



UNIVERSITA' DEGLI STUDI DI BARI

CORSO DI LAUREA IN INFORMATICA

TESI DI LAUREA

IN

SISTEMI AD AGENTI

**EmoFEATURES: RICONOSCIMENTO DELLE EMOZIONI
DALLA VOCE**

Relatore

Chiar.ma Prof.ssa Berardina De Carolis

Laureando

Michele Metta

ANNO ACCADEMICO 2018\2019

Indice

INTRODUZIONE	5
 CAPITOLO 1 - L'emozione.....	7
1.1 Il modello discreto delle emozioni	8
1.2 La teoria di Ekman	8
1.3 Il modello dimensionale delle emozioni	10
1.4 Il modello circonflesso delle emozioni di Russell.....	11
1.5 Il modello di Plutchik	12
1.6 L'affective computing	14
1.7 Speech emotion recognition	17
1.8 Analisi e classificazione del parlato.....	18
 CAPITOLO 2 - Dataset.....	21
2.1 Dataset Ravdess (Ryerson Audio-Visual Database of Emotional Speech and Song)	21
2.2 Dataset Tess (Toronto emotional speech set).....	22
2.3 Dataset Savee (Surrey Audio-Visual Expressed Emotion).....	23
2.4 Dataset Emovo.....	24
2.5 Dataset EmoFilm.....	25
2.6 Dataset Emodb (Berlin Database of Emotional Speech)	26
2.7 Dataset Demos (Database of Elicited Mood in Speech)	27
 CAPITOLO 3 - ESTRAZIONE FEATURES AUDIO	28
3.1 Pre-elaborazione audio	28
3.2 Mfcc (Mel-frequency-Cepstral-Coefficient)	29
3.3 Mfcc Delta e Delta-Delta	33
3.4 Pitch.....	34
3.5 Zero-Crossing-Rate (ZCR)	34
3.6 Energy	34

Capitolo 4 - Classificatori	37
4.1 Random Forrest	37
4.2 Gradient Boosting	38
4.3 Support Vector Machine (SVM).....	39
4.4 Multi-Layer Perceptron (MLP).....	41
4.5 K-Nearest Neighbors.....	44
 Capitolo 5 - Sperimentazione e Risultati.....	 45
5.2 Tess	46
5.3 Savee.....	47
5.4 Emovo.....	48
5.5 Emofilm	49
5.6 Emodb	49
5.7 Demos	50
5.8 Unico dataset con audio del bilanciamento	51
5.9 Analisi dei risultati:.....	52
 Capitolo 6 Convolutional Neural Network (CNN)	 53
6.1 Convolutional Layer	54
6.2 Rectified Linear Unit.....	59
6.3 Pooling Layer	59
6.4 Fully Connected Layer.....	60
6.5 Softmax Unit.....	61
6.6 Cross-Entropy	62
6.7 Dropout Layer.....	62
6.8 Data Augmentation	63
6.9 Spettrogramma	65
6.10 Architettura.....	66
6.11 Pre-processing.....	67
6.12 Risultati ottenuti.....	68

Capitolo 7 - Sperimentazione finale in the wild	71
7.1 Risultati ottenuti.....	71
 Capitolo 8 - Applicazione EMOFEATURES per Windows.....	 73
8.1 Interfaccia grafica	73
8.2 Esempi di funzionamento del programma	78
 Capitolo 9 - Conclusioni e sviluppi futuri	 81
 Ringraziamenti	 82
Bibliografia	83

INTRODUZIONE

Il riconoscimento delle emozioni è da sempre stato uno dei compiti più affascinanti e stimolanti per chi si occupa di apprendimento automatico. Grazie infatti ad alcune tecniche di analisi e di elaborazione di segnali audio è possibile insegnare ad una macchina il riconoscimento delle principali emozioni umane attraverso l'uso della voce. Dopo che il segnale audio viene digitalizzato è possibile estrarre da esso alcune caratteristiche, in inglese dette "features", che permettono di individuare delle informazioni abbastanza significative del segnale vocale che possono essere apprese da una macchina o per meglio dire da un modello di apprendimento, che sarà poi in grado di elaborarle, per cercare di categorizzare in maniera corretta un determinato file audio secondo la sfera emozionale dell'uomo. Uno degli ostacoli maggiori che riguarda la classificazione delle emozioni è la soggettività stessa delle emozioni, in quanto, per individui diversi, uno stesso audio può appartenere o meno ad una certa categoria di emozione piuttosto che ad un'altra. Inoltre, altri problemi possono essere il decidere ad esempio quali intervalli di tempo utilizzare per etichettare correttamente l'emozione oppure quante e quali emozioni bisognerebbe definire per il task di classificazione. La raccolta dei dati è anch'essa complicata. Molti audio possono essere ottenuti da film, serie tv, programmi televisivi e notiziari; tuttavia film e serie tv, sono di parte, in quanto in entrambi le emozioni sono imitate dagli attori, mentre nei notiziari la comunicazione di una notizia deve essere il più possibile neutrale, gli audio ottenuti dai programmi televisivi invece hanno bisogno comunque di essere trattati accuratamente per eliminare eventuali distorsioni del segnale dovuto al dispositivo di registrazione e all'ambiente in cui vengono raccolte. L'etichettatura dei dati richiede costi elevati sia in termini di tempo che di sforzo umano. A differenza del lavoro di etichettatura fatto su un'immagine, infatti, nel caso di un audio è necessario che questo venga ascoltato per intero, analizzato ed etichettato da più individui competenti a causa della sua soggettività detta prima. Un altro aspetto non di secondo piano è che nella realtà difficilmente si hanno emozioni pure, ben più spesso svariati stati emotivi sfumati, la cui codifica presenta difficoltà ancora maggiori di quante già ne comporti la codifica delle emozioni di base. Un database ideale

dovrebbe contemplare entrambe le tipologie di stati emotivi, il che renderebbe la sua portata di gran lunga più estesa dei database e dei dataset attualmente esistenti. Altro elemento qualificante di un database è costituito dal fatto di essere naturale o meno. Si apre qui una questione dibattutissima sui pro e sui contro riguardo al materiale raccolto naturalmente rispetto al materiale ottenuto artificialmente. Nel primo caso, ovviamente, i vantaggi sono dati da una maggiore rappresentatività, da parte del materiale analizzato, della reale espressione delle emozioni (ma si tenga presente quanto detto poc'anzi sulla scarsa frequenza di emozioni pure nella vita di tutti i giorni). Gli svantaggi sono di ordine pratico, deontologico e teorico: registrare del parlato emotivo in situazioni reali comporta inevitabilmente che il suono non sia pulito e che possa difficilmente essere confrontato con altro materiale raccolto in modo analogo. Inoltre, la registrazione di parlato emotivo dal vero, comporta spinose questioni di copyright, e conseguentemente di accessibilità ai dati da parte di altri studiosi.

In questo studio di tesi sono stati adoperati vari dataset per la classificazione di 7 diverse emozioni (disgusto, gioia, neutro, paura, rabbia, sorpresa, tristezza) secondo la valence e l'arousal, le quali verranno presentate nel capitolo 1. Il task di classificazione è stato condotto addestrando prima 5 classificatori diversi (**Random Forrest, Gradient Boosting, Support vector machine, K-nearest neighbors e la rete Multi-Layer Perceptron (MLP)**) e successivamente gli stessi test sono stati eseguiti su una **CNN (Convolutional Neural Network)**, addestrata invece sul riconoscimento degli spettrogrammi, ricavati dai file audio e categorizzati sempre secondo valence e arousal. Infine, i risultati ottenuti sono stati confrontati tra loro per decidere quale fosse il modello migliore da utilizzare nell'applicazione realizzata per Windows, chiamata EmoFEATURES, che verrà presentata nel capitolo 8.

