazioni accurate. Tali sistemi sfruttano le feature selezionate per effettuare la classificazione delle canzoni. Le tecnologie impiegate principalmente in questi sistemi sono Machine e Deep learning.

Un primo esempio è fornito da uno studio di Han *et al.* [32], in cui è stato sviluppato un classificatore multi-classe basato su Support Vector Machine (SVM) utilizzante solo feature audio a basso livello come: energia, Zero Crossing Rate (ZCR), frequenza e wavelet. Le canzoni sono state etichettate su un insieme di 11 classi, corrispondenti ad emozioni categoriche. Lo studio, tuttavia, risulta limitato ad un dataset di sole 120 canzoni. I risultati ottenuti mostrano un'accuratezza media di 67.54%.

Nel 2018 Bahuleyan [33] ha mostrato come sia possibile utilizzare reti CNN per classificare un'immagine generata da un file audio. Le immagini create rappresentavano dei spettrogrammi di Mel, ovvero uno spettrogramma dove la frequenza, rappresentata sull'asse delle ordinate, è stata convertita in una scala di Mel mentre sull'asse delle ascisse è rappresentato il tempo [34]. Infine, per considerare l'ampiezza si utilizza una scala colorata, come terza dimensione. Lo studio si concentra sul classificare correttamente il genere musicale. Il modello utilizzato per la classificazione è la rete VGG-16 [35], una delle reti che ha raggiunto le migliori performance nella challenge ImageNet. La ricerca è stata effettuata su un vasto numero di file audio (40540) suddivisi in 7 generi musicali. Per addestrare la rete sono stati usati i due approcci del transfer learning: fine-tuning e feature extractor. La modalità che ha ottenuto l'accuratezza migliore è stata quella del modello addestrato tramite fine-tuning, visibile in figura 3.1, con un'accuratezza

media di 64%. Sebbene questo studio non sia propriamente di ambito MER, ma di classificazione di genere, risulta comunque importate da citare grazie alla dimostrazione che sia possibile utilizzare reti CNN in ambito musicale, sfruttando gli spettrogrammi di Mel, ottenendo ottimi risultati.

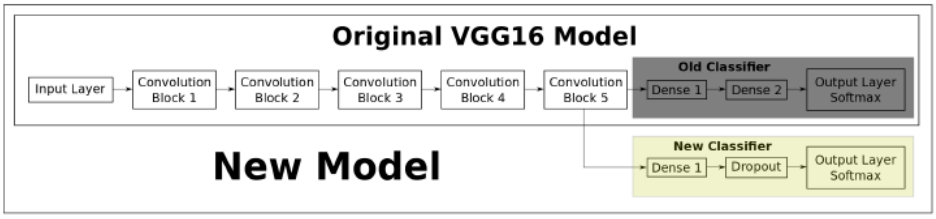


Figura 3.1: Architettura della rete CNN, ottenuta tramite processo di fine-tuning[1].

Nel 2020, Hizlisoy *et al.*[36] hanno studiato come classificare le emozioni indotte dall'ascolto di canzoni tradizionale turche. La rete utilizzata nel processo si basa sulla fusione di due modelli: una CNN, per la ricerca delle feature rilevanti da spettrogrammi di Mel, e una rete LSTM, unita ad una rete neurale profonda, per effettuare la classificazione. Il dataset di musica tradizionale Turca utilizzato è composto da soli 124 frammenti di canzone, dalla durata di 30 secondi l'uno. Le annotazioni sono state effettuate su 3 classi di emozioni, basate su arousal e valence. Le performance ottenute da questo studio sono sorprendenti, raggiungendo un accuratezza massima di 99.19%. D'altro canto, tale studio rimane limitato ad un piccolo dataset e la musica utilizzata ha il bias di essere selezionata in un dominio molto specifico, peccando quindi in generalizzazione. Tuttavia, lo studio rimane molto interessante, dimostrando come l'unione di diversi modelli di deep learning applicati ad un task di MER possano raggiungere valori di accuratezza molto elevati.

[1]Source: `https://github.com/Hvass-Labs/TensorFlow-Tutorials/blob/master/10_Fine-Tuning.ipynb`

3.3 Approccio Ibrido

Questo approccio, tra i più studiati negli ultimi anni, consente di utilizzare multipli modelli di machine learning, unendo diverse metodologie. Ad esempio, implementando analisi content-based audio e NLP, con l'obiettivo di creare modelli che fanno uso di molteplici tecniche congiunte per riconoscere l'emozione associata ad una canzone. Un primo studio è fornito da Liu *et al.* [3] con lo sviluppo di un sistema composto da due differenti approcci. Il primo, basato su LSTM, per la parte della classificazione delle feature relative all'audio. Il secondo è ottenuto integrando un sistema chiamato *BERT* (Bidirectional Encoder Representations from Transformers) [37] per la classificazione del testo. BERT è un sistema basato su transformer, sviluppato da *Google AI* nel 2018, capace di raggiungere risultati sorprendenti nei task di NLP. Tale studio dimostra come l'utilizzo di diverse metodologie congiunte consenta di analizzare con maggior precisione l'umore associato alla canzone, cogliendo cosa l'autore volesse comunicare attraverso sia la musica che le parole. È possibile osservare il modello in questione in figura 3.2. Questa ricerca è stata effettuata utilizzano 1000 canzoni e raggiunge un risultato medio di accuratezza di 79.70%.

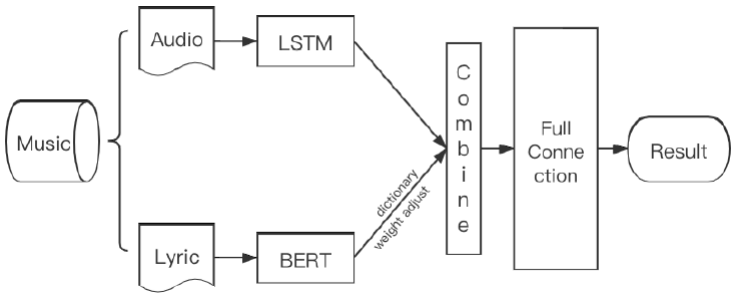


Figura 3.2: Rappresentazione visiva del modello ibrido utilizzato da Liu *et al.* [3].

Capitolo 4

Dataset

Al fine di ottenere una generalizzazione migliore e non avere problemi di overfitting, causati da un bias dovuto a canzoni troppo simili all'interno di un singolo dataset, sono stati presi in esami 3 diversi dataset utilizzati in ambito MER. Questi dataset sono stati studiati e combinati in modo tale da evitare di avere classi eccessivamente sbilanciate. La scelta delle fonti è stata effettuata utilizzando 3 vincoli sulle proprietà che i dataset dovevano avere:

1. I file audio associati alle canzoni dovevano essere disponibili: per poter sfruttare un modello di deep learning capace di analizzare e comprende le emozioni associate ad una canzone era necessario avere a disposizione i file audio da elaborare. Il formato scelto è stato *mp3* essendo quello più comune in questo ambito.

2. Ogni file audio doveva avere una durata di almeno 30 secondi: questo vincolo è stato imposto a causa della necessità di avere una lunghezza fissa per trasformare i file audio in spettrogrammi di Mel, fissata a 30 secondi come descritto nel capitolo successivo.

3. Ogni file audio doveva avere un'annotazione riguardante un'emozione: questo aspetto è fondamentale in uno progetto di deep learning; infatti, ogni campione deve essere propriamente etichettato così da poter addestrare il modello a riconoscere tale etichetta.

Una difficoltà riscontrata è stata quella di allineare tutti i vari dataset con un'etichetta basata su uno standard condiviso. Per il progetto è stato scelto di utilizzare 4 etichette, basate sui rispettivi quadranti di Russell. La notazione adottata è visibile in tabella 4.1. Nella fase di pre-processing tutte le etichette emozionali associate alle canzoni, presenti nei dataset di riferimento, sono state convertite nello standard di annotazione adottato.

Tabella 4.1: Notazione utilizzata per etichettare le emozioni associate alle canzoni nei quadranti di Russel

Arousal	**Valence**	**Etichetta**	**Quadrante**
Basso	Negativo	0	A-V-
Basso	Positivo	1	A-V+
Alto	Negativo	2	A+V-
Alto	Positivo	3	A+V+

I tre dataset selezionati ed utilizzati nel progetto sono: *PMEmo*[38], *4Q*[39][4] ed *Emotify*[40][41].

4.1 PMEmo

PMEmo[1] è un dataset disponibile pubblicamente alla comunità di ricerca, sviluppato da Zhang *et al.* [38] nel 2018. Il dataset è nato per supportare la ricerca, in ambito MER ed affective computing, andando ad annotare le emozioni esplicite ed i segnali fisiologici associati a 457 soggetti in ascolto di musica. Il dataset comprende 794 file audio, corrispondenti ai ritornelli delle canzoni in cima alle classifiche dei brani più ascoltati nel 2016. Nel dettaglio:

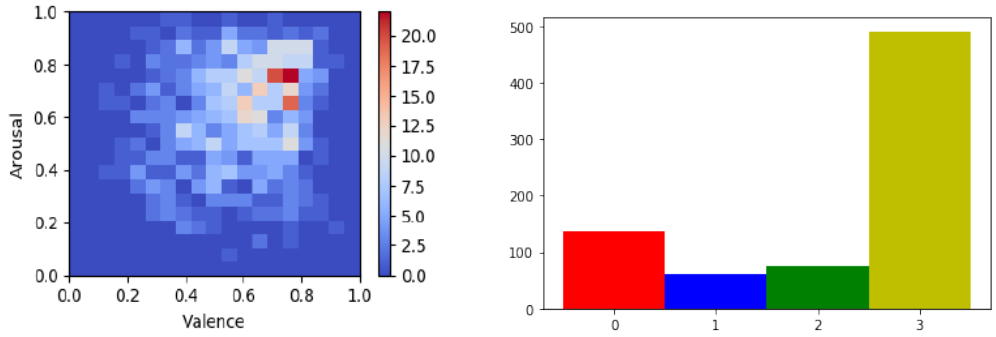
- 487 canzoni da *Billboard Hot 100*;
- 616 canzoni da *iTunes Top 100*;
- 226 canzoni da *UK Top 40 singles*.

I file audio presenti in questo dataset mostrano una lunghezza variabile, è stato dunque necessario convertire ogni file ad una lunghezza fissa di 30 secondi, ovvero la lunghezza fissata per i file audio presenti nel dataset, tra i tre selezionati, con la lunghezza minore. Le annotazioni relative alle emozioni provate dai soggetti in ascolto sono state descritte come un valore compreso tra $[0, 1]$ di arousal e valence. Al fine di utilizzare la notazione standardizzata per tutti i dataset è stata effettuata la seguente conversione:

- $(Arousal \leq 0.5) \wedge (Valence \leq 0.5) \rightarrow Etichetta = 0$
- $(Arousal \leq 0.5) \wedge (Valence > 0.5) \rightarrow Etichetta = 1$
- $(Arousal > 0.5) \wedge (Valence \leq 0.5) \rightarrow Etichetta = 2$
- $(Arousal > 0.5) \wedge (Valence > 0.5) \rightarrow Etichetta = 3$

Il dataset risulta estremamente sbilanciato, dovuto ad un grande numero di canzoni con annotazione relativa a valence ed arousal maggiore di 0.5, com'è possibile osservare in figura 4.1. Per questa ragione, solo un sottoinsieme di 150

[1] https://github.com/HuiZhangDB/PMEmo



(a) Distribuzione delle annotazioni nel dataset PMEmo [38].

(b) Distribuzione delle etichette dopo la conversione.

Figura 4.1: Distribuzioni delle annotazioni prima e dopo la conversione.

file audio, scelti facendo random subsampling dei campioni, è stato utilizzato nell'addestramento del modello finale, con l'obiettivo di evitare un training set troppo sbilanciato.

4.2 4Q

4Q[2] è un dataset creato da Panda *et al.* nel 2018 [39][4]. Questo dataset contiene 900 file audio musicali suddivise ed organizzate in quattro diverse cartelle, ciascuna contenente 225 file e corrispondente ad un quadrante di Russell. Il dataset contiene inoltre una serie di metadati associati alle canzoni, ad esempio l'artista e il genere di riferimento, tuttavia queste informazioni aggiuntive non sono state tenute in considerazione per il progetto. Le clip musicali sono state ottenute attraverso diverse query all'API *AllMusic*[3] e successivamente sono state filtrate ottenendo un totale di 39983 canzoni. Da questo insieme è stato effettuato un bilanciamento, ottenendo un sotto-insieme di 900 canzoni candidate. Il dataset risultante è estremamente bilanciato e dunque tutte le occorrenze sono state utilizzate ai fini del progetto. Le clip, della durata di 30 secondi, non hanno dovuto subire modifiche e sono state analizzate nella loro interezza.

4.3 Emotify

Emotify è un dataset[4] creato da Aljanaki *et al.* nel 2014[40][41]. Questo dataset contiene un totale di 400 canzoni, divise per 4 diversi generi: musica classica, elettronica, pop e rock. La musica è stata ottenuta dalla compagnia *Magnatune recording*[5]. Le annotazioni sono state fornite da diversi utenti, tramite un gioco

[2]http://mir.dei.uc.pt/downloads.html
[3]http://developer.rovicorp.com
[4]http://www2.projects.science.uu.nl/memotion/emotifydata/
[5]http://magnatune.com/

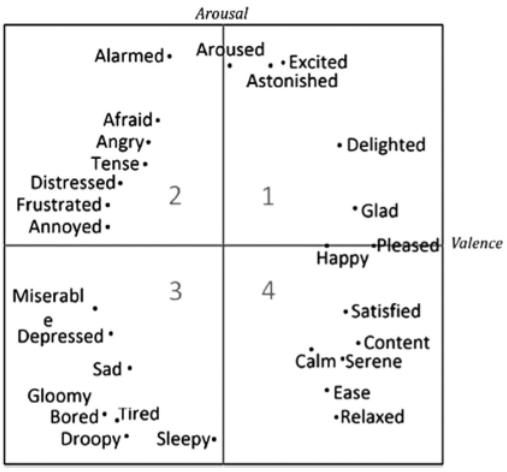


Figura 4.2: Russell circumplex contenente emozioni categoriche usato per la conversione[4].

nel quale ogni soggetto poteva ascoltare la traccia audio ed esprimere diverse preferenze, come il gradimento e la scelta di massimo tre emozioni, che descrivessero ciò che ha provato all'ascolto. A differenza dei dataset descritti precedentemente, *Emotify* utilizza delle annotazioni di tipo categorico basate sulla scala GEMS (Geneva Emotional Music Scale) [42]. Per ciascuna canzone è stata assegnata un'annotazione contente il numero di voti totali per ogni emozione, attribuiti dai partecipanti. È stato quindi necessario, al fine del progetto, scegliere una singola emozione da associare ad ogni canzoni; pertanto, si è deciso di utilizzare la categoria emozionale con il numero di voti più alto. La conversione tra emozioni categoriche nei quattro quadranti di Russell è stata effettuata seguendo lo schema in figura 4.2.