CAPITOLO 1 - L'emozione

L'emozione è una caratteristica fondamentale, nonché tra le più distintive, dell'essere umano e in generale degli esseri viventi più senzienti. È lo stato mentale e fisiologico associato al sistema nervoso a seguito dei cambiamenti chimici indotti internamente da pensieri, sensazioni interpretate come piacevoli o meno ed esternamente da cambiamenti dell'ambiente circostante. La parola emozione è nata nel 1579, adattata dal francese *émouvoir* ("scatenare", "risvegliare"), come termine generale per intendere passione, sentimento e affezione. Solo nei primi dell'800 si è iniziato ad intendere con quella parola il moderno concetto di emozione. Dal punto di vista medico è stato verificato che l'emozione è generata in parte dal sistema limbico del cervello (composto dall'ipotalamo, corteccia cingolata e ippocampo). Tale sistema è inoltre collegato con la corteccia prefrontale, sede responsabile della pianificazione dei comportamenti cognitivi complessi (ad esempio: la presa di una decisione), della personalità e del meccanismo di moderazione nella condotta sociale. Le analisi effettuate mediante l'induzione di stimoli visivi positivi e negativi hanno dimostrato che gli stimoli ricevuti dalla corteccia prefrontale sinistra provocano l'emergere di emozioni. Per quanto riguarda la categorizzazione dell'emozione, non esiste un approccio unico riconosciuto globalmente dalla comunità scientifica, ed infatti questo rappresenta ancora un problema aperto. Al momento, gli approcci accettati e maggiormente usati per descrivere e categorizzare l'emozione umana sono principalmente due: quello categorico/discreto e continuo. Di seguito si riportano le caratteristiche di alcuni modelli di entrambe le categorie e ciò che ha portato alla loro formulazione.

1.1 Il modello discreto delle emozioni

Charles Darwin nel 1872 scrisse il libro “The Expression of the Emotions in Man and Animals” nel quale ipotizzava la capacità delle persone, indipendentemente dalla cultura e società, di esprimere le emozioni allo stesso modo. Affermò inoltre che, poiché l’essere umano condivide lo stesso passato evolutivo di taluni mammiferi, è possibile ipotizzare che altri mammiferi a noi affini presentino somiglianze nel provare ed esternare l’emozione. Fino al 1950 questa teoria era considerata superata e l’idea preponderante era che diverse persone di culture diverse avessero modi comportamentali differenti, inclusa la sfera emozionale.

L’emozione risulta essere difficile da gestire, perché può avere diverse sfaccettature. L’esperienza umana è così complessa e variegata che è tale da rendere difficile delineare un confine tra la fine di una emozione e l’inizio di un’altra. Ad esempio: emozioni quali la felicità e la rabbia sappiamo per esperienza quotidiana possedere diversi livelli di intensità, e in talune circostanze è possibile anche provare più di una emozione contemporaneamente. Gli psicologi hanno cercato comunque di creare un modello di classificazione ritenuto valido per delineare le emozioni nella loro essenza base e poter estendere il campo di ricerca su più sfaccettature della stessa emozione.

1.2 La teoria di Ekman

A partire dagli anni ’70, sono stati effettuati studi molto importanti riguardo la classificazione dell’emozione i quali hanno ribaltato le teorie precedenti.

Tra le ricerche più influenti si annoverano quelli dello psicologo Paul Ekman, il quale ha avallato l’idea che le emozioni sono categorie discrete, misurabili e fisiologicamente distinte. Ha stilato nel 1972 una lista delle emozioni base dopo aver condotto sperimentazioni sulle espressioni facciali.

L’esperimento da lui ideato consisteva nel descrivere una situazione ad una persona (ad esempio: una storia o un atto effettuato da qualcuno) e chiedeva successivamente di scegliere quale tra 3 foto di volti è quella che raffigura l’espressione facciale più adatta alla situazione descritta. Successivamente chiedeva ad altri soggetti di etichettare l’emozione che traspariva dai volti selezionati. L’esperimento è stato somministrato sia a

persone letterate dotate di una certa cultura che illetterati (nello specifico: una popolazione totalmente isolata in Nuova Guinea, coadiuvato dal collega Wallace Friesen).

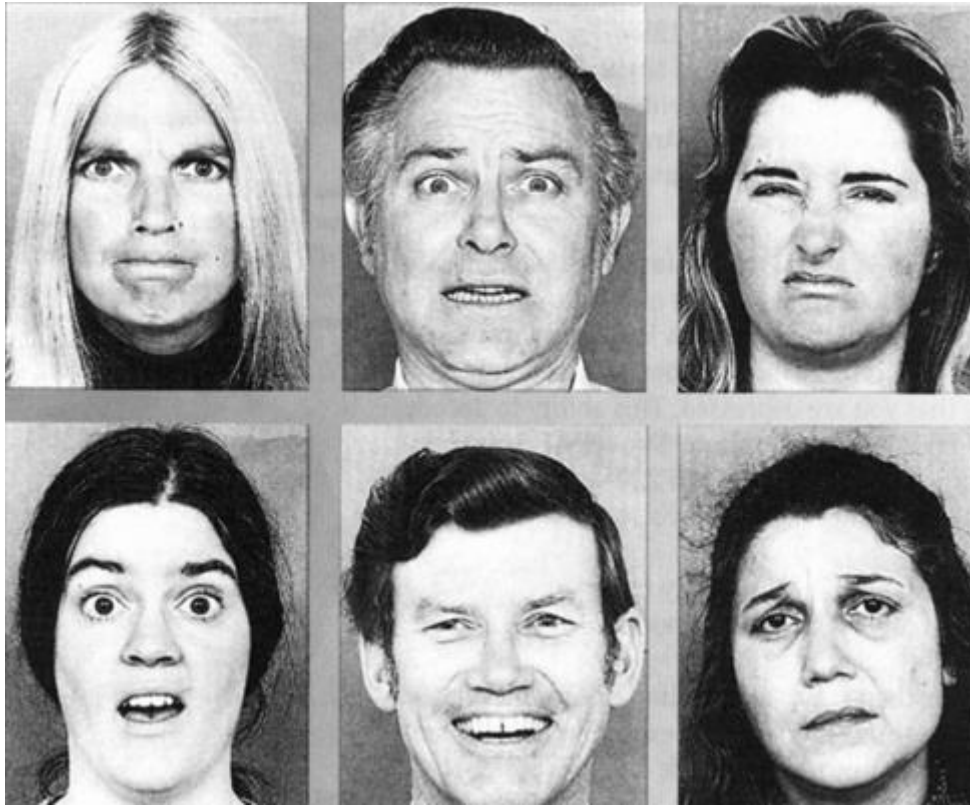


Figura 1: Immagine rappresentante alcune espressioni facciali. [19]

L'aspetto più interessante emerso dai risultati di questa ricerca è che il 93% delle persone illetterate sceglievano le stesse espressioni del volto indicate da quelli letterati, segno che le espressioni emotive non dipendono da una particolare cultura, ma sono invece universali per tutti. Questo ha riqualificato la teoria di Darwin, formulata 100 anni prima. È stato quindi possibile identificare 6 emozioni base che possono rappresentare universalmente al meglio le espressioni (facciali e non):

- Rabbia;
- Disgusto;
- Paura;
- Felicità;
- Tristezza;
- Sorpresa.

A questa lista, Ekman ha aggiunto nel 1999 altre emozioni (alcune non riscontrabili nelle espressioni facciali):

- Divertimento;
- Disprezzo;
- Soddissfazione;
- Imbarazzo;
- Eccitamento;
- Colpa;
- Orgoglio;
- Solievo;
- Soddissfazione;
- Piacere sensoriale;
- Vergogna.

1.3 Il modello dimensionale delle emozioni

Seppure nella comunità scientifica il modello discreto sia ritenuto valido, non tutti gli psicologi sono concordi nell'affermare che l'emozione può essere interpretata solamente secondo un sistema categoriale. Proprio perché spesso stesse emozioni possono avere forme e intensità diverse, utilizzare un modello discreto potrebbe non essere sufficiente a delineare le diverse sfaccettature della stessa emozione.

Emozioni diverse possono avere un unico comune denominatore e condividere aspetti simili ma non tali da poterli unire sotto un'unica etichetta.

1.4 Il modello circonflesso delle emozioni di Russell

Tra i più forti critici del modello discreto delle emozioni figura lo psicologo americano James A. Russell, il quale ideò nel 1980 il modello circonflesso dell'emozione.

Egli suppose di delineare una circonferenza con al centro un sistema di assi cartesiani. Sull'asse delle ascisse è posta la valenza, che può essere positiva o negativa e su quello delle ordinate l'arousal con valori quali alta e bassa. L'idea posta da Russell è che qualunque emozione è possibile indicarla all'interno di questa circonferenza e ricavarla in termini di valenza ed arousal. Il centro della circonferenza rappresenta il neutro, ossia emozioni con valori di valenza ed arousal medi.

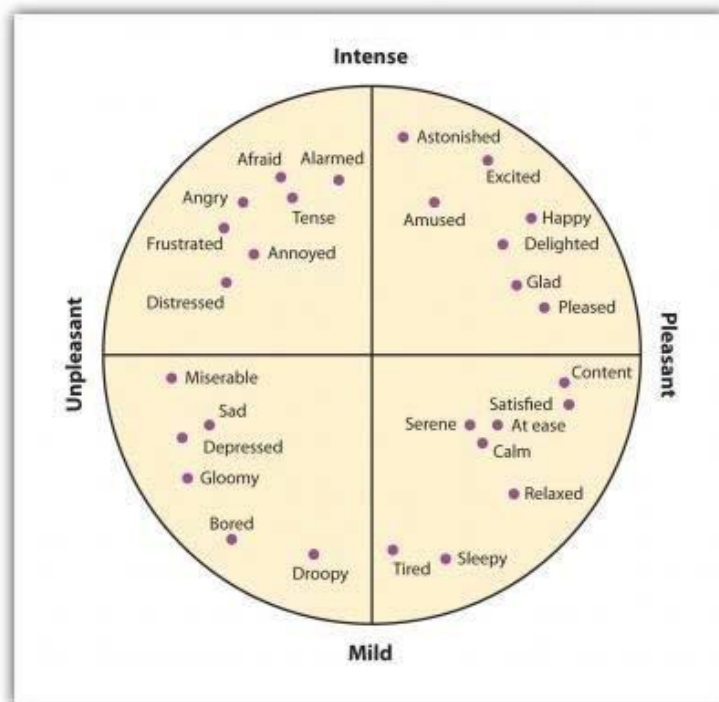


Figura 1.1: Le emozioni secondarie. Le emozioni secondarie sono quelle che hanno una maggiore componente cognitiva. Sono determinati sia dal loro livello di eccitazione (da lieve a intenso) sia dalla loro valenza (da piacevole a spiacevole). [20]

Con la valenza si intende il grado di “piacevolezza” o di “negatività” di una emozione. Emozioni positive sono ad esempio la gioia, entusiasmo, calma; viceversa emozioni negative sono la depressione, rabbia, stress. L'arousal è il grado di “eccitamento” o “risveglio” dell'emozione o anche più in generale di una persona. Avere un alto livello di arousal significa provare forti stimoli e un alto grado di eccitazione, viceversa un basso

livello indica uno stato di quiete e/o di bassa attenzione. Esempi di emozioni con arousal alta sono tutte quelle con un alto carico di eccitamento; quali la gioia, rabbia, nervosismo. Viceversa, emozioni con arousal bassa sono la tristezza, il disgusto, il rilassamento.

Queste sono solo alcune delle emozioni che è possibile rivedere in termini di valenza ed arousal, ma mediante questa metodologia ogni emozione e qualunque sua sfaccettatura è possibile individuarla all'interno della circonferenza con precisi valori di valenza ed arousal. È possibile anche così definire quali sono le emozioni con pari valore di arousal o di valenza, studiando le eventuali corrispondenze che esistono tra esse.

Successivamente a questo modello, Russell, ideò assieme al collega Albert Mehrabian una sua variante, il modello PAD (sigla di pleasure, arousal, dominance) che si contraddistingue dal primo per la presenza di una dimensione in più, la dominance, che indica il grado che ha il soggetto nel controllare o meno l'emozione. Questo modello è principalmente usato per interpretare il linguaggio non verbale del corpo umano.

1.5 Il modello di Plutchik

Un altro studio delle emozioni secondo l'aspetto dimensionale fu quello di Robert Plutchik che nel 1980 propose un modello in cui è possibile distinguere 8 emozioni base suddivise in 4 coppie antagoniste e che da esse è possibile descrivere tutte le altre emozioni e i collegamenti esistenti.

Le 4 coppie sono:

- Gioia contro dolore;
- Rabbia contro paura;
- Accettazione contro disgusto;
- Sorpresa contro attesa.

Questo modello prende il nome di *ruota delle emozioni* o se visto tridimensionalmente *cono delle emozioni*.

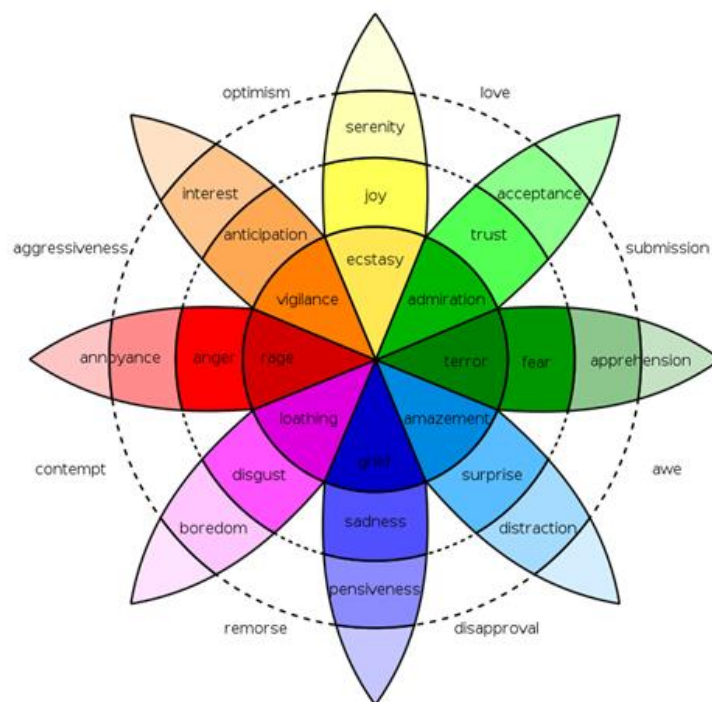


Figura 1.2: ruota delle emozioni.[21]