Una volta effettuata la conversione è stato riscontrato che il dataset risultava estremamente sbilanciato, con la maggior parte delle canzoni etichettate come basso arousal e valence positiva: ciò è dovuto probabilmente al numero di aggettivi categorici proposti nel gioco, i quali propendevano per questo specifico quadrante. È possibile osservare lo sbilanciamento citato in figura 4.3. Dunque, anche in questo caso, al fine di evitare un eccessivo sbilanciamento del training set, sono stati utilizzati solamente 150 file audio, selezionati in maniera causale.

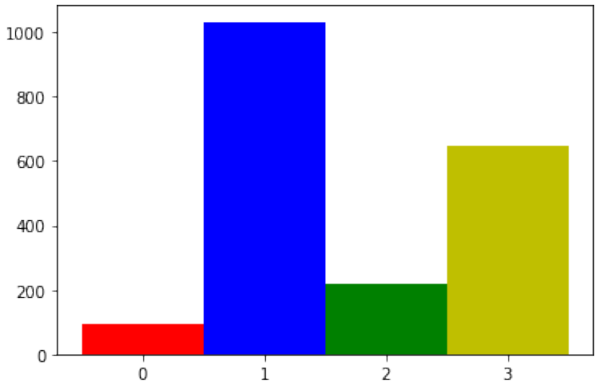


Figura 4.3: Distribuzione delle etichette dopo la conversione da categorico ai quadranti di Russell.

4.4 Preprocessing

Una volta selezionati i dataset da utilizzare è stata effettuata la fase di preprocessing dei dati. In primo luogo, si è standardizzato il sistema di etichettatura utilizzato, come descritto nelle sezioni precedenti, utilizzando metodi di conversioni ad hoc per ogni notazione dei diversi dataset. Una volta che le etichette sono state standardizzate, è stato possibile unire i dataset creando un nuovo insieme contenente 1200 file audio dalla durata di 30 secondi ciascuno. Il modello ideato utilizza una rete CNN per analizzare visivamente le varie canzoni; pertanto, è stato necessario generare per ogni file audio diversi spettrogrammi di Mel. La scelta di utilizzare gli spettrogrammi di Mel è stata fatta sulla base della loro capacità di includere informazioni para-linguali, rendendoli particolarmente utili per task di MER [43], come dimostrato dallo studio di Bahuleyan [33] e Hizlisoy *et al.* [36], descritti nel capitolo 3 relativo allo stato dell'arte. Ogni file audio è stato diviso in 5 frame da 6 secondi, ottenendo 6000 frame, su cui è stato generato lo spettrogramma. Il preprocessing dei file audio e la generazione degli spettrogrammi sono stati implementati utilizzando la libreria Python *Librosa* [44]. E' possibile vedere Un esempio di spettrogrammi di Mel, generati utilizzando la libreria *Librosa*, in figura 4.4.

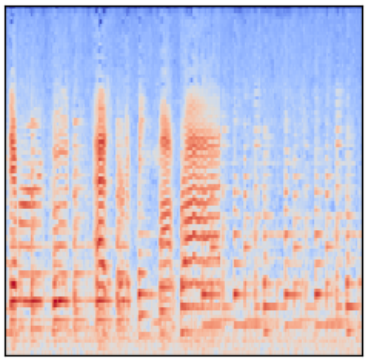
I parametri selezionati per la generazione degli spettrogrammi di Mel sono:

- **Frequenza di campionamento:** 22050;
- **Dimensione del frame:** 2048;
- **Hop Size:** 512;
- **Numero di Mel bins:** 96;
- **Dimensione dell'immagine:** 300 x 300 pixel.

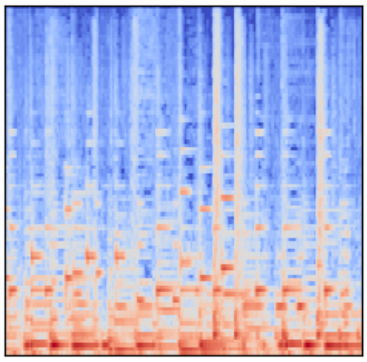
Infine, il dataset ottenuto è stato diviso in tre sottoinsiemi per effettuare l'addestramento delle reti neurali: training set, test set e validation set. Rispettivamente questi insiemi sono stati ottenuti seguendo la seguente proporzione:

- 70% training set: 4200 frame;
- 15% test set: 900 frame;
- 15% validation set: 900 frame.

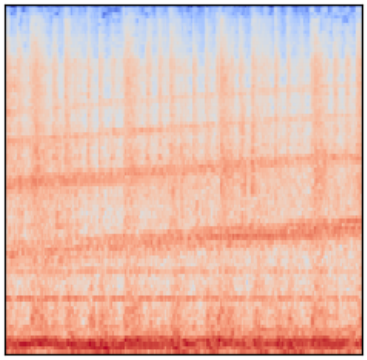
La divisione è stata effettuata utilizzando gli ID dei file audio, prima della divisione in frame, così da evitare di avere frame relativi ad una stessa canzone in due insiemi diversi.



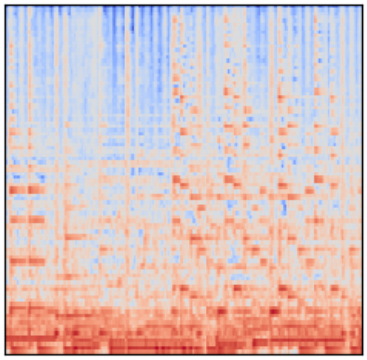
(a) Basso Arousal - Valence Negativo



(b) Basso Arousal - Valence Positivo



(c) Alto Arousal - Valence Negativo



(d) Alto Arousal - Valence Positivo

Figura 4.4: Esempio di 4 spettrogrammi di Mel generati da file audio etichettati con differenti classi emozionali.

Capitolo 5

Descrizione dei Modelli MER Proposti

L'obiettivo del lavoro è quello di creare un sistema di raccomandazione basato su modelli di Deep Learning che sfruttano le emozioni per creare la giusta playlist, in base allo stato emotivo dell'utente. Il modello proposto consiste in due reti fondamentali, capaci di ottenere ottime performance in task di riconoscimento emozionale in ambito musicale. Le reti implementate sono:

- Convolutional Neural Network (CNN);
- Long Short-Term Memory (LSTM).

5.1 Convolutional Neural Network

La prima rete impiegata nel modello di riconoscimento emozionale è una 3D CNN. Tale rete è capace di riconoscere i pattern caratteristici e le feature rilevanti degli spettrogrammi di Mel, rispetto alle classi emozionali di appartenenza. La rete è definita 3D perché, oltre a basarsi sulle dimensioni di frequenza e tempo, riesce ad osservare il colore dello spettrogramma e quindi a tenere conto anche dell'ampiezza. Com'è stato possibile notare in figura 4.4, gli spettrogrammi differiscono visivamente quando il mood associato alla canzone è diverso; dunque, la rete impara a riconoscere tali differenze. Al fine di ottenere performance soddisfacenti sono state analizzate due reti neurali convoluzionali pre-addestrate, con performance allo stato dell'arte nel task *ImageNet.* Le reti analizzate presentano sostanziali differenze architetturali, così da scegliere empiricamente quella più adatta ai fini del progetto. Le reti prese in esame sono: *EfficientNet-B3* [45] e *MobileNetV3-Large* [7].

5.1.1 EfficientNet-B3

Le EfficientNet sono una famiglia di reti neurali convoluzionali, presentate per la prima volta nel maggio 2019 da Tan *et al.* [45], basate su un metodo di model scaling innovativo rispetto ad altre reti allo stato dell'arte. Il vantaggio di questa architettura è la capacità di scalare uniformemente le dimensioni di larghezza, profondità e risoluzione utilizzando un coefficiente composto ϕ. Nello specifico:

$$\begin{aligned} profondità : d &= \alpha^{\phi} \\ larghezza : w &= \beta^{\phi} \\ risoluzione : r &= \gamma^{\phi} \\ s.t.\ \alpha \cdot \beta^2 \cdot \gamma^2 &\approx 2 \\ \alpha \geq 1, \beta &\geq 1, \gamma \geq 1 \end{aligned} \tag{5.1}$$

dove α, β, γ sono costanti determinabili da una grid search.

La versione B7 di EfficientNet ha raggiunto un'accuratezza top-1, su *ImageNet*, pari al 84.3%, raggiungendo lo stato dell'arte a parità di *Gpipe* [46], pur essendo 8.4 volte più piccola e 6.1 volte più veloce nell'inferenza. EfficientNet si è inoltre dimostrata molto performante in caso di utilizzo in transfer learning, in particolare raggiungendo un'accuratezza pari al 91.7% su Cifar-100 [47] e al 98.8% su Flowers [48].

Nel caso dell'obiettivo di riconoscimento emozionale trattato in questo progetto è stata scelta la versione B3, la quale prende in input immagini di dimensione 300x300 pixel. Questa decisione è dettata dal trade-off tra compattezza, quindi limiti hardware, e performance della rete. È possibile osservare l'architettura della rete in figura 5.1. Le performance delle varie versioni di EfficientNet su *ImageNet* rispetto ad altre reti sono mostrate in figura 5.2.

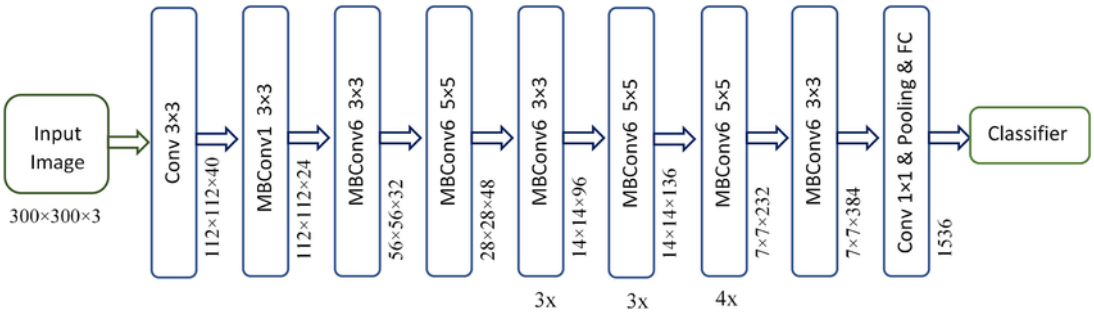


Figura 5.1: Architettura compatta della rete EfficientNet-B3 [5].

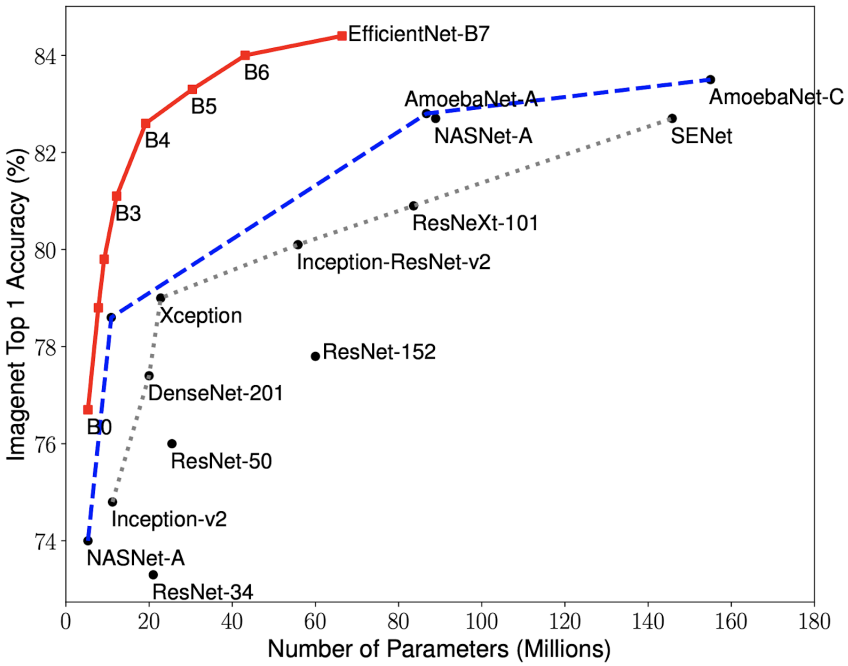


Figura 5.2: Performance di EfficientNet rispetto ad altre reti nella challenge *ImageNet*.

5.1.2 MobileNetV3

La rete MobileNet nasce nel aprile 2017, dallo studio di Howard *et al.* [49] nel creare una rete neurale convoluzionale "leggera", con pochi parametri e basso costo computazionale, così da poter essere utilizzata in applicazioni mobile ed applicazioni embedded di computer vision. L'architettura utilizza quella che viene chiamata *Depthwise Separable Convolution* (DSC), un metodo che consente di ridurre drasticamente il numero di parametri rispetto ad altre reti con la stessa profondità [50]. Il DSC è formato da due operazioni fondamentali, fattorizzando un layer convoluzionale standard in due layer:

- **Depthwise Convolution**: per ogni canale si effettua una convoluzione spaziale in maniera separata;
- **Pointwise Convolution**: si effettua in coda una convoluzione 1x1 per cambiare la dimensione.

L'idea è quella di separare le dimensioni spaziali e la profondità dei filtri, ottenendo reti più profonde ma molto più leggere. È possibile osservare visivamente il DSC in figura 5.3. I vantaggi principali sono bassa latenza e basso costo computazionale, quindi bassa energia richiesta per l'utilizzo.