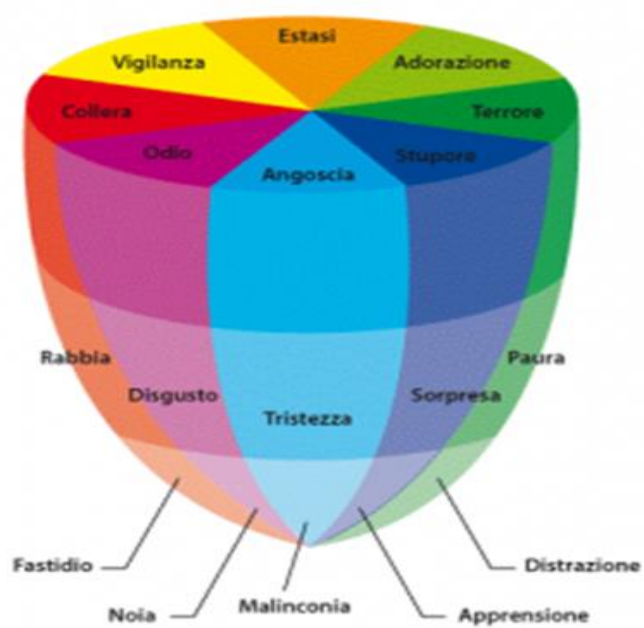


Figura 1.3: Cono delle emozioni.[22]

L'idea alla base di questo modello è che esistono emozioni basilari e centrali (le 8 precedentemente citate) e che “miscele” di emozioni attigue permettono di determinarne altre. Le emozioni attigue sono quindi quelle più somiglianti tra loro, mentre quelle agli antipodi (corrispondenti alle 4 coppie antagoniste) sono quelle più diverse tra loro.

Le emozioni più lontane dal centro del cerchio (o quelle più vicine la punta nel modello conico) rappresentano le emozioni base ma con valori man mano sempre più bassi di intensità (arousal).

1.6 L'affective computing

Prima di definire cosa è l'affective computing, è bene discutere di come i sistemi software più moderni e di uso comune si interfacciano con i loro utilizzatori.

Dispositivi informatici e sistemi software, a seconda delle loro funzioni e finalità d'uso, possono presentare diverse tipologie di interfacce e di interazione con l'utente. Oltre a quelle più conosciute e che comunemente si usano con i programmi per computer, in cui l'utente si interfaccia mediante l'utilizzo di mouse e tastiera ed interfaccia grafica su schermo, o con app per smartphone e tablet dove si effettuano tocchi e movimenti sul display touch; un altro esempio è dato dagli assistenti virtuali quali Amazon Alexa, Google Home, Microsoft Cortana ed Apple Siri. Questi sistemi adoperano principalmente una interfaccia vocale usando tecniche di speech recognition (riconoscimento del parlato) e sono stati pensati per assistere gli utenti nell'eseguire svariati compiti ed ottenere rapidamente informazioni reperibili in Internet o dai propri servizi online. Alcuni possono essere eseguiti sia da computer (Cortana e Siri, ad esempio) che da smartphone e sono anche installati sotto dispositivi creati ad hoc con i quali è possibile effettuare interrogazioni e impartire comandi.

Assistenti virtuali di questo tipo stanno sempre più prendendo piede nella vita delle persone. L'integrazione con servizi già ampiamente usati come quelli offerti da colossi dell'informatica quali Amazon o Google, associati a sistemi domotici casalinghi anche molto economici e di facile installazione quali luci e prese elettriche smart, le capacità di eseguire rapidamente brevi ricerche e piccoli task (impostare un timer, un promemoria, eccetera) e altre funzioni accessibili con il semplice utilizzo della voce, sono considerati un fattore importante ed imprescindibile nella vita d'ogni giorno da parte degli utilizzatori più assidui.

Oltre che migliorare ulteriormente ciò che è già stato progettato, gli sviluppatori si stanno focalizzando per integrare in questi sistemi un modello di riconoscimento dell'emozione dell'utente e di empatia. In Amazon sono infatti al lavoro proprio questo, sviluppando con l'intelligenza artificiale un riconoscitore dello stato emotivo mediante l'analisi della voce.

La branca dell'informatica che si occupa del riconoscimento, interpretazione e simulazione dell'affettività umana da parte di un software è definita *affective computing*.

Questo è un campo che presenta molti aspetti interdisciplinari, spaziando in primis dalla psicologia fino alla scienza cognitiva, al machine learning e all'interazione uomo-macchina ed è tenuta in considerazione anche in ambiti quali l'ambient assisted living (AAL). La definizione e l'approccio moderno all'affective computing è dato a seguito delle ricerche di Rosalind W. Picard.

Nel suo libro "*Affective Computing*" afferma:

Sono giunta alla conclusione che se vogliamo i computer essere genuinamente intelligenti, per adattarsi e interagire naturalmente con noi, allora hanno bisogno dell'abilità di riconoscere ed esprimere emozioni, provare emozioni, e avere quella che è stata definita "intelligenza emotiva".

Come è possibile intuire, il fine dell'affective computing è quindi quello di adattare il modello comportamentale del sistema in base all'umore dell'utente, esternando essi stessi emozioni, al fine di indurre un certo livello di empatia tra l'essere umano con la macchina. Questo è un aspetto da non tralasciare, soprattutto nell'ottica della creazione di sistemi

inerenti al lato affettivo e sociale; quali assistenti robotici per la già citata AAL e l'assistenza a soggetti quali persone anziane o diversamente abili. Studi di questo tipo rappresentano un sostanziale punto di rottura con la precedente visione dell'informatica, perché quello che si richiede ai programmatori è di tenere in considerazione il lato emotivo dell'utente durante la progettazione di un software.

C'è quindi bisogno di un approccio con l'utente quanto più user-friendly e naturale possibile, affinché sistemi e dispositivi basati sull'affective computing siano funzionali e che possano davvero essere utili nella vita degli esseri umani.

Picard negli anni 90 ipotizzò che in futuro i “new wearable computers” (ossia i “nuovi computer indossabili”) avrebbero apportato un grosso contributo al campo dell'affective computing.

Allo stato attuale, per l'acquisizione dei dati per il successivo riconoscimento dello stato emotivo, si possono applicare diversi approcci.

Il primo è basato sull'applicazione fisica di dispositivi e sensori sul soggetto da analizzare, quali elettrodi per l'elettroencefalogramma o casco neurale; misuratori della pressione sanguigna e battito cardiaco; temperatura corporea e resistenza galvanica della pelle; eccetera.

Tale metodologia, si può ben intuire, non è sempre applicabile. Strumentazione del genere è spesso costosa, richiede in taluni casi l'impiego di personale medico e a seconda della situazione non è sempre possibile collegare tali dispositivi ai soggetti, ma è la tecnica che maggiormente può offrire dati oggettivi e certi per la ricerca dell'emozione. È applicabile in caso di conduzione di esperimenti in ambiente controllato per fini di ricerca.

Il secondo metodo prevede invece l'uso di dispositivi non a contatto con il soggetto, quali telecamere per riprendere le espressioni facciali; posture e movimenti del corpo; gesti e microfoni per rilevare la voce.

Si vuole citare anche un'altra fonte di acquisizione dello stato emotivo di una persona: tramite il testo scritto o il parlato trascritto mediante speech-to-text. Difatti, mediante la tecnica della sentiment analysis, si può intuire quello che è lo stato (la polarizzazione, se

positiva o negativa o neutra) del soggetto. Il vantaggio dato dal predire l'emozione (o il sentimento) mediante questa tecnica è dato anche dalla possibilità di usare database lessicali con i quali è possibile costruire modelli di riconoscimento molto validi. Giusto per citarne alcuni liberalmente accessibili: WordNet, ConceptNet e BabelNet.

L'acquisizione e l'analisi dei dati da più fonti, di solito, restituisce una maggiore precisione del riconoscimento. In questa tesi ci si è concentrati sul riconoscimento dello stato emotivo dalla voce: lo *speech emotion recognition*.

1.7 Speech emotion recognition

La parola è uno dei modi più naturali che ha l'essere umano per comunicare con gli altri, ed anche per esprimere le proprie emozioni. Da tempo il riconoscimento del parlato, ossia lo *speech recognition*, è disponibile per tutti tramite diverse piattaforme, anche gratuite, che offrono il servizio di *speech-to-text* in real-time.

Tra i servizi di *text-to-speech* disponibili per gli sviluppatori ci sono ad esempio quello di Google, sulla piattaforma Google Cloud, accessibile mediante API (Application Programming Interface) ed Amazon Polly, offerto da Amazon, disponibile su Amazon Web Services (AWS).

Con il parlato, oltre che la semantica delle frasi, gli altri possono avvertire con una certa facilità anche l'emozione della persona.

Si definisce *speech emotion recognition* (SER) la capacità di riconoscere lo stato emotivo di una persona dalla voce. Al momento, i servizi di riconoscimento dell'emozione dalla voce sono molto limitati. Infatti, rispetto lo *speech* e *text recognition*, allo stato attuale non si hanno a disposizione modelli di riconoscimento dell'emozione dal parlato efficaci come quelli per lo *speech recognition*. Questo perché, spesso le emozioni sono soggettive e non possono essere misurate e categorizzate oggettivamente.

Lo stato dell'arte fino a qualche anno fa prevedeva principalmente l'impiego di uno o più dataset (corpus) di voci etichettate, dalle quali estrarre le caratteristiche (features) che

meglio evidenziano le differenze delle emozioni. Tuttavia negli ultimi anni, grazie anche all'evoluzione tecnologica e soprattutto hardware, una nuova strategia di riconoscimento delle emozioni si sta sviluppando. Essa consiste nell'utilizzare le **Convolutional Neural Network (CNN)** utilizzando come input, non più le features classiche (definite nel capitolo 3), ma bensì gli spettrogrammi, i quali verranno discussi in una sezione apposita (6.9) successivamente. Si è infatti notato che grazie alla loro struttura, gli spettrogrammi riescono in alcuni casi a rappresentare meglio le discrepanze in termini temporali, di frequenza e intensità di energia delle diverse emozioni.

1.8 Analisi e classificazione del parlato

Poiché l'arousal e la valence sono le dimensioni fondamentali per poter individuare quale tipo di emozione prova l'utente, è possibile trovare una corrispondenza tra una loro combinazione e le emozioni umane. Per poter classificare l'arousal e la valence del parlato emotivo è necessario analizzarlo e classificarlo, individuando un insieme discreto di classi. In base alle precedenti considerazioni, le possibili classi per l'arousal sono le seguenti:

- **High**
- **Medium**
- **Low**

Per la valence, invece, è possibile stabilire le seguenti classi:

- **Positive**
- **Neutral**
- **Negative**

Emozione	Arousal	Valence
Disgusto	bassa	negativa
Gioia	alta	positiva
Neutrale	media	neutrale
Paura	alta	negativa
Rabbia	alta	negativa
Sorpresa	alta	positiva
Tristezza	media	negativa

*Tabella 1: Corrispondenza tra i valori di **valence** ed **arousal** e le emozioni di base.*

La corrispondenza dei valori di **arousal**, nei dataset che sono stati creati e serializzati, è la seguente:

- **High** -----> 1
- **Medium** -----> 2
- **Low** -----> 3

La corrispondenza dei valori di **valence** nei dataset creati è la seguente:

- **Positive** -----> 1
- **Neutral** -----> 2
- **Negative** -----> 3

Per quanto riguarda la rete neurale invece, poiché è stata utilizzata la libreria Keras, quest'ultima richiede che il valore della prima etichetta numerica sia 0, per queste ragioni la corrispondenza che è stata adottata nei dataset per la CNN è la seguente:

Per l'**arousal**:

- **High** -----> 0
- **Medium** -----> 1
- **Low** -----> 2

Per la **valence**:

- **Positive** -----> **0**
- **Neutral** -----> **1**
- **Negative** -----> **2**

Come si può notare dalla tabella 1, le etichette per la valence e arousal per gioia/sorpresa e paura/rabbia sono uguali, per questo motivo, per gestire questi due casi, sono stati testati i 5 classificatori che verranno presentati nel capitolo 4 con la **stratified k-fold cross-validation** che verrà presentata, invece, all'inizio del capitolo 5. Qui sotto vengono presentate le tabelle riassuntive con i valori di accuratezza per ogni classificatore.

CLAS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.93%	0.91%	0.93%	0.91%	0.92%	0.80%
GB	0.94%	0.91%	0.94%	0.92%	0.92%	0.93%
SVM	0.97%	0.95%	0.98%	0.95%	0.97%	0.97%
MLP	0.71%	0.69%	0.72%	0.73%	0.68%	0.71%
KNN	0.88%	0.88%	0.91%	0.88%	0.89%	0.89%

Tabella 1.1: Risultati per classificare gli audio secondo gioia o sorpresa.

CLAS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.93%	0.93%	0.92%	0.93%	0.94%	0.93%
GB	0.94%	0.95%	0.93%	0.92%	0.95%	0.94%
SVM	0.98%	0.96%	0.97%	0.97%	0.98%	0.97%
MLP	0.73%	0.74%	0.76%	0.71%	0.72%	0.73%
KNN	0.93%	0.92%	0.93%	0.91%	0.94%	0.93%

Tabella 1.2: Risultati per classificare gli audio secondo paura o rabbia.

Osservando le tabelle 1.1 e 1.2, si capisce come la SVM (Support Vector Machine) sia il classificatore migliore in entrambe le situazioni (con lo 0.97% di accuratezza media), per queste ragioni, è stata utilizzata per gestire questi due casi.

CAPITOLO 2 - Dataset

2.1 Dataset Ravdess (Ryerson Audio-Visual Database of Emotional Speech and Song)

Il dataset “RAVDESS” (Ryerson Audio-Visual Database of Emotional Speech and Song) è stato utilizzato per addestrare e testare i classificatori. RAVDESS contiene 7356 file. Ogni file è stato valutato 10 volte sulla validità emotiva, intensità e genuinità. Le valutazioni sono state fornite da 247 persone provenienti dal Nord America. Un'ulteriore serie di 72 partecipanti ha fornito dati di test-retest. Il set di dati contiene il set completo di 7356 file RAVDESS (dimensione totale: 24,8 GB) che sono così distribuiti:

- **Audio-only:**

Voce (1440 file)

Canzoni (1012 file)

- **Audio-Visual & Video-only:**

Video e Voce insieme (2880 file)

Video e Canzoni insieme (2024 file)

In questa tesi è stata utilizzata solamente la porzione “**Audio-only**” contenente file audio vocali. Questa porzione del dataset contiene 1440 file, tra cui: **60 registrazioni per attore x 24 attori** = 1440. Nello specifico, poiché in Ravdess viene considerata anche l'emozione calma, che in questo caso non ci interessa, **le registrazioni in totale saranno 1248**. Ravdess contiene 24 attori professionisti (12 di sesso femminile e 12 di sesso

maschile), che pronunciano due frasi (“Kids are talking by the door”, “Dogs are sitting by the door”) in un accento nordamericano. Le espressioni riguardano codeste emozioni: calma, neutro, gioia, tristezza, rabbia, paura, sorpresa e disgusto. Ogni espressione è prodotta a due livelli d’intensità emotiva (normale, forte). Ognuno dei 24 attori è costituito da tre formati di modalità: solo audio (16 bit, 48 kHz .wav), audio-video (720p H.264, AAC 48 kHz, .mp4) e solo video (senza audio).