Nel 2019 Howard *et al.* [7] rilasciano una versione migliorata dell'originale MobileNet, pubblicando uno studio sulla versione di nuova generazione chiamata MobileNetV3. Tale versione ottiene un miglioramento di circa il 25% nella velocità

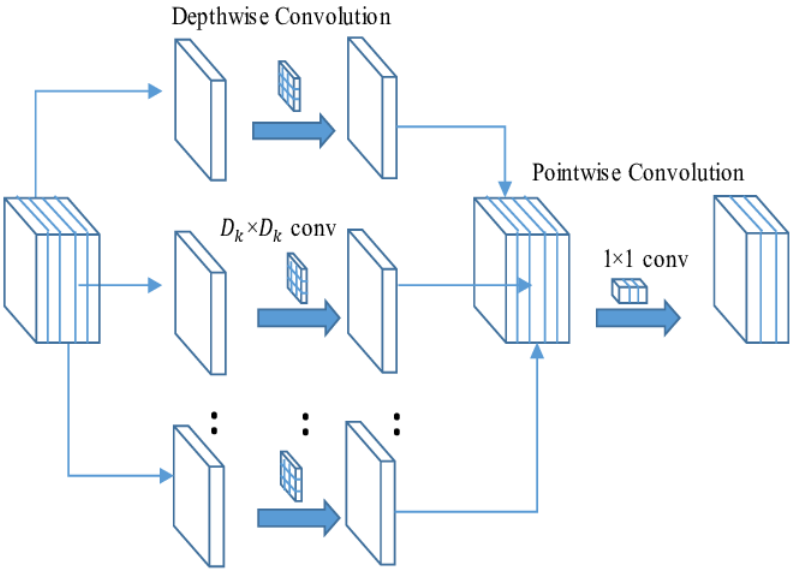


Figura 5.3: Depthwise Separable Convolution (DSC) [6].

di inferenza rispetto alla versione V2, mantenendo la stessa accuratezza. Per ottenere questo avanzamento è stato inserito un layer chiamato "squeeze and excitation". Questo strato, posizionato dietro il global average pooling layer, lavora con feature map molto piccole (1x1) a differenza delle reti precedenti (7x7), alleggerendo la rete [8]. È possibile osservare la comparazione tra la rete originale e la versione V3 in figura 5.4. Nel progetto è stata implementata la versione di MobileNetV3 "Large", ovvero quella con più parametri che raggiunge i valori di accuratezza migliore.

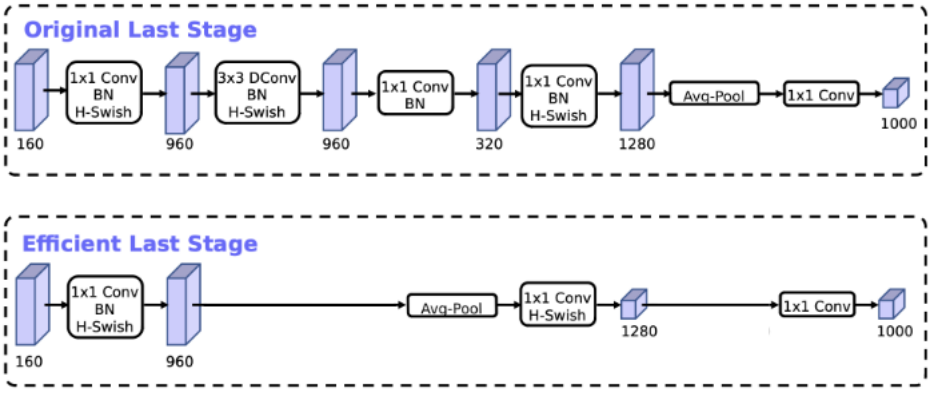


Figura 5.4: Comparazione tra rete originale e la versione più efficiente V3 [7][8].

5.1.3 Addestramento

Nella fase di training dei modelli sono stati utilizzati in partenza i pesi ottenuti dalle reti su *ImageNet*, al fine di effettuare un transfer learning sul nuovo task di MER. È stato scelto di utilizzare questa configurazione grazie alla grande varietà di feature a basso livello riconosciute dalle reti, avendo quindi una buona base di partenza per riconoscere le varie differenze tra gli spettrogrammi. Per effettuare la classificazione sono stati aggiunti all'ultimo layer di output delle CNN i seguenti layer:

- Flatten layer: ottiene le feature dall'ultimo layer delle reti CNN e ne cambia forma, togliendo le varie dimensioni agli array e restituendo un vettore ad una dimensione. Questo array monodimensionale sarà l'input della rete fully-connected usata come classificatore;
- Dense layer di 1024 neuroni: classico strato di una rete fully-connected dove ogni neurone è collegato in input ed output a tutti quelli precedenti e successivi;
- Dropout Layer con dropout rate di 0.5: strato usato per la regolarizzazione della rete. Ogni iterazione vengono disattivati il 50% dei neuroni, scelti in maniera causale, così da evitare che l'informazione passi sempre per neuroni troppo specializzati, creando il fenomeno di overfitting. Lo spegnere in maniera random diversi neuroni fa si che l'informazione trovi di volta in volta una via diversa per confluire nella rete, distribuendo meglio i pesi;
- Dense layer di 512 neuroni;
- Dropout layer con dropout rate di 0.4: andando avanti negli strati è bene diminuire la percentuale di dropout dato che i neuroni diventano sempre di numero minore;
- Dense layer di 128 neuroni;
- Dropout layer con dropout rate di 0.3;
- Dense layer di 4 neuroni con funzione di attivazione softmax: layer di output della rete, ogni neurone rappresenta una classe. La funzione softmax ottiene le stime percentuali di appartenenza alle classi e ne sceglie quella maggiore.

In tutti i dense layer è stata usata la funzione di attivazione ReLU (Rectified Linear Unit) [51], la funzione più utilizzata nelle reti neurali profonde, grazie alla sua capacità di non saturare il gradiente. Inoltre, ogni dense layer utilizza anche una regolarizzazione L2 con $\alpha = 0.1$. La regolarizzazione L2, chiamata anche *ridge regression*, viene utilizzata per penalizzare i pesi troppo grossi, evitando quindi fenomeni di overfitting [52]. La funzione obiettivo va a minimizzare la *sparse categorical crossentropy loss*, funzione di errore utilizzata nelle classificazioni multi-classe [53]. La funzione di ottimizzazione adottata è stata *Adam* [54]

la quale, grazie alle proprietà adattive, riesce a trovare i punti di minimo in maniera estremamente rapida, sfruttando il momentum nella discesa del gradiente. Per stimare il learning rate ottimale è stato utilizzato un algoritmo di ricerca: si effettua un training basato su 5 epoche e si aumenta ad ogni iterazione il learning rate in maniera esponenziale, in un range da 0.0001 ed 1. Il learning rate ottimale risulta quello che consente alla loss di scendere in maniera più ripida. In figura 5.5 è possibile osservare un esempio del grafico ottenuto dal processo di ricerca descritto. Il training è stato effettuato attraverso 50 epoche, utilizzando un batch size di 32 istanze. Il dataset è stato suddiviso rispettivamente nei sotto-insiemi di training (70%), validation (15%) e testing (15%). Al fine di utilizzare il modello che ha ottenuto performance di generalizzazione migliori, vengono memorizzati solo i pesi che hanno ottenuto i valori di accuratezza maggiori sul validation set.

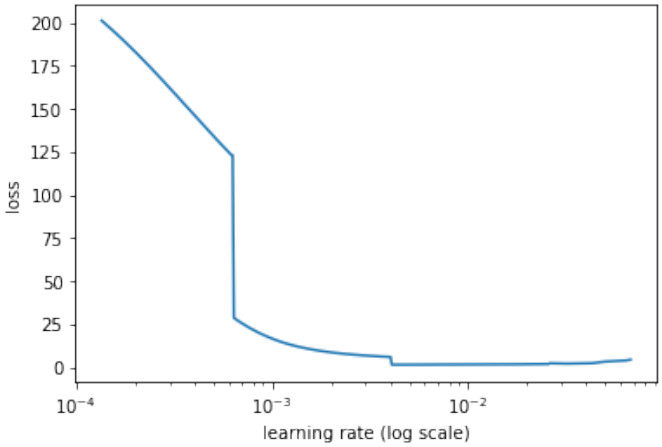


Figura 5.5: Grafico raffigurante la curva di loss ottenuta dall'algoritmo di ricerca del learning rate ottimale. Nel esempio specifico il risultato finale è stata del valore di 0.0006.

La libreria utilizzata per implementare il codice *Python* per l'addestramento delle reti è stata *TensorFlow* [55], una delle librerie principali relative allo sviluppo di modelli di Deep Learning. I notebook contenenti i training dei modelli sono stati eseguiti sulla piattaforma *Kaggle*[1], utilizzando le seguenti specifiche hardware:

- **CPU:** Intel(R) Xeon(R) CPU @ 2.30GHz;
- **RAM:** 12GB;
- **GPU:** NVIDIA TESLA P100 16GB.

In figura 5.6 è possibile osservare le metriche di accuratezza e loss durante la fase di addestramento, relative agli insiemi di training e di validation. Si nota come sia presente overfitting sul training set, pur avendo attuato diversi tipi di regolarizzazioni per evitarlo. Questo è dovuto alla complessità ed al gran numero

[1] `https://www.kaggle.com/`

di parametri delle reti utilizzate. Per far fronte a questo problema si sono utilizzati solo i pesi che hanno ottenuto i migliori punteggi di accuratezza sul validation set, come descritto in precedenza. La metrica di loss invece decresce in maniera netta per entrambi i modelli.

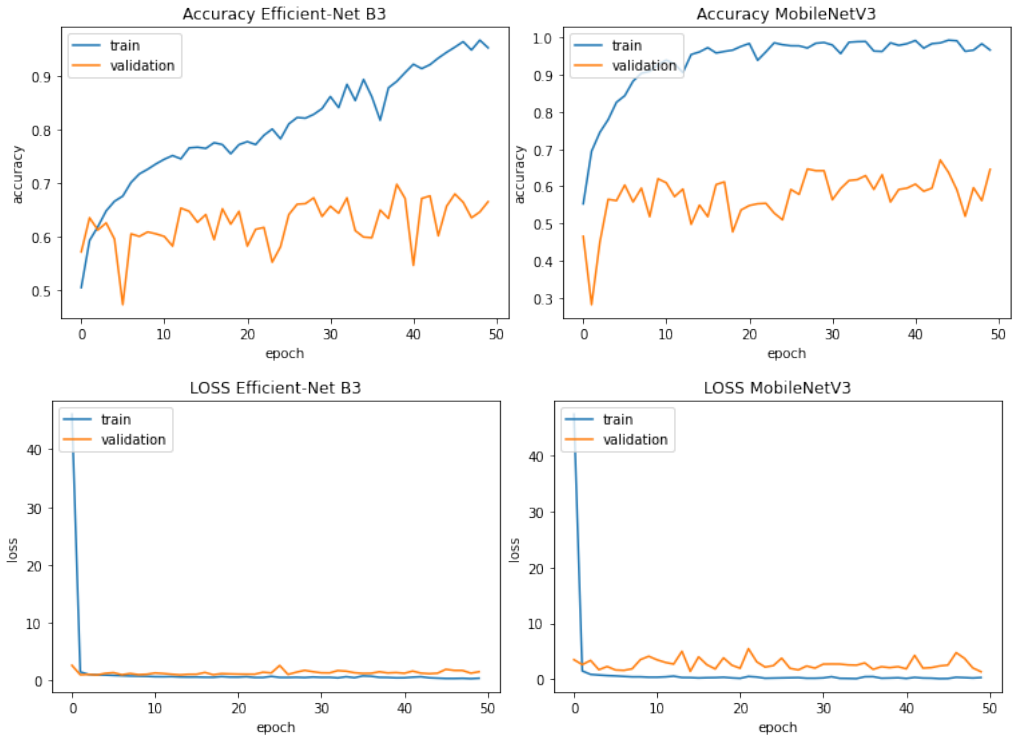


Figura 5.6: Performance delle reti CNN durante il training.

5.1.4 Confronto Risultati

Le performance ottenute durante l'addestramento non sono sufficienti a valutare il modello. È necessario utilizzare un ulteriore insieme di dati, chiamato test set, per osservare come il modello si comporta su dati che non ha mai visto in precedenza. È importante fare questa ulteriore verifica per avere la certezza che la rete abbia delle buone performance nel mondo reale, ovvero che abbia la giusta capacità di generalizzare. Pertanto, la valutazione dei risultati è stata effettuata sul test set. Le performance sono visibili in tabella 5.1.

Tabella 5.1: Performance delle reti CNN.

Modello	Accuratezza	F-score	Tempo Medio di Training per Epoca
EfficientNet-B3	0.69	0.67	90s
MobileNetV3	0.60	0.60	32s

Come mostrato in tabella 5.1, il modello basato sulla rete MobileNet V3 ha un leggero deficit in accuratezza, rispetto la rete EfficientNet-B3. Tuttavia, il modello basato su MobileNet ha un tempo medio di training per ogni epoca di quasi un terzo rispetto la controparte. Idealmente, l'applicazione contenente il sistema di raccomandazione, di cui si effettua lo studio, dovrebbe avere la capacità di essere eseguita su dispositivi mobili, quindi con carenza di capacità hardware. Diventa quindi importante considerare l'aspetto relativo alla leggerezza del modello e pertanto la scelta finale cade sulla rete basata su MobileNet, di cui il trade-off tra velocità e performance è risultato favorevole.

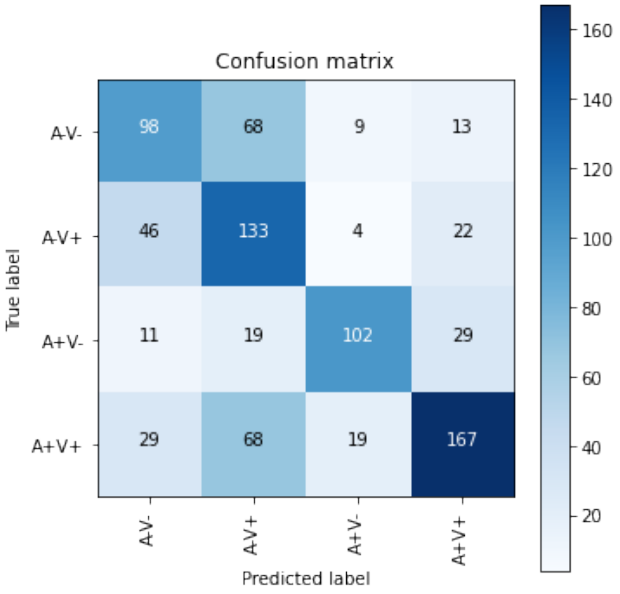


Figura 5.7: Grafico raffigurante matrice di confusione ottenuta sul test set dal modello MobileNetV3.