EMOZIONE	NUMERO AUDIO
Rabbia	192
Disgusto	192
Paura	192
Gioia	192
Tristezza	188
Neutrale	96
Sorpresa	192

Tabella 2: Rappresentazione della distribuzione degli audio in Ravdess dopo aver eliminato tutti gli audio che avevano una durata al di sotto di 1 secondo.

2.2 Dataset Tess (Toronto emotional speech set)

Il dataset è composto da 2800 frasi totali di cui 1400 pronunciate dalla prima attrice di 26 anni e le restanti 1400 pronunciate da un'altra attrice di 64 anni, entrambe parlano inglese come prima lingua e sono state reclutate nell'area di Toronto. Ciascuna attrice ha pronunciato la frase “Say the word” 200 volte per ognuna delle 7 emozioni (rabbia, felicità, tristezza, disgusto, paura, piacevole sorpresa e neutralità). Il set di dati è organizzato in modo tale che le registrazioni di ogni emozione per le due donne siano riportate in varie cartelle differenti (14 in totale). Tutte le registrazioni sono in formato WAV. La cosa interessante è che questo set di dati è solo femminile ed è di qualità audio molto elevata. La maggior parte degli altri set di dati, in genere, possiedono una quantità maggiore di audio per gli oratori maschili e quindi questo provoca una rappresentazione leggermente squilibrata. Quindi grazie a questo dataset si riesce ad avere un ottimo set di

dati di allenamento per il classificatore delle emozioni in termini di generalizzazione (senza eccesso di adattamento).

EMOZIONE	NUMERO AUDIO
Rabbia	400
Disgusto	400
Paura	393
Gioia	400
Tristezza	387
Neutrale	390
Sorpresa	400

Tabella 2.1: Rappresentazione della distribuzione degli audio in Tess dopo aver eliminato tutti gli audio che avevano una durata al di sotto di 1 secondo.

2.3 Dataset Savee (Surrey Audio-Visual Expressed Emotion)

Questo dataset è composto da 480 audio totali registrati da 4 oratori maschili, per ogni emozione sono state registrate 15 frasi da ciascuno di essi fatta eccezione per la neutralità, per la quale sono state eseguite 30 registrazioni da parte di ognuno. Ogni audio è in formato WAV, monocanale con una frequenza di 44100 Hz.

EMOZIONE	NUMERO AUDIO
Rabbia	58
Disgusto	57
Paura	59
Gioia	59
Tristezza	55

Neutrale	111
Sorpresa	59

Tabella 2.2: Rappresentazione della distribuzione degli audio in Savee dopo aver eliminato tutti gli audio che avevano una durata al di sotto di 1 secondo.

2.4 Dataset Emovo

Un ulteriore dataset utilizzato in questo lavoro, è Emovo, il quale è completamente in lingua italiana. EMOVO è il primo database di parlato emotivo per la lingua italiana, ed è stato realizzato per conto del Dipartimento di Studi Glottoantropologici e Discipline musicali dell'Università di Roma "La Sapienza" e della Fondazione Ugo Bordoni. Sono stati convocati 6 attori, 3 maschi e 3 femmine, di provata professionalità e si sono fatte loro recitare 14 frasi (assertive, interrogative, elenchi) in base ai 6 stati emotivi di base (disgusto, paura, rabbia, gioia, sorpresa, tristezza) più lo stato neutro. Tali emozioni sono le ben note "big six" riscontrabili in molta della letteratura inerente al parlato emotivo, nonché quelle prese in considerazione dal CNR di Padova, che ha condotto i principali studi sul parlato emotivo italiano. Le registrazioni sono state effettuate, con adeguate strumentazioni professionali, presso i laboratori della Fondazione Ugo Bordoni, dove è stata garantita agli attori la possibilità di autoindursi lo stato emotivo desiderato in base alle proprie tecniche professionali. Si è optato per una frequenza di campionamento di 48 kHz, 16 bit stereo, formato wav (stesse caratteristiche degli audio di Ravdess). Ogni singola elicitazione delle frasi è stata etichettata ed archiviata in un database di tipo Access, in modo da consentirne una rapida individuazione. In totale sono stati archiviati 588 record: 6 attori X 14 frasi X 7 stati. EMOVO è stato validato con un test di discriminazione delle emozioni su due frasi ('la casa forte vuole col pane', 'il gatto sta scorrendo nella pera'), condotto, parallelamente e separatamente, dall'autore e dall'Università della Calabria. Entrambi i test hanno avuto un campione di 12 soggetti riconoscitori, i quali hanno dovuto di volta in volta indicare, scegliendo fra due possibili risposte, lo stato emotivo delle frasi ascoltate. Il test ha avuto esito positivo in quanto è

risultata un'accuratezza complessiva dei riconoscimenti pari all'80%, senza significative discrepanze fra i risultati dei due test. È stata valutata inoltre la scala di riconoscibilità delle emozioni, e si è riscontrata una sostanziale concordanza con la scala di riconoscibilità desunta dalla letteratura al riguardo.

EMOZIONE	NUMERO AUDIO
Rabbia	83
Disgusto	84
Paura	84
Gioia	84
Tristezza	83
Neutrale	84
Sorpresa	84

Tabella 2.3: Rappresentazione della distribuzione degli audio in Emovo dopo aver eliminato tutti gli audio che avevano una durata al di sotto di 1 secondo.

2.5 Dataset EmoFilm

EmoFilm è un dataset di discorsi emotivi multilingue che comprende 1115 istanze audio prodotte in inglese, italiano e spagnolo. Le clip audio (con una lunghezza media di 3,5 secondi e 1,2 secondi standard) sono state estratte in formato WAV (monocanale, frequenza di campionamento di 48 kHz) da 43 film (originariamente in inglese e poi tradotte nelle versioni di italiano e spagnolo). Sono stati considerati generi come commedia, dramma, horror e thriller per estrarre frasi contenenti 5 stati emotivi diversi: la rabbia, il disgusto, la felicità, la paura e la tristezza. EmoFilm è stato presentato a Interspeech 2018. La versione finale di EmoFilm è composta da 1115 clip con una lunghezza media di 3,5 sec. con **360 clip in lingua inglese** (182 prodotti da femmine), con una media di 34.3 espressioni per emozione; **413 clip audio in lingua italiana** (190

per femmine), con una media di 41.3 espressioni per emozione; **342 clip audio in lingua spagnola** (165 per femmine), con una media di 35,9 espressioni per emozione.

EMOZIONE	NUMERO AUDIO
Rabbia	209
Disgusto	156
Paura	190
Gioia	195
Tristezza	200

Tabella 2.4: Rappresentazione della distribuzione degli audio in Emofilm dopo aver eliminato tutti gli audio che avevano una durata al di sotto di 1 secondo.

2.6 Dataset Emodb (Berlin Database of Emotional Speech)

Il set di dati EMODB contiene 494 espressioni registrate in una camera anecoica da 10 attori professionisti (5 maschi / 5 femmine). Ogni attore ha simulato 10 frasi in sette diverse emozioni (neutro, rabbia, paura, gioia, tristezza, disgusto e noia). Sono state registrate in totale 800 frasi. Effettuando un test di percezione relativo alla riconoscibilità delle emozioni e alla loro naturalezza, sono state scelte tutte le espressioni con una riconoscibilità superiore all'80% e una naturalezza superiore al 60% come campioni finali del set di dati. Poiché in questo lavoro di tesi si concentra sulle 7 emozioni principali allora la noia è stata eliminata, ottenendo così un dataset formato da 454 audio.