Un ulteriore dettaglio relativo all'accuratezza è visibile in figura 5.7 in cui viene mostrata la matrice di confusione relativa a MobileNetV3. Si nota come la diagonale risulta marcata e quindi la maggior parte delle predizioni risulta corretta per ogni classe. Sono visibili alcuni falsi positivi rispetto la classe alto arousal e valence positiva, causati da un leggero sbilanciamento dei dati rispetto a quella specifica etichetta. Nel complesso il risultato è accettabile.

5.2 Long Short-Term Memory

La seconda rete impiegata nel modello di riconoscimento emozionale è una LSTM. Una volta che la rete CNN è stata addestrata a riconoscere le feature rilevanti dalle immagini, raffiguranti gli spettrogrammi di Mel, è possibile utilizzarla come feature extractor. La rete LSTM sarà quindi usata come nuovo classificatore del modello finale. Questi modelli sono vastamente utilizzati in task di riconoscimento musicale grazie alla loro capacità di processare informazioni sequenziali [23]. Al fine di analizzare ogni iterazione una porzione di canzone di lunghezza media, ossia non troppo lunga ma nemmeno eccessivamente frammentata, è stato deciso di utilizzare, come input alla rete, frame audio di 24 secondi. Per ottenere questa lunghezza si generano 4 spettrogrammi, relativi a 6 secondi ciascuno. Quest'ultimi vengono utilizzati come input della rete CNN: le feature estratte in output subiranno un reshape per essere considerate un singolo input, di quattro sequenze, per la rete LSTM. È possibile osservare il processo appena descritto raffigurato in maniera schematica in figura 5.8.

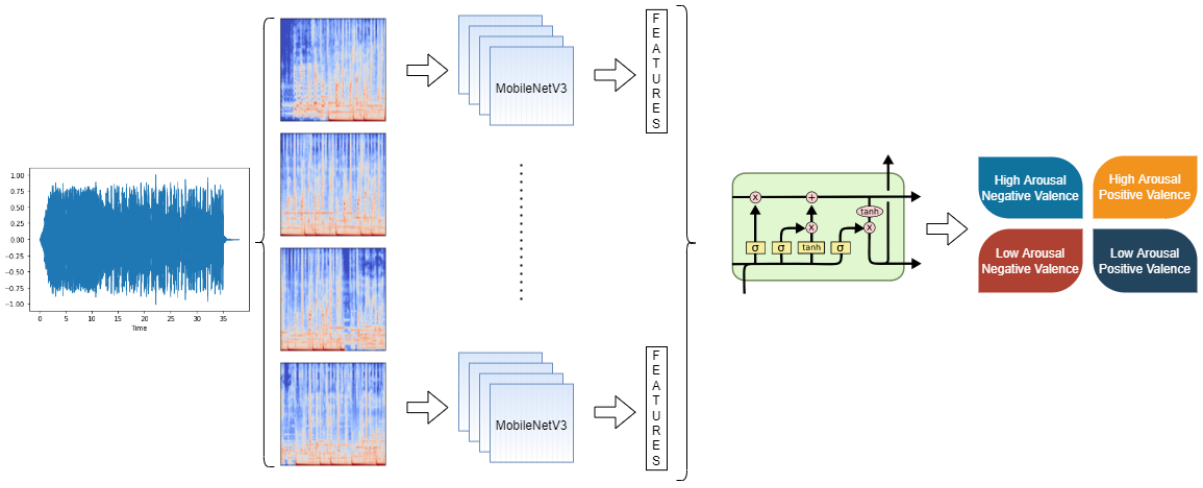


Figura 5.8: Le feature sono estratte usando la rete CNN ed utilizzate come singolo input di 4 sequenze per la LSTM.

5.2.1 Addestramento

L'architettura della rete LSTM è formata da:

- 128 LSTM hidden unit con funzione di attivazione ReLU;
- Dropout layer con dropout rate di 0.2;
- Dense layer di 4 output, con funzione di attivazione softmax.

La funzione obiettivo minimizza la *categorical sparse crossentropy loss*. Viene utilizzata come funzione di ottimizzazione *RMSProp* [56] poiché presente empiricamente migliori performance quando applicata a reti LSTM. L'addestramento è effettuato utilizzando 20 epoche e una batch size di 32 istanze. Il dataset è stato suddiviso rispettivamente nei sotto-insiemi di training (70%), validation (15%) e testing (15%). Anche in questo caso si andranno ad utilizzare come modello finale i pesi che hanno ottenuto il migliore punteggio di accuratezza sul validation set. Per l'esecuzione del notebook è stato sempre utilizzato *Kaggle*, con le specifiche elencate nello scorso capitolo. È possibile osservare in figura 5.9 le performance ottenute nelle varie epoche, utilizzando le feature ottenute dalla rete MobileNetV3.

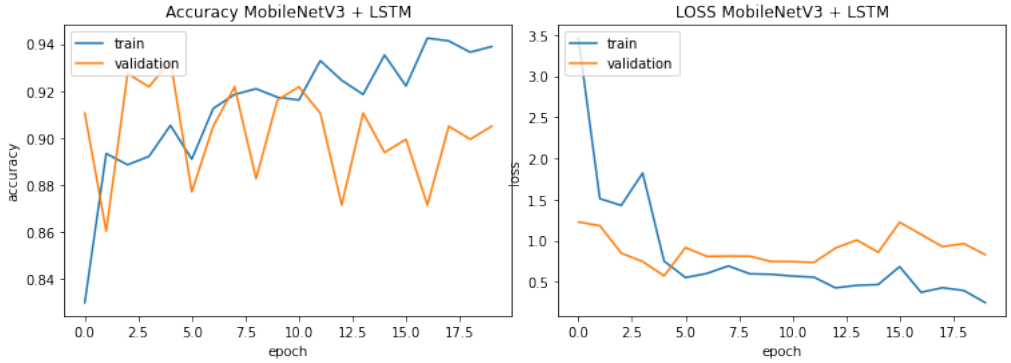


Figura 5.9: Performance della rete LSTM durante il training.

I risultati ottenuti nel training migliorano nettamente i risultati ottenuti tramite l'utilizzo della sola rete CNN. È possibile notare tuttavia un inizio di overfitting, visibile dalla divergenza delle due curve nella crescita di accuratezza. La loss invece decresce e rimane stabile per entrambi i set.

5.2.2 Risultati

La rete LSTM raggiunge ottimi risultati anche in fase di testing, raggiungendo un valore di accuratezza e F-score di 91%. La matrice di confusione, mostrata in figura 5.10, mostra come quasi tutte le classificazioni siano effettuate con successo. I risultati sono in linea con quelli ottenuti dagli studi citati nel capitolo relativo allo stato dell'arte. Il modello dimostra di essere capace di generalizzare in maniera efficace; pertanto, risulta utilizzabili ai fini del sistema di raccomandazione.

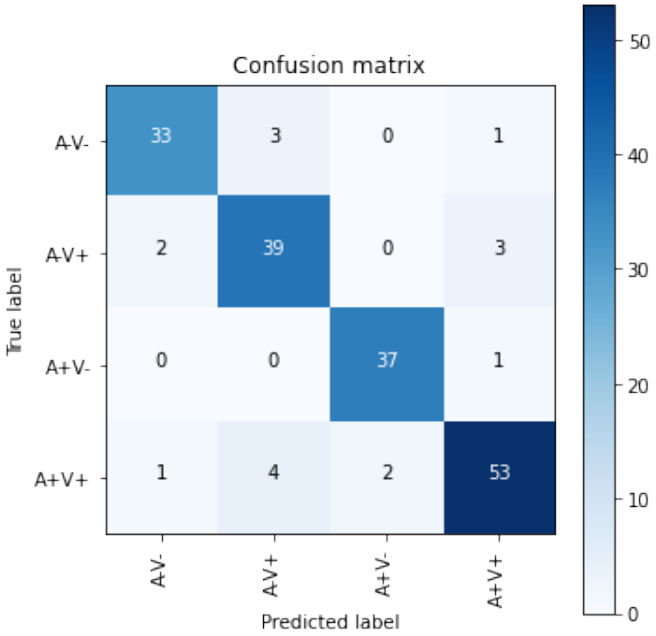


Figura 5.10: Grafico raffigurante matrice di confusione ottenuta sul test set dal modello LSTM, basato sulle feature estratte da MobileNetV3.

5.3 Variazione Intra-Dataset ed Inter-Dataset

Un aspetto importante da valutare, considerando l'utilizzo di fonti multiple, sono le variazioni dei dati tra i diversi dataset impiegati nel processo di training. Esistono, infatti, dipendenze tra i campioni presenti all'interno di un singolo dataset che possono portare a bias comuni. Ad esempio, le canzoni, selezionate da classifiche di brani più ascoltati in un certo periodo storico, sono soggette alle mode musicali del momento, le quali possono portare ad avere esempi musicalmente simili e quindi ad ottenere dei risultati di performance maggiori che quelle raggiungibili in un'applicazione nel mondo reale, composto da esempi più eterogenei. L'utilizzo di dati da diverse fonti, come nel caso delle tre impiegate in questo lavoro, sono già un espediente per risolvere questo problema di dipendenze tra i dati, fornendo un campione più eterogeneo e quindi aumentando la capacità del modello di generalizzare [57]. È tuttavia interessante andare a studiare queste dipendenze, effettuando diverse prove empiriche, attraverso l'utilizzo del modello introdotto nei capitoli precedenti. Nello specifico, sono state effettuate 3 diverse prove, con lo scopo di valutare quanto le performance variassero tra modelli addestrati con solo esempi di un dataset, quindi intra-dataset, e modelli addestrati con campioni presi da diverse fonti, quindi inter-dataset.

5.3.1 Modello Interamente Addestrato su Singolo Dataset

Il primo esperimento è stato addestrare il modello finale, composto dalle reti CNN (MobileNetV3) e LSTM, solo su un dataset ed effettuare il test sui rimanenti. Questa prova valuta le performance della rete, addestrata utilizzando solo esempi relativi ad un dataset. Nello specifico è stato selezionato *4Q*, grazie al suo bilanciamento tra classi. Una volta addestrato il modello si verificano le performance, utilizzando esempi ottenuti dalle restanti fonti. Per il testing si è usato l'intero set di dati relativi ai due dataset non utilizzati nel training. Il dataset *4Q* è stato invece diviso nei sotto-insiemi di training (90%), validation (5%) e testing (5%), anche in questo caso ponendo particolare attenzione nel non utilizzare spettrogrammi relativi alla stessa canzone in due insiemi differenti. Così facendo, le feature riconosciute dalla rete CNN appartengono solo a spettrogrammi di un singolo dataset e il classificatore è addestrato a utilizzare solo tali informazioni per effettuare la predizione. I risultati ottenuti, come prevedibile, mostrano un netto distacco tra le performance del modello testato sul dataset di training e quelle relative alle due altre fonti. È possibile osservare le performance ottenute in tabella 5.2. Tale differenza mostra una certa dipendenza tra le canzoni inter-dataset, tuttavia i valori ottenuti, utilizzando interi dataset come testing, sono accettabili per un problema di classificazione multi-classe e dimostrano che il modello ha una capacità di generalizzazione di circa il 50%, valore confermato empiricamente nel capitolo 6.

Tabella 5.2: Riepilogo del modello addestrato sul dataset *4Q* e testato sui restanti.

	4Q	**PMEmo**	**Emotify**
Accuratezza	0.8	0.47	0.54

5.3.2 CNN Addestrata su Singolo Dataset

Il secondo esperimento va maggiormente nel dettaglio. In questa prova non si addestra l'intero modello su un singolo dataset, ma solo la prima parte della rete, ovvero la CNN. Lo scopo di tale configurazione è quello di andare ad osservare come una rete convoluzionale, addestrata su un solo dataset, riesca ad estrarre feature rilevanti di spettrogrammi generati da altri dataset. Tali informazioni saranno utilizzate per addestrare la rete LSTM. Nello specifico:

- Per l'addestramento della rete CNN si utilizza *4Q* suddiviso in training (90%) e validation (5%);
- Per l'addestramento della rete LSTM verranno utilizzate le feature estratte dall'insieme di training della rete CNN insieme a quelle ottenute dai restanti dataset, sempre suddivisi in training (90%), validation (5%);
- Per la fase di testing si utilizzano le restanti porzioni dei tre dataset (5% ciascuno).

In questo caso i valori di accuratezza ottenuti, sulle porzioni di test, risultano essere migliorati per entrambi i dataset non impiegati nell'addestramento della rete CNN, ma solo di quella LSTM, in relazione alla prova precedente. È possibile osservare tali valori in tabella 5.3. Questo aumento è imputabile al classificatore, capace di interpretare meglio le feature estratte dalla rete CNN, relative ai dataset non impiegati nell'addestramento di quest'ultima. Un punto a favore di quest'interpretazione è la miglior capacità di classificare esempi relativi alle fonti escluse dal training della CNN. Le performance ottenute sul dataset *4Q* risultano invece leggermente peggiorate: questo è spiegabile a causa di una perdita di specializzazione del classificatore sul dataset di riferimento, in funzione di una migliore generalizzazione.

Tabella 5.3: Riepilogo della modello composto dalla rete CNN, addestrata sul dataset *4Q*, e dalla rete LSTM addestrata su tutti.

	4Q	**PMEmo**	**Emotify**
Accuratezza	0.75	0.59	0.65

5.3.3 Modello Addestrato su Insieme Inter-Dataset

Quest'ultimo esperimento è necessario a verificare come il modello, inteso in tutte le sue parti (CNN e LSTM), si comporti se addestrato sull'insieme congiunto di tutti e tre i dataset. A differenza di quanto visto nel capitolo 5.2, in questa fase non si utilizzeranno dei sotto-insiemi bilanciati dei vari dataset, ma l'intero campione dei dati. Nell'ambito del Deep Learning, l'aspetto più fondamentale per le performance di un modello non è solamente l'architettura della rete, ma soprattutto la qualità dei dati utilizzati in fase di training. L'utilizzo dei dataset completi, senza pre-processing per bilanciare le classi, potrebbe portare a performance peggiori. Anche in questo caso si vanno a creare i tre sotto-insiemi di training (90%), validation (5%) e testing (5%) per ogni dataset e successivamente si uniscono quelli di training e validation per creare un unico grande insieme per l'addestramento della rete. I test invece rimarranno disgiunti, così da poter osservare come un modello inter-dataset si comporti in relazione ai singoli dataset di partenza. In tabella 5.4 è possibile osservare i risultati ottenuti.

Tabella 5.4: Riepilogo della modello addestrato tramite l'utilizzo di tutti i dataset, senza bilanciamento delle classi.

	4Q	**PMEmo**	**Emotify**
Accuratezza	0.73	0.77	0.57

L'accuratezza ottenuta dai dataset *4Q* e *PMEmo* risulta essere buona, seppur più bassa di quella ottenuta dal modello finale progettato nel capitolo 5.2. Il dataset *Emotify* invece raggiunge risultati di performance più bassi. Al fine di indagare le ragioni di questa diminuzione di performance generali, sono state generate le matrici di confusione per ogni test set. È possibile osservarle in figura 5.11:

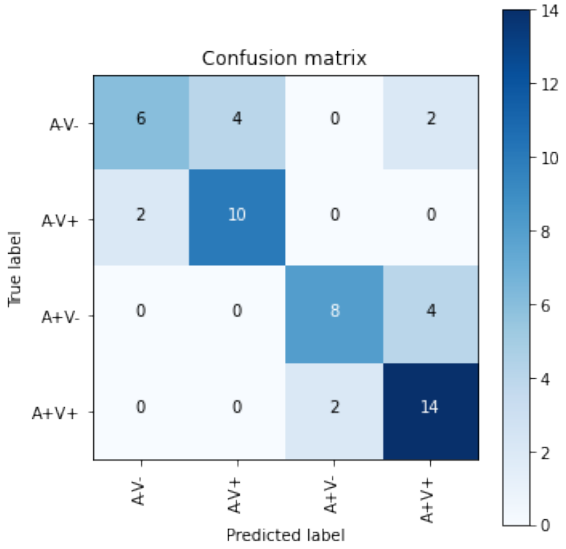
- 5.11a (4Q): la matrice di confusione mostra una diagonale marcata, indice di buon bilanciamento nel test set delle classi e di capacità del modello di generalizzare. Si notano tuttavia alcuni falsi positivi relativi alle classi A-V+ e A+V+, ossia le etichette su cui i restanti dataset presentano uno sbilanciamento. Ciò dimostra come la fusione di dataset, con bilanciamenti interni delle classi molto diversi, senza un'opportuna fase di pre-processing, porti ad un peggioramento generale del modello e delle performance ottenute sui dataset con un giusto rapporto tra le classi.

- 5.11b (PMEmo): questo dataset presenta un estremo sbilanciamento generale dei dati verso la classe A+V+: la maggior parte degli esempi appartengono a questa classe e il modello li riconosce correttamente, a discapito delle restanti etichette. Questo fenomeno è dovuto al partizionamento nei

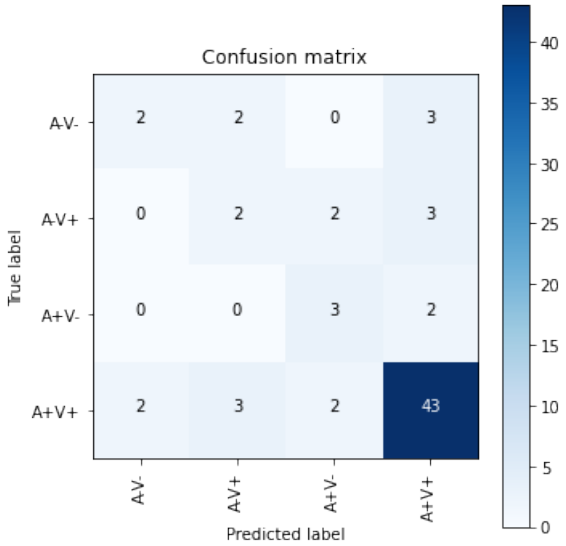
vari insiemi di training e test, i quali mantengono la proporzione sbilanciata del dataset di partenza.

- 5.11c (Emotify): analogamente a quanto descritto per *PMEmo*, anche in questo caso il dataset presenta un grande sbilanciamento in favore della classe A-V+ e ciò si riflette sulle classificazioni ottenute. Interessante notare come non ci sia nessuna predizione corretta della classe A-V- e A+V-. È possibile notare come molte delle predizioni errate siano falsi negativi della classe maggioritaria, indice di un modello predisposto a generalizzare su un insieme di dati sbilanciato.

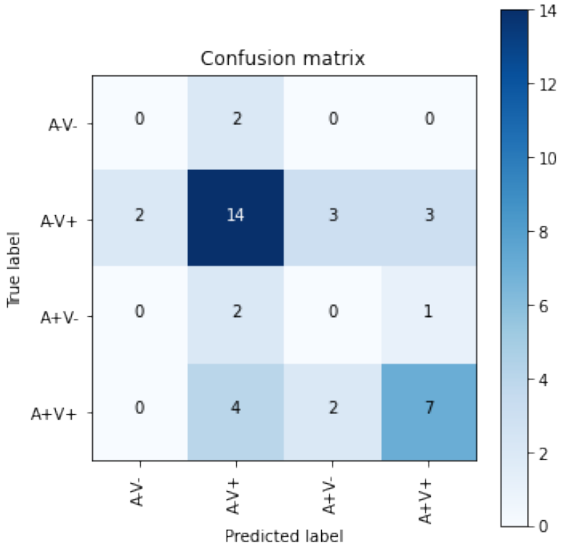
Al termine di questi esperimenti si ha la prova empirica che la fase di bilanciamento dei dati, attuata durante il pre-processing, sia stata necessaria al fine di addestrare un modello su un insieme bilanciato di dati, migliorando notevolmente le performance. Tuttavia rimane interessante notare come la rete riesca, in alcuni casi, comunque a mantenere performance buone, maggiori del 70% di accuratezza, anche utilizzando l'insieme sbilanciato dei dataset congiunti. Ciò dimostra che il modello preso in esame sia architetturalmente solido.



(a) 4Q



(b) PMEmo



(c) Emotify

Figura 5.11: Matrici di confusione associate ad ogni test set.

Capitolo 6

User-Tuning

Nel precedente capitolo è stato esposto come è stato possibile ottenere un modello generalizzato di deep learning, in ambito MER, per la classificazione delle emozioni partendo da spettrogrammi di Mel. I risultati ottenuti, sui dataset di partenza, sono stati incoraggianti e le percentuali di accuratezza rilevate dimostrano che il modello riesce a riconoscere correttamente le emozioni. Questa capacità risolve uno dei principali problemi dei sistemi di raccomandazione precedentemente citati, ovvero il cold start. Grazie ad esso si potrà avere una stima, e quindi un'etichetta, subito dopo che una canzone è stata rilasciata, consentendo al sistema di effettuare le raccomandazioni anche senza tag espliciti degli utenti. Inoltre, un utente appena iscritto alla piattaforma potrà usufruire fin da subito delle raccomandazioni per mood, basandosi sul modello generale. Tuttavia, le emozioni sono per loro natura soggettive, dunque la rete creata non è capace di riconoscere questa soggettività intrinseca nel problema: risulterà invece ottima nella generalizzazione del task, ovvero nel trovare quale emozioni sia mediamente più probabile rispetto alle altre. Esistono però utenti che hanno percezioni completamente fuori dall'ordinario: ad esempio chi usufruisce unicamente di musica metal è più probabile che tenderà a reputare noiose canzoni commerciali, rispetto a chi invece ascolta quella tipologia di musica. Pertanto, il modello generale non è altro che una base di partenza ed è necessario far sì che, per ogni utente, si crei una rete capace di considerare la soggettività dell'individuo, imparando a riconoscere le emozioni nello stesso modo in cui l'utente le percepisce. In deep learning, quando si parte da un modello generale e lo si raffina in uno più specifico, si parla di fine-tuning: ai fini del lavoro proposto è stato invece coniato il termine "*user-tuning*", sottolineando che il processo di fine-tuning è effettuato su ogni singolo utente. L'idea è stata quella di simulare un'applicazione reale, nella quale ogni soggetto può decidere di esprimere in maniera esplicita la sua percezione emotiva, rispetto ad un ascolto. Il modello imparerà dunque a capire il modo in cui l'utente percepisce la musica, basandosi su ognuna delle emozioni da esso indicate.

6.1 Modello User-Tuned

Il modello user-tuned nasce sulla base della rete generale, addestrata nel capitolo precedente: essa sarà il punto di partenza su cui si andrà ad effettuare fine-tuning su ogni utente. Nello specifico, si utilizza la rete CNN MobileNetV3, usata come feature extractor, e la rete LSTM come classificatore. I pesi utilizzati nel modello sono quelli che hanno ottenuto il miglior punteggio di accuratezza nel validation set, ovvero quelli che riescono meglio a generalizzare.

Si può pensare allo user-tuning come un sistema di Federated Learning (FL) [58], in cui si passa da una rete unica centralizzata ad un paradigma decentralizzato, dove ogni utente possiede una copia privata del modello, chiamato meta-modello [59], basato su dati di cui esso è proprietario.

Per effettuare una prova sperimentale e verificare in maniera empirica se il modello user-tuned riesca davvero ad adattarsi ai singoli utenti, è stato creato un nuovo dataset contenente 50 canzoni, scelte in maniera tale da ricoprire in maniera più organica ed eterogenea possibile i differenti generi musicali. La lunghezza media finale dei file audio è di 280 secondi. I titoli sono stati selezionati tra le canzoni più ascoltate, relative ai diversi generi di riferimento. I generi includono: rock, metal, punk, pop, commerciale, indie, rap, jazz, blues e cantautorato italiano. L'esperimento è stato eseguito su 3 utenti, alla quale è stato richiesto di ascoltare tutte le canzoni ed etichettarle rispetto alla loro emozione predominante all'ascolto. Ogni canzone è stata etichettata utilizzando la notazione valence-arousal descritta nel capitolo 4. Il soggetto poteva inoltre indicare 10 e 20 canzoni di cui voleva che il modello fosse a conoscenza della sua percezione emotiva, così da poter effettuare il fine-tuning.

6.1.1 Addestramento

Partendo dai file musicali, etichettati dall'utente e resi espliciti alla rete, l'addestramento del modello user-tuned è il seguente. Vengono generati gli spettrogrammi di Mel corrispondenti all'intera canzone, suddivisi in frame di 6 secondi ciascuno, ovvero la lunghezza su cui la CNN è stata addestrata ad estrarre le feature rilevanti. Tali immagini sono dunque etichettate con l'emozione suggerita dall'utente e aggiunti al dataset finale ottenuto nel capitolo 4, su cui si è effettuato il training del modello generale. Il modello subirà un processo di fine-tuning con questo nuovo dataset esteso, ricavando le feature dalla CNN ed addestrando il classificatore LSTM sul nuovo insieme di dati, utilizzando un learning rate estremamente piccolo (0.000001) ed attraverso 50 epoche. Il valore del learning rate è stato scelto empiricamente con lo scopo di evitare un eccessivo cambio di pesi durante il training, generando un'eccessiva variazione del modello ed esponendolo ad un overfitting fuori controllo. In questo modo si otterrà un nuovo modello, capace di adattarsi all'utente mantenendo l'abilità di generalizzare. Lo scopo principale di questa fase è quello di generare un overfitting "controllato", così da spingere il modello ad imparare le percezioni dello specifico utente. Al fine di ottenere questo effetto, sono stati utilizzati i nuovi spettrogrammi etichettati, aggiunti al

dataset originale anche come validation set, così da poter verificare che il modello abbia imparato correttamente a classificare le canzoni rispetto alla percezione soggettiva dell'utente. Anche in questo caso, il modello finale utilizzerà i pesi che hanno ottenuto il valore di accuratezza migliore sul validation set. Uno schema riassuntivo del processo descritto è visibile in figura 6.1.

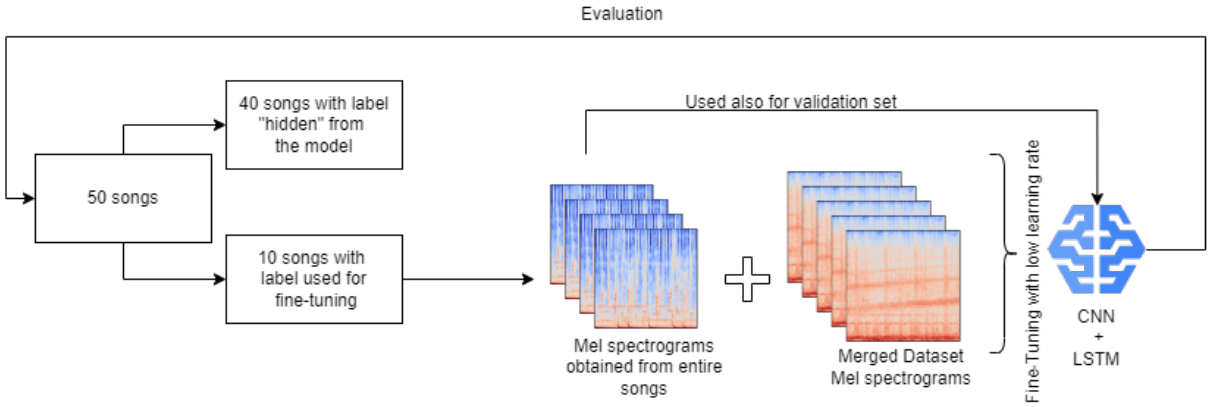


Figura 6.1: Schema riassuntivo del processo di user-tuning.