EMOZIONE	NUMERO AUDIO
Rabbia	127
Disgusto	46
Paura	69
Gioia	71

Tristezza	62
Neutrale	79

Tabella 2.5: Rappresentazione della distribuzione degli audio in Emodb dopo aver eliminato tutti gli audio che avevano una durata al di sotto di 1 secondo.

2.7 Dataset Demos (Database of Elicited Mood in Speech)

Demos è un altro dataset in lingua italiana più grande. Sono stati registrati 68 soggetti (23 femmine, 45 maschi). Il dataset comprende 9365 campioni di sette stati emotivi (rabbia, paura, tristezza, felicità, disgusto, sorpresa e colpevolezza) e 332 campioni neutrali. Poiché la colpevolezza non è d'interesse per questo lavoro di tesi è stata tolta, per cui gli audio totali disponibili sono 8568. La distribuzione degli audio è rappresentata nella tabella 2.6. Le registrazioni sono state effettuate in formato WAV a singolo canale e frequenza di campionamento 48 kHz. Successivamente, il discorso emotivo è stato segmentato manualmente in campioni (lunghezza media 2,9 s, std 1,1 s).

EMOZIONE	NUMERO AUDIO
Rabbia	1433
Disgusto	1669
Paura	1150
Gioia	1362
Tristezza	1518
Neutrale	332
Sorpresa	1000

Tabella 2.6: Rappresentazione della distribuzione degli audio in Demos dopo aver eliminato tutti gli audio che avevano una durata al di sotto di 1 secondo.

EMOZIONE	NUMERO AUDIO
Sorpresa	2001
Rabbia	2502
Disgusto	2604
Paura	2137
Gioia	2363
Tristezza	2493
Neutrale	1092

Tabella 2.7: Tabella riassuntiva della distribuzione degli audio (maggiori di 1 secondo).

CAPITOLO 3 - ESTRAZIONE FEATURES AUDIO

3.1 Pre-elaborazione audio

Per poter avere file audio completamente omogenei ho eseguito il campionamento di ogni registrazione di ciascun dataset ad una frequenza di 16Khz e rimosso il silenzio sia all'inizio che alla fine di ciascun audio in modo tale da non avere valori sonori inutili al fine del corretto riconoscimento di ogni emozione. Per eseguire il campionamento ho utilizzato l'applicazione **online-audio-converter**, mentre per eliminare il silenzio ho utilizzato **silrem**. Dopo l'eliminazione del silenzio alcuni file audio risultavano essere troppo corti (con durata al di sotto di 1 secondo) e quindi non permettevano ai modelli di acquisire abbastanza informazioni per eseguire la classificazione, per queste ragioni questi audio sono stati eliminati da ogni dataset.

3.2 Mfcc (Mel-frequency-Cepstral-Coefficient)

Sono features che seguono il comportamento della Coclea che è una componente dell'orecchio umano di forma a spirale, contenente al proprio interno delle ciglia che permettono di trasformare il segnale sonoro entrante in un segnale elettrico in uscita, il quale è inviato direttamente al nervo acustico e che infine arriverà al cervello. Poiché all'interno di un calcolatore gli unici segnali che possono essere gestiti sono quelli elettrici allora si è cercato di costruire un filtro in grado di simulare questa parte dell'orecchio. All'interno della Coclea il numero delle ciglia (che si comportano come dei recettori) non è distribuito uniformemente, in particolare nella parte più esterna vengono percepite le alte frequenze, mentre nella parte più interna quelle basse. Il numero di recettori sono distribuiti in maniera logaritmica nella parte esterna e lineare nella parte più interna, questo si traduce in una maggiore facilità di riconoscimento da parte dell'uomo di segnali a bassa frequenza rispetto a quelli ad alta frequenza. Per emulare quindi questo funzionamento, in ambito informatico è stato inventato un filtro, chiamato **Mel**, il quale è formato da una serie di filtri che hanno la particolarità di andare a trasformare i dati sonori (memorizzati nel calcolatore generalmente in formato vettoriale) in modo lineare alle basse frequenze (sotto i 1000Hz) e logaritmico alle alte (sopra i 1000 Hz), emulando quindi direttamente il funzionamento della Coclea spiegata inizialmente. In psicoacustica la conversione da Hz a Mel è la seguente:

$$f_{mel} = 2595 \log_{10}\left(1 + \frac{f_{hz}}{700}\right)$$

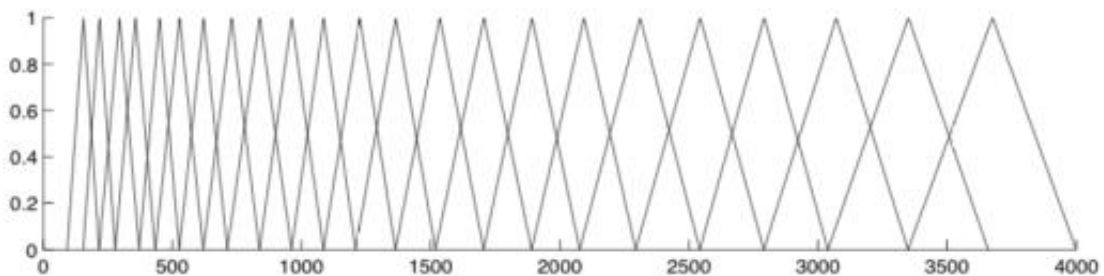


Figura 3: Filterbank utilizzato per il calcolo degli mfcc.[23]

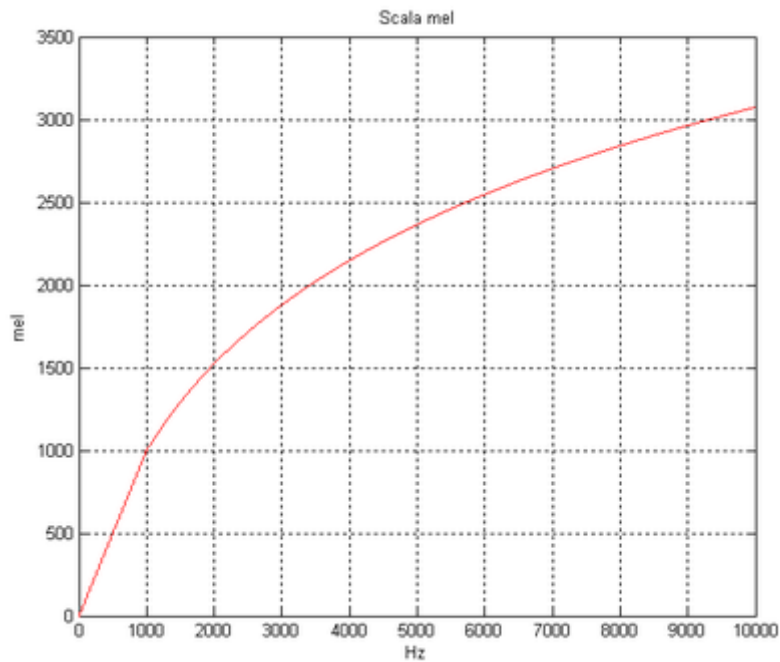


Figura 3.1: Trasformazione da Hz a Mel.[24]

Il filterbank che si può notare in Figura 3 lavora nel range di frequenze [64,4000] Hz. Il numero di filtri è 24 e sono solitamente progettati con una risposta triangolare.