Com'è possibile osservare in figura 6.2, i risultati ottenuti sono ottimi, sia in termini di accuratezza che di loss. Applicando questa strategia, in fase di training il risultato raggiunto è di un'accuratezza del 89% sul validation set.

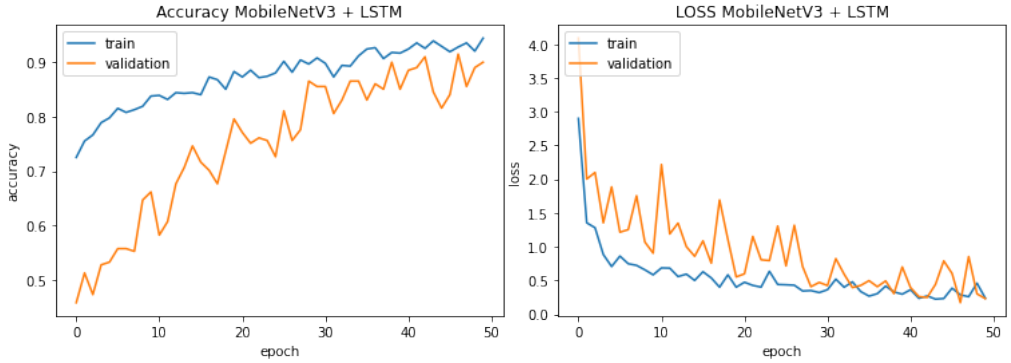


Figura 6.2: Performance della rete user-tuned durante il training.

6.1.2 Valutazione dei Risultati

Il modello proposto in questo lavoro è stato valutato utilizzando il nuovo dataset di 50 canzoni. La valutazione viene effettuata attraverso le informazioni suggerite dai 3 soggetti, impiegati nell'ascolto ed annotazione delle emozioni percepite. Gli esperimenti effettuati per ogni soggetto sono 3:

- 0 emozioni suggerite: si utilizzano le 50 annotazioni dell'utente sul modello finale ottenuto senza user-tuning. Risulta importante effettuare questa valutazione al fine di verificare come il modello generale si comporta in una casistica di uso reale;
- 10 emozioni suggerite: si addestra il modello su solamente 10 delle 50 etichette totali, utile per osservare come il modello si comporta basandosi su pochi suggerimenti dell'utente;
- 20 emozioni suggerite: si addestra il modello su 20 delle 50 etichette totali. Questo esperimento valuta come il modello si adatti all'utente con un numero modesto di suggerimenti. È atteso un miglioramento all'accuratezza del modello, rispetto alle altre casistiche.

Per effettuare questi esperimenti, ogni canzone è stata suddivisa in frame da 24 secondi ed ognuno di essi è stato classificato attraverso la rete LSTM. Ogni frame avrà quindi una predizione del modello: l'etichetta finale trovata, rispetto ad ogni canzone, sarà data dalla predizione che otterrà maggior frequenza rispetto alle 4 classi emozionali. Così facendo, per ogni canzone si otterrà non solo una singola stima emozionale, ma una lista con le percentuali di frequenza rispetto a tutte le classi, come mostrato in figura 6.3. Questo risulta molto utile per fare distinzioni tra due canzoni che hanno una stessa classe emozionale: alcune avranno una percentuale più alta dell'etichetta finale rispetto ad altre, potendo quindi fare ulteriori distinzioni al momento della raccomandazione. Questo argomento verrà trattato in maniera più estesa nella sezione successiva, relativa al sistema di raccomandazione e generazione playlist proposto.

```
------------------------------------------------------
File: ID-04.mp3
Predizioni:
[1, 1, 3, 2, 1, 1, 3, 1]
Mood in percentuale: [(1, 62.5), (3, 25.0), (2, 12.5)]
------------------------------------------------------
```

Figura 6.3: Esempio di output del modello relativo al file con ID = 04. Vengono mostrate le predizioni relative ai frame di 24 secondi ed infine la stima percentuale per ogni classe emozionale.

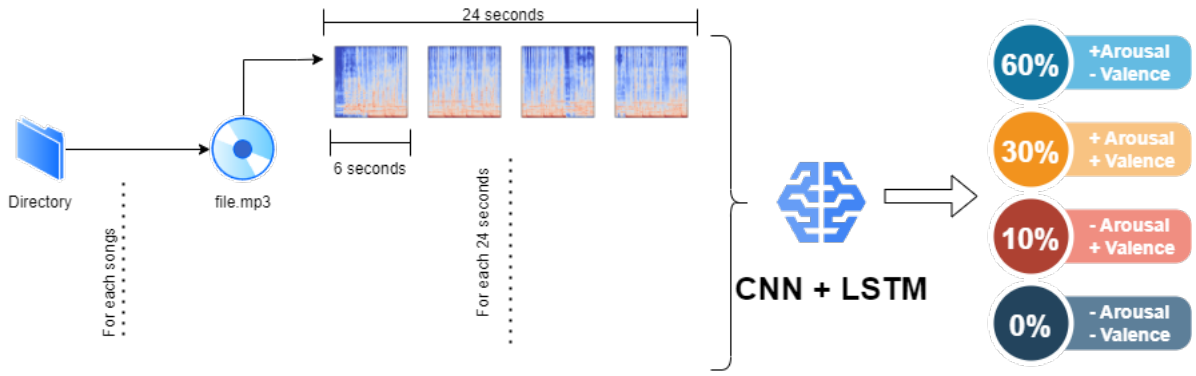


Figura 6.4: Diagramma dell'algoritmo di classificazione del modello user-tuned, applicato all'intero dataset di canzoni.

Dopo aver compilato un file *csv* per ogni esperimento svolto, annotando tutte le etichette trovate dai soggetti, è stato creato uno script Python che, preso in input la cartella contenente le 50 canzoni ed il file *csv*, esegue i seguenti step:

1. Per ogni file audio nella directory genera dei spettrogrammi di Mel, relativi alla durata di 6 secondi ciascuno;
2. Le feature rilevanti sono estratte dalla rete CNN MobileNetV3 considerando 4 spettrogrammi ad ogni iterazione, relativi a 24 secondi totali;
3. Le feature ottenute dai 4 spettrogrammi subiscono una trasformazione in un singolo vettore, contenente i 4 vettori di feature: questo passaggio è necessario per essere considerati un singolo input di 4 sequenze per la rete LSTM;
4. La rete LSTM genera diverse predizioni, una per ogni sequenza della lunghezza di 24 secondi;
5. Una stima percentuale di tutte le predizioni effettuate, rispetto ad una canzone, viene memorizzata. Tale stima mostra i vari mood che cambiano all'interno di una canzone. Il mood che ottiene la percentuale maggiore verrà selezionato come classe finale. Nel caso una canzone abbia due valori con la stessa stima percentuale come maggioritari ne verrà selezionato solo uno.

Un diagramma dell'algoritmo descritto è visibile in figura 6.4.

	A-V-	A-V+	A+V-	A+V+	Max_Overall_Emotion	User_Emotion	Wrong_Classification	Difference
0	0.0	100.0	0.0	0.0	1	1	False	0.0
1	60.0	40.0	0.0	0.0	0	0	False	0.0
2	11.1	11.1	11.1	66.7	3	0	True	55.6
3	14.3	0.0	14.3	71.4	3	1	True	71.4
4	12.5	0.0	62.5	25.0	2	1	True	62.5

Figura 6.5: Esempio del risultato ottenuto su 5 canzoni dopo l'esecuzione del modello. Le prime 4 colonne corrispondono alle stime percentuali dei quadranti di Russell. La colonna *Max_Overall_Emotion* corrisponde alla classe emozionale con la frequenza rilevata più elevata. La *User_Emotion* rappresenta l'etichetta annotata dal soggetto. Nel caso le due colonne non contengano lo stesso valore l'errore (*Difference*) è calcolato.

Al fine di valutare le performance del modello user-tuned non basta utilizzare semplicemente l'accuratezza, basata sulla stima dell'emozione più frequente. È stato ritenuto opportuno introdurre una nuova metrica di errore, per quantificare la distanza tra la stima ottenuta come risultato della classificazione e l'emozione reale annotata dal soggetto, nel caso le due etichette divergano. Questa metrica stabilisce quanto il modello si è sbagliato nell'effettuare la predizione: un valore alto significa che la stima dell'emozione annotata dall'utente è stata nulla, ovvero non è stata mai rilevata nella canzone. D'altro canto, un valore basso, ad esempio minore di 25, significa che, anche se l'emozione non è stata trovata correttamente, il modello non ha sbagliato di molto. Tale stima è quindi calcolata come la differenza tra la frequenza dell'emozione predetta (quella che ha ottenuto il valore maggiore) e la frequenza della vera classe emozionale del soggetto. Una volta ottenute tutte le stime di errore è stato possibile calcolare l'errore medio totale del modello. È stata utilizzata la seguente formula:

$$\forall_i : \text{ wrong emotion}_i > \text{true emotion}_i :$$
$$\frac{1}{n = \text{Number Wrong Predictions}} \sum_{i}^{n} \text{wrong emotion}_i - \text{true emotion}_i \tag{6.1}$$

Un errore medio alto significa che il modello non sta performando bene, inoltre grazie a questa metrica è possibile valutare come la rete si comporta aumentando il numero di canzoni suggerite nel processo di user tuning. La figura 6.5 mostra un esempio di output, relativo a 5 canzoni (righe nella tabella) del classificatore user-tuned, in termini di frequenza di ogni classe emozionale, la corrispondente classe predetta dal classificatore (colonna Max_Overall_Emotion), comparata alla vera etichetta (User_Emotion) ed in fine la metrica di errore proposta (Difference).

Il risultato medio, ottenuto sulle 50 canzoni, per ogni utente e configurazione, è riportato rispettivamente nelle tabelle 6.1 e 6.2. I risultati mostrano diversi aspetti importanti da considerare:

- Per quanto riguarda il modello generale, ovvero quello non soggetto ad user-tuning, è possibile osservare che la media di accuratezza è del 45%. Tale valore, contestualizzato in un task di classificazione multi-classe ed in contesto reale risulta più che accettabile.

- In riferimento ai modelli soggetti ad user-tuning, i risultati sono ottimi. Come atteso, le canzoni suggerite al modello risultano sempre corrette nella classificazione. L'accuratezza, mostrata nei grafici 6.6 e 6.7, ottiene un netto miglioramento all'aumentare delle canzoni suggerite. Ciò dimostra che il sistema proposto di user-tuning, valutato in un contesto reale, comporta un'esperienza utente migliore, basata su un modello che impara a riconoscere le percezioni soggettive, in un costante processo di miglioramento tramite l'utilizzo del sistema. Un altro aspetto fondamentale da tenere in considerazione è la metrica di differenza errore proposta: è chiaramente visibile in figura 6.7 come questa subisca una decrescita costante all'aumentare delle canzoni suggerite. Questo fenomeno è spiegabile come un sostanziale miglioramento nel modello, non solo sulle canzoni suggerite, che quindi impara in maniera diretta a classificare correttamente, ma anche su quelle che non vengono classificate correttamente. La rete, infatti, imparando le percezioni dell'utente, riesce a classificare in maniera sempre migliore anche le canzoni che non sono suggerite al modello. Questo addestramento passivo è molto importante perché consente di classificare correttamente anche nuove canzoni mai viste dal modello, senza che l'utente debba per forza indicare direttamente l'emozione provata all'ascolto.

A fronte di questi valori è possibile affermare che i risultati ottenuti risultino promettenti, tuttavia è necessario estendere lo studio ad ulteriori soggetti, al fine di determinare se, con un bacino di utenza più ampio ed eterogeneo, le performance mantengano questo livello di precisione.

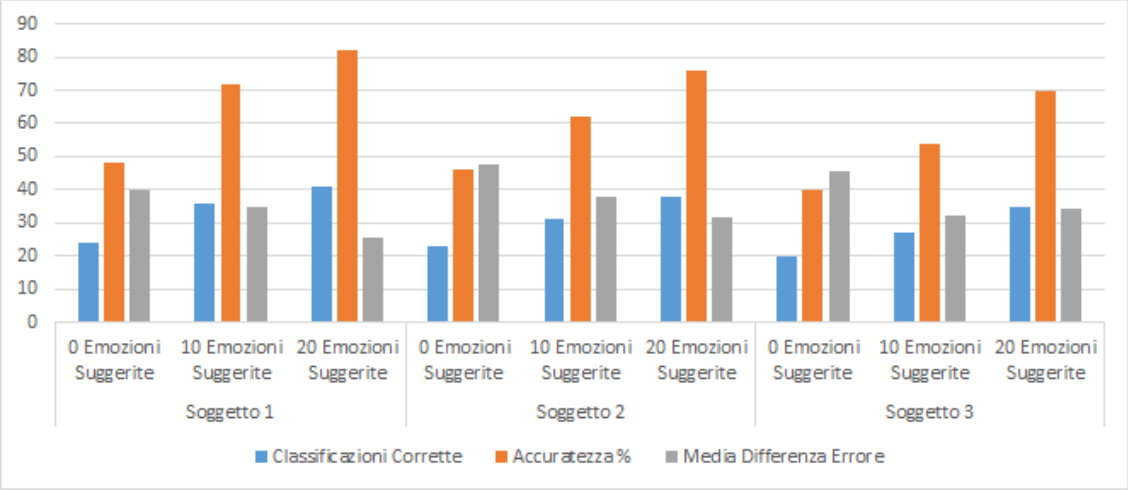


Figura 6.6: Grafico risultati ottenuti basati sui tre soggetti.

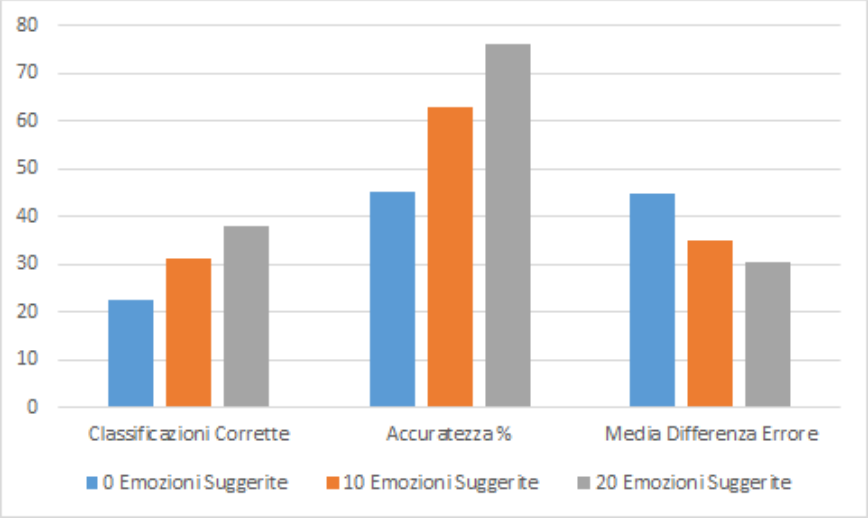


Figura 6.7: Grafico risultati medi ottenuti per configurazione.

Tabella 6.1: Valutazione basata sui tre soggetti.

Soggetto	User-Tuning	Classificazioni Corrette	Accuratezza %	Media Differenza Errore
1	0 Emozioni Suggerite	24	0.48	40.03
	10 Emozioni Suggerite	36	0.72	34.92
	20 Emozioni Suggerite	41	0.82	25.29
2	0 Emozioni Suggerite	23	0.46	47.73
	10 Emozioni Suggerite	31	0.62	37.88
	20 Emozioni Suggerite	38	0.76	31.80
3	0 Emozioni Suggerite	20	0.40	45.83
	10 Emozioni Suggerite	27	0.54	32.20
	20 Emozioni Suggerite	35	0.70	34.43

Tabella 6.2: Valori medi per ogni configurazione.

User-Tuning	Classificazioni Corrette	Accuratezza %	Media Differenza Errore
0 Emozioni Suggerite	22.34	0.45	44.62
10 Emozioni Suggerite	31.33	0.63	35
20 Emozioni Suggerite	38	0.76	30.51

Capitolo 7

Sistema di Raccomandazione

L'ultimo tassello mancante, del sistema proposto in questo lavoro, è quello riguardante l'impiego del modello creato, ovvero il sistema di raccomandazione con cui l'utente si interfaccerà. Come descritto nel capitolo relativo allo stato dell'arte dei sistemi di raccomandazione musicale, la tipologia di modello adottata è quella utilizzata nei sistemi model-based: l'architettura è basata su un modello di deep learning che effettua classificazioni basate sullo specifico utente. Le feature utilizzate entrano nella tassonomia dei sistemi content-based audio analysis, ovvero quelli che utilizzano informazioni direttamente ricavate dai file audio per effettuare le predizioni. Il progetto, tuttavia, non ha l'obiettivo di creare un sistema di raccomandazione classico in ambito musicale, in cui l'utente può filtrare tra le sue canzoni per trovare quelle con il mood richiesto, ma fa un passo ulteriore. È stato posto il focus su un sistema che potesse generare vere e proprie playlist basate sul mood di riferimento dell'utente.

7.1 Generatore di Playlist

Nell'ambito dei sistemi di raccomandazione musicali, i generatori automatici di playlist possono essere considerati una loro sotto-categoria specializzata. Per generare una playlist è necessario sfruttare un sistema di raccomandazione al fine di creare un elenco nelle corde dell'utente, che contenga ciò che esso ha voglia di ascoltare in quel momento. Una playlist non è altro che una lista di riproduzione musicale, dove i titoli generalmente appartengono ad un insieme logico scelto dall'utente. Ad esempio, esistono playlist basate sui generi musicali, sul mood delle canzoni, sulla data d'uscita dell'album di riferimento, sull'evento dove dovrà essere riprodotta e tanti altri. Nel caso relativo al progetto, la playlist da generare sarà basata sul mood delle canzoni, rilevato dal modello user-tuned. L'idea principale è stata quella di simulare un'applicazione reale, integrata in un'app mobile di riproduzione audio, oppure in una delle piattaforme di streaming musicale presenti sul mercato, dove un utente può dichiarare il proprio umore di partenza e quello che vorrebbe raggiungere alla fine dell'ascolto. Il sistema genererà una playlist che avrà nelle canzoni al suo interno una graduale variazione nell'emozione

suscitata nell'utente, così da consentirgli di raggiungere lo stato emotivo atteso. Un'illustrazione è mostrata in figura 7.1.

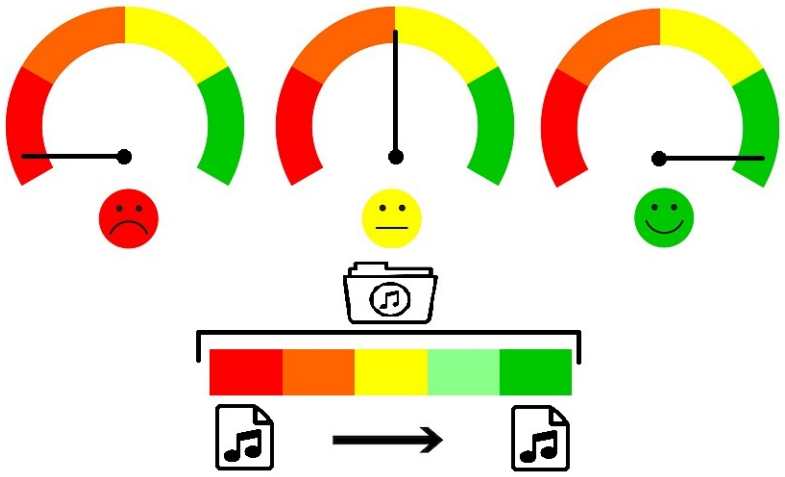


Figura 7.1: Illustrazione relativa al funzionamento della playlist generata. La playlist è strutturata per variare l'umore dell'utente durante l'ascolto in maniera graduale.

7.2 Integrazione Modello User-Tuned

Come descritto nel capitolo precedente, lo scopo finale di questo lavoro non è solo il dimostrare la fattibilità nel creare un modello di Deep Learning capace di riconoscere il mood delle canzoni e adattarsi alla percezione dell'utente. L'obiettivo è quello di rendere usabile tale modello in un contesto reale di utilizzo, da parte di diversi utenti. L'output generato dalla rete è stato studiato fin dal principio per essere usabile da un sistema di raccomandazione: invece di generare un'unica classificazione per ogni canzone, si effettua anche una stima percentuale dei mood contenuti al suo interno, come è stato possibile osservare nella figura 6.5 della sezione precedente. L'utente finale potrà quindi selezionare il suo umore di partenza e quello che vorrà raggiungere alla fine dell'ascolto della playlist e, tramite una variazione delle percentuali emozionali contenute in ogni canzone, tale passaggio di mood sarà reso guidato e mai repentino. La playlist generata partirà la riproduzione dalle canzoni che hanno una percentuale emozionale, che corrisponde a quella dell'utente, con valori elevati e poco per volta scenderà di percentuale lasciando spazio a quella desiderata. Questo passaggio risulta importante per non creare nell'utente un salto di umore troppo drastico: ad esempio, se esso si trova in uno stato emotivo triste (basso arousal e valence negativa) e vuole raggiungere un mood felice (alto arousal e valence positiva), l'iniziare a riprodurre canzoni classificate come il target richiesto fin dall'inizio potrebbe infastidire e far terminare l'ascolto.

7.3 Algoritmo Progettato

Ai fini di implementare ciò che è descritto nella sezione precedente è stato sviluppato il seguente algoritmo:

1. Si richiede all'utente di definire il mood attuale e quello target. Inoltre, viene richiesto il numero di canzoni della quale dovrà essere composta la playlist (*playlist_length*).

2. Si verifica che il mood di partenza e quello target non coincidano: nel caso siano uguali la playlist sarà generata usando canzoni con valore percentuale elevato dell'emozione selezionata;

3. Si inizializzano le variabili *starting_value* ed *target_value* rispettivamente a 100 e 0. Queste variabili rappresentano la stima percentuale dell'emozione di partenza dell'utente (*starting_value* = 100) e quella target da raggiungere (*target_value* = 0).

4. Si definisce la variabile *jump*, ovvero quanto dovrà aumentare la percentuale del mood target ad ogni iterazione e quanto dovrà diminuire quella attuale. La funzione per trovare il valore ottimale è definita come:

$$jump = \frac{100}{playlist\_length - 1} \tag{7.1}$$

5. Si definisce la funzione *find_nearest*: dato in input il dataframe contenente tutte le predizioni del modello, il valore percentuale del mood di partenza e il valore percentuale target, essa restituisce l'elemento che ha la distanza tra i due valori di input minore. Ad esempio, se in input si passano i valori di partenza = 70 e target = 30, la funzione restituisce l'elemento che ha le percentuali più vicine ai valori passati (es. partenza = 75, target = 25) tra tutti gli elementi del dataframe delle predizioni.

6. Si itera sulla *playlist_length* e, a ogni iterazione, viene trovata la canzone con la minor distanza dai valori di riferimento tramite la funzione *find_nearest*, aumentando la percentuale target e diminuendo quello iniziale di *jump*. Un esempio di output ottenuto in questa fase è visibile in figura 7.2.

7. Si riproduce la playlist ottenuta, come in figura 7.3.

L'algoritmo, scritto in Python, nella sua interezza è visibile in appendice A.

	ID	A-V-	A-V+	A+V-	A+V+
0	Nirvana - All Apologies	75.0	12.5	0.0	12.5
1	Radiohead - Fake Plastic Trees	58.3	25.0	0.0	16.7
2	Muse - Feeling Good	50.0	12.5	12.5	25.0
3	David Bowie - Space Oddity	38.5	30.8	0.0	30.8
4	Machine Gun Kelly - Home	33.3	0.0	22.2	44.4
5	System Of A Down - Lonely Day	16.7	33.3	0.0	50.0
6	twenty one pilots - Ride	22.2	0.0	11.1	66.7
7	Soundgarden - Black Hole Sun	23.1	7.7	0.0	69.2
8	Rino Gaetano - Ma il cielo sempre pi blu	5.0	5.0	0.0	90.0
9	The Rolling Stones - Sympathy For The Devil	0.0	6.7	0.0	93.3

Figura 7.2: Esempio di output ottenuto con parametri *starting_mood* = A-V-, *target_mood* = A+V+ e *playlist_length* = 10.

```
Riproducendo:../input/my-playlist/Playlist/Nirvana - All Apologies.mp3
Riproducendo:../input/my-playlist/Playlist/Radiohead - Fake Plastic Trees.mp3
Riproducendo:../input/my-playlist/Playlist/Muse - Feeling Good.mp3
Riproducendo:../input/my-playlist/Playlist/David Bowie - Space Oddity.mp3
Riproducendo:../input/my-playlist/Playlist/Machine Gun Kelly - Home.mp3
Riproducendo:../input/my-playlist/Playlist/System Of A Down - Lonely Day.mp3
Riproducendo:../input/my-playlist/Playlist/twenty one pilots - Ride.mp3
Riproducendo:../input/my-playlist/Playlist/Soundgarden - Black Hole Sun.mp3
Riproducendo:../input/my-playlist/Playlist/Rino Gaetano - Ma il cielo  sempre pi blu.mp3
```

Figura 7.3: Esempio di output finale dello script, i file audio corrispondenti alle canzoni vengono riprodotti.

Capitolo 8

Conclusione

In questa tesi è stato presentato un nuovo approccio, basato sulle emozioni, per i sistemi di raccomandazione e generazione di playlist. L'applicazione della tecnica MER permette di riconoscere le emozioni, indotte dalle canzoni, al fine di creare diverse playlist personalizzate sulla base dello stato emozionale dell'utente che usufruisce del sistema. Tale metodologia consente di accentuare il focus sul singolo soggetto, migliorando l'esperienza di utilizzo e risolvendo uno dei problemi principali dei sistemi di raccomandazione: il "cold start".

Nella prima parte è stata esplorata la letteratura di riferimento delle diverse tipologie di sistemi di raccomandazione, in ambito di Music Emotion Recognition, e lo stato dell'arte delle possibili metodologie applicabili. Successivamente sono stati descritti i dataset utilizzati, andando a definire per ognuno di essi le caratteristiche specifiche e le operazioni di pre-processing necessarie all'utilizzo. Infine, sono stati esposti i dettagli tecnici di progettazione.

Il sistema ottenuto al termine della progettazione è basato su due diversi modelli: uno generale e uno "user-tuned". Quello generale è stato progettato impiegando diverse reti neurali: CNN, per l'estrazione di feature rilevanti da spettrogrammi di Mel associati alle canzoni, e LSTM come classificatore delle sequenze ottenute. Il sistema risultante, addestrato sull'unione bilanciata di diversi dataset presenti in letteratura, ha ottenuto ottimi risultati in fase di training, dimostrando la possibilità, da parte di nuovi utenti, di utilizzarlo fin dal primo accesso all'applicazione. Un altro vantaggio importante nell'impiego di questo modello è la capacità di applicarlo non solo a nuovi utenti, ma anche a canzoni mai etichettate prima, osservando la risoluzione di ciò che è definito cold start. Il modello "user-tuned", ottenuto tramite tecnica di fine-tuning applicata a quello generale, consente di imparare direttamente dall'utente il suo modo di percepire emozionalmente la musica, basandosi su alcuni suggerimenti diretti ed adattandosi, migliorando notevolmente l'esperienza. I risultati empirici, ottenuti tramite prove sperimentali applicate a tre soggetti, hanno dimostrato che il modello "user-tuned" riesce a perfezionare le inferenze delle canzoni non suggerite direttamente al sistema, abbassando l'errore di predizione. Infine, la generazione della playlist proposta utilizza le informazioni ricavate dal modello per garantire un passaggio dall'umore attuale dell'utente a quello desiderato in maniera graduale, utilizzan-

do le canzoni dell'utente etichettate dal sistema ed evitando cambiamenti troppo repentini che potrebbero rovinare l'esperienza di ascolto. A fronte dei risultati ottenuti è possibile immaginare l'integrazione del sistema proposto in piattaforme di streaming o applicazioni di riproduzione musicale, le quali difficilmente tengono conto della dimensione emozionale dell'utente.

8.1 Sviluppi Futuri

Esistono diversi possibili sviluppi futuri applicabili al progetto di tesi:

- I risultati ottenuti, per quanto promettenti, risultano limitati ad un piccolo bacino di utenti. Effettuare nuovi test, basati su un campione di utenza più ampio, è un passo necessario al confermare le potenzialità, e trovare eventuali limiti, del sistema proposto.
- Le architetture delle reti neurali utilizzate nella creazione del sistema possono essere ulteriormente migliorate, facendo uso di nuove tecnologie che stanno prendendo sempre più piede in ricerca chiamate *transformer*. Tali modelli, applicati inizialmente a task di NLP, si stanno dimostrando sempre più robusti in applicazioni di pattern analysis, andando a superare le performance di reti CNN [60]. Sebbene le reti convoluzionali abbiano al momento una letteratura di riferimento più ampia e consolidata, i transformer stanno avendo molto risalto ed è possibile che in un futuro prossimo soppianteranno completamente le prime.
- I dataset utilizzati presentavano alcuni difetti nel bilanciamento dei dati, rendendo necessario una fase di pre-processing per ridurre il problema. Un possibile sviluppo futuro è dato dall'ampliamento di questi dataset con ulteriori esempi, al fine di utilizzare una maggiore mole di dati in fase di addestramento, migliorando la capacità del modello di generalizzare. Un'ulteriore possibilità è quella di applicare strumenti di domain adaptation basati su istanze, come Kliep [61] e TrAdaBoost [62]. Tali tecniche consentono di bilanciare diverse fonti di dati, andando a cambiare il peso di ogni istanza in base alla sua fonte di appartenenza, come mostrato in figura 8.1.

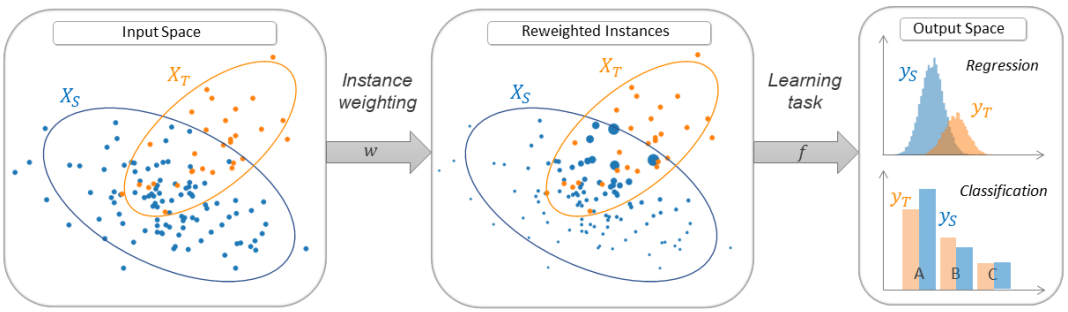


Figura 8.1: Funzionamento di metodi domain adaptation basati su istanze[1].

- Infine, una possibile estensione del progetto è quella di utilizzare segnali ottenuti tramite sensori indossabili, ad esempio braccialetti per il riconoscimento dello stato emozionale in tempo reale, entrando nel campo del Wearable Computing. Gli studi presenti in letteratura riguardanti l'utilizzo di segnali fisiologici ottenuti direttamente da dispositivi indossabili hanno mostrato ottimi risultati empirici, con percentuali di accuratezza nella stima dell'emozione dell'utente elevate. Ad esempio Ayata *et al.* [63] hanno proposto un sistema di classificazione delle emozioni attraverso un modello che utilizza segnali fisiologici multi-canale basati su *galvanic skin response* (GSR) e *photo-plethysmography* (PPG) per inferire l'emozione nello spazio continuo arousal-valence. Questo studio utilizza tali segnali per integrare un sistema di raccomandazione esistente, estendendone le capacità e migliorando le raccomandazioni. L'utilizzo di uno strumento simile nel progetto consentirebbe di rimuovere il bisogno di richiedere esplicitamente all'utente l'emozione attuale, in fase di generazione della playlist, riconoscendola automaticamente. Ciò porterebbe anche ad avere una maggior quantità di dati per effettuare un addestramento più accurato per la fase di user-tuning del modello.

[1] Source: `https://adapt-python.github.io/adapt/contents.html#adapt-instance-based`

Appendice A

Codice Python Generazione Playlist

```
# dati relativi alla creazione playlist, personalizzabili
playlist_length = 10
starting_mood = 'A-V-'
expected_mood = 'A-V-'

# funzione che prende in input le colonne relative alla stima del mood e il valore atteso a cui esse devono avvicinarsi
def find_nearest(array_starting, value_starting, array_target, value_target):

    # gli array contengono la differenza tra il valore reale e quello atteso
    starting = np.abs(array_starting - value_starting)
    target = np.abs(array_target - value_target)

    # si sommano i valori degli array
    nearest_array = starting + target

    # l'elemento minore sara quello con i due mood combinati che piu si avvicinano all obiettivo
    return nearest_array.argmin()

# si inizializza la lista di riproduzione vuota
song_list = []


if starting_mood != expected_mood:
    # caso nel quale l'utente vuole shiftare mood
    jump = (100/(playlist_length-1))
    starting_value = 100
    expected_value = 0




    # si itera sulla lunghezza della plylist
```

```
33      for i in range(playlist_length):
34
35          # viene trovato l'indice della canzone da aggiungere
36          idx = find_nearest(df[starting_mood].to_numpy(),
    starting_value, df[expected_mood].to_numpy(), expected_value)
37
38          # alla playlist viene aggiunta la canzone
39          song_list.append(df.iloc[idx]['ID'])
40
41          # si cancella la canzone aggiunta dal dataframe per evitare
    di aggiungerla due volte alla playlist
42          df = df.drop(idx)
43
44          # l'indice del dataframe viene resettato
45          df = df.reset_index(drop=True)
46
47          # i valori di partenza e target vengono aggiornati in base
    al valore del salto
48          starting_value = starting_value - jump
49          expected_value = expected_value + jump
50
51  else: # il mood dell'utente non deve cambiare, si la playlist cerca
    tutte le canzoni con il mood richiesto massimizzato
52      df = df.sort_values(by=[starting_mood])
53      df = df[len(df)-playlist_length:]
54      song_list = df['ID'].to_numpy()
```

Listing A.1: Playlist-Generator.py

Bibliografia

[1] Y.-S. Seo and J.-H. Huh, "Automatic emotion-based music classification for supporting intelligent iot applications," *Electronics*, vol. 8, no. 2, p. 164, 2019.

[2] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, "A review on deep learning for recommender systems: challenges and remedies," *Artificial Intelligence Review*, vol. 52, pp. 1–37, 2018.

[3] G. Liu and Z. Tan, "Research on multi-modal music emotion classification based on audio and lyirc," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, vol. 1. IEEE, 2020, pp. 2331–2335.

[4] R. Panda, R. Malheiro, and R. P. Paiva, "Novel audio features for music emotion recognition," *IEEE Transactions on Affective Computing*, vol. 11, no. 4, pp. 614–626, 2018.

[5] A. Soleimanipour, M. Azadbakht, and A. Asl, "Cultivar identification of pistachio nuts in bulk mode through efficientnet deep learning model," *Journal of Food Measurement and Characterization*, vol. 16, pp. 1–11, 08 2022.

[6] D. Hossain, M. Imtiaz, T. Ghosh, V. Raju, and E. Sazonov, "Real-time food intake monitoring using wearable egocnetric camera," vol. 2020, 07 2020, pp. 4191–4195.

[7] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[8] M. Hollemans. [Online]. Available: https://machinethink.net/blog/mobile-architectures/

[9] J. A. Konstan and J. Riedl, "Recommender systems: from algorithms to user experience," *User Model. User-Adap. Inter.*, vol. 22, no. 1, pp. 101–123, Apr 2012.

[10] "Ascolto musica in Italia 2021," Feb 2022, [Online; accessed 11. Feb. 2022]. [Online]. Available: https://www.scfitalia.it/notizie/engaging-with-music-2021.kl

[11] L. J. Lehmberg and C. Fung, "Benefits of music participation for senior citizens: A review of the literature," 2010.

[12] T. Eerola and H.-R. Peltola, "Memorable Experiences with Sad Music—Reasons, Reactions and Mechanisms of Three Types of Experiences," *PLoS One*, vol. 11, no. 6, p. e0157444, Jun 2016.

[13] Y. L. Ferguson and K. M. Sheldon, "Trying to be happier really can work: Two experimental studies," *Journal of Positive Psychology*, vol. 8, no. 1, pp. 23–33, Jan 2013.

[14] C. L. Krumhansl, "An exploratory study of musical emotions and psychophysiology. - PsycNET," Feb 2022, [Online; accessed 11. Feb. 2022].

[15] M. T. Mitterschiffthaler, C. H. Y. Fu, J. A. Dalton, C. M. Andrew, and S. C. R. Williams, "A functional MRI study of happy and sad affective states induced by classical music," *Hum. Brain Mapp.*, vol. 28, no. 11, pp. 1150–1162, Nov 2007.

[16] C. V. O. Witvliet and S. R. Vrana, "Play it again sam: Repeated exposure to emotionally evocative music polarises liking and smiling responses, and influences other affective reports, facial emg, and heart rate," *Cognition and Emotion*, vol. 21, no. 1, pp. 3–25, 2007. [Online]. Available: https://doi.org/10.1080/02699930601000672

[17] L. F. Barrett, "Are Emotions Natural Kinds?" *Perspect. Psychol. Sci.*, vol. 1, no. 1, pp. 28–58, Mar 2006.

[18] S. Knuuttila, *Emotions in ancient and medieval philosophy.* Clarendon Press, 2004.

[19] V. Chaturvedi, A. B. Kaur, V. Varshney, A. Garg, G. S. Chhabra, and M. Kumar, "Music mood and human emotion recognition based on physiological signals: a systematic review," *Multimedia Systems*, vol. 28, no. 1, pp. 21–44, Feb 2022.

[20] R. W. Picard, *Affective computing.* MIT press, 2000.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] R. Plutchik and H. Kellerman, *Theories of emotion.* Academic Press, 2013, vol. 1.

[25] J. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, pp. 1161–1178, 12 1980.

[26] "AI-Based Recommendation Systems - InData Labs," June 2021. [Online]. Available: https://indatalabs.com/blog/ai-based-recommender-system

[27] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705113001044

[28] Y. E. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. Scott, J. A. Speck, and D. Turnbull, "Music emotion recognition: A state of the art review," in *Proc. ismir*, vol. 86, 2010, pp. 937–952.

[29] R. Malheiro, R. Panda, P. Gomes, and R. P. Paiva, "Music emotion recognition from lyrics: A comparative study," 09 2013.

[30] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[31] M. Levy and M. Sandler, "A semantic space for music derived from social tags," *Austrian Compuer Society*, vol. 1, p. 12, 2007.

[32] B.-j. Han, S. Rho, S. Jun, and E. Hwang, "Music emotion classification and context-based music recommendation," *Multimedia Tools and Applications*, vol. 47, no. 3, pp. 433–460, 2010.

[33] H. Bahuleyan, "Music genre classification using machine learning techniques," *arXiv preprint arXiv:1804.01149*, 2018.

[34] L. Roberts, "Understanding the mel spectrogram," Aug 2022. [Online]. Available: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[36] S. Hizlisoy, S. Yildirim, and Z. Tufekci, "Music emotion recognition using convolutional long short term memory deep neural networks," *Engineering Science and Technology, an International Journal*, vol. 24, no. 3, pp. 760–767, 2021.

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[38] K. Zhang, H. Zhang, S. Li, C. Yang, and L. Sun, "The pmemo dataset for music emotion recognition," in *Proceedings of the 2018 acm on international conference on multimedia retrieval*, 2018, pp. 135–142.

[39] R. Panda, R. Malheiro, and R. P. Paiva, "Musical texture and expressivity features for music emotion recognition," in *19th International Society for Music Information Retrieval Conference (ISMIR 2018)*, 2018, pp. 383–391.

[40] A. Aljanaki, F. Wiering, R. Veltkamp *et al.*, "Collecting annotations for induced musical emotion via online game with a purpose emotify," *Technical Report Series*, vol. 2014, no. UU-CS-2014-015, 2014.

[41] A. Aljanaki, F. Wiering, and R. C. Veltkamp, "Studying emotion induced by music through a crowdsourcing game," *Information Processing & Management*, vol. 52, no. 1, pp. 115–128, 2016.

[42] M. Zentner, D. Grandjean, and K. R. Scherer, "Emotions evoked by the sound of music: characterization, classification, and measurement." *Emotion*, vol. 8, no. 4, p. 494, 2008.

[43] X. Ma, Z. Wu, J. Jia, M. Xu, H. Meng, and L. Cai, "Emotion recognition from variable-length speech segments using deep learning on spectrograms." in *Interspeech*, 2018, pp. 3683–3687.

[44] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8. Citeseer, 2015, pp. 18–25.

[45] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[46] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu *et al.*, "Gpipe: Efficient training of giant neural networks using pipeline parallelism," *Advances in neural information processing systems*, vol. 32, 2019.

[47] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[48] K. Tung, "Flowers Dataset," 2020. [Online]. Available: https://doi.org/10.7910/DVN/1ECTVN

[49] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

[50] A. Pujara, "Image classification with mobilenet," Jul 2020. [Online]. Available: https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470

[51] A. F. Agarap, "Deep learning using rectified linear units (relu)," *CoRR*, vol. abs/1803.08375, 2018. [Online]. Available: http://arxiv.org/abs/1803.08375

[52] A. Nagpal, "L1 and l2 regularization methods," Oct 2017. [Online]. Available: https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c

[53] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," 2018. [Online]. Available: https://arxiv.org/abs/1805.07836

[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[55] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

[56] T. Tieleman and G. Hinton, "Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning," *COURSERA Neural Networks Mach. Learn*, 2012.

[57] W. Wu and S. Yang, "Leveraging intra and inter-dataset variations for robust face alignment," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 2096–2105.

[58] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[59] J. Vanschoren, "Meta-learning," in *Automated machine learning*. Springer, Cham, 2019, pp. 35–61.

[60] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on vision transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022.

[61] M. Sugiyama, S. Nakajima, H. Kashima, P. Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," *Advances in neural information processing systems*, vol. 20, 2007.

[62] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1855–1862.

[63] "Emotion based music recommendation system using wearable physiological sensors," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 2, pp. 196–203, 2018.