Ricapitolando, dal punto di vista informatico quindi, applicare il filtro **Mel** ad un file audio che abbiamo in formato vettoriale, vuol dire semplicemente effettuare una moltiplicazione tra i valori numerici di quest'ultimo con i valori, anch'essi presenti all'interno di un vettore, del filterbank mostrato in figura 3.

Prima di determinare i passi da seguire per calcolare gli MFCC bisogna definire da cosa è composto il segnale audio che il nostro orecchio percepisce: il segnale finale che otteniamo, che chiamiamo **sn**, è composto da due parti, la prima è la sorgente di eccitazione chiamata **yn**, ovvero l'aria che attraversa l'apparato fonatorio e può mettere in vibrazione le corde vocali producendo quindi un segnale voiced, oppure non metterle in vibrazione producendo quindi un segnale in questo caso unvoiced, e la seconda è la descrizione dell'apparato fonatorio che possiamo chiamare **gn**, il quale varia da individuo a individuo. Tutto questo è rappresentato nell'immagine di sotto (3.2).

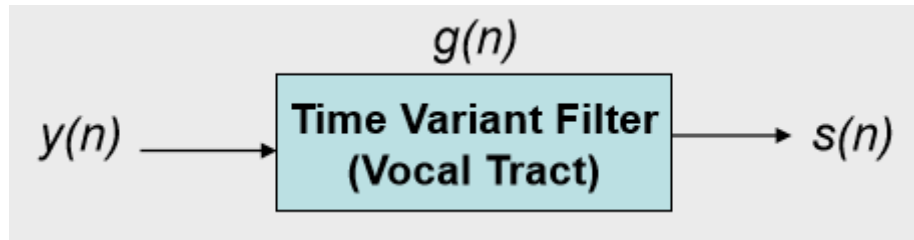


Figura 3.2: Rappresentazione della relazione che c'è tra il segnale d'ingresso composto dalla sorgente d'eccitazione ($y(n)$) e l'apparato fonatorio ($g(n)$) e il segnale audio d'uscita ($s(n)$). [25]

Seguendo la matematica si può dire che la relazione tra ingresso e uscita del sistema (3.2) la si può trovare eseguendo il prodotto di convoluzione, ovvero:

$$S(n) = y(n) * g(n)$$

A questo punto per semplificare le cose, quello che si fa è eseguire la trasformata di Fourier perché questo ci permette di trasformare l'operazione di convoluzione in una semplice moltiplicazione tra $y(n)$ e $g(n)$, in maniera formale:

- $Y(\omega) = F(y(n))$
- $G(\omega) = F(g(n))$
- $S(\omega) = Y(\omega) G(\omega)$ (prodotto normale)

Successivamente viene applicato il logaritmo ad entrambi i membri dell'ultima equazione in modo tale da poter applicare la proprietà che permette di separare le due componenti $Y(\omega)$ e $G(\omega)$, ottenendo questo:

$$\log(S(\omega)) = \log(Y(\omega)) + \log(G(\omega))$$

Occorre considerare però solo il modulo degli argomenti dei logaritmi, essendo essi definiti solo per numeri positivi:

$$\log(|S(\omega)|) = \log(|Y(\omega)|) + \log(|G(\omega)|)$$

Arrivati a questo punto quindi, le due componenti iniziali sono state separate, quello di cui ora si ha bisogno però, è averle nel dominio del tempo, per cui si esegue la trasformata inversa di Fourier ottenendo così:

$$Cp(n) = \text{IFFT}(\log(|S(\omega)|))$$

Il dominio in cui siamo arrivati ora non è quello del tempo ma è chiamato Cepstrum e tutti i passaggi eseguiti fino ad ora sono quelli fatti per calcolare la cosiddetta trasformata Cepstrum.

Vediamo ora come vengono calcolati gli MFCC:

- Viene effettuata la registrazione tramite un dispositivo hardware, il segnale registrato viene diviso in frame di 20-30ms con sovrapposizione di 5-15ms. Questo viene fatto perché in questo modo si può assumere che l'audio non vari in un intervallo così piccolo, e che il segnale sia quindi stazionario per la breve durata del frame, in modo tale da poter applicare correttamente la trasformata di Fourier.
- Di ogni frame viene eseguita la pre-enfasi in modo tale da diminuire la differenza che c'è tra le alte e le basse frequenze aumentando la potenza del segnale perché non è detto che per un determinato suono l'orecchio umano sia lo strumento migliore per poterlo riconoscere, poiché magari alle alte frequenze il segnale audio contiene delle informazioni molto utili che un calcolatore potrebbe utilizzare.
- Viene applicata una finestra di Hamming, questo perché la trasformata di Fourier può essere calcolata su segnali infiniti, ma poiché il segnale che si ha a disposizione per ovvie ragioni non lo è, allora con la trasformata potrebbero apparire delle repliche spettrali che in realtà non esistono, per far sì che ciò non accada allora si applica la finestra suddetta.
- Viene applicata la trasformata di Fourier (DFT-Discrete Fourier-Transform) sempre per ogni frame.
- Si applica il filtro Mel.
- Successivamente viene calcolato il logaritmo applicandolo solo al modulo della trasformata eseguita precedentemente.
- Infine, viene applicata la trasformata inversa di Fourier.

Generalmente vengono mantenuti 13 o 19 coefficienti per frame, in modo da mantenere solo i parametri più discriminativi e ridurre la dimensione dei dati. Da notare il fatto che a differenza del calcolo della trasformata Cepstrum, per gli MFCC viene aggiunto il passo di applicazione del filtro Mel.

I passi appena descritti sono riportati in figura 3.3.

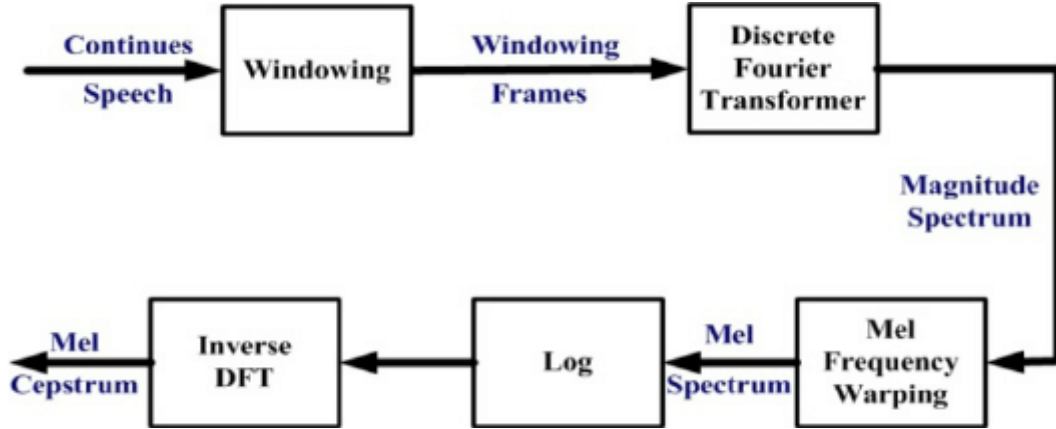


Figura 3.3: Schema di calcolo degli MFCC.[25]

3.3 Mfcc Delta e Delta-Delta

Oltre al calcolo delle mfcc sono stati calcolati i valori delta (quindi della derivata prima) e delta-delta (derivata seconda) conosciuti anche come coefficienti differenziali e di accelerazione. Il vettore della funzione MFCC descrive solo l'involuppo spettrale di potenza di un singolo fotogramma, ma si è capito che il parlato contiene anche informazioni sulla dinamica, ovvero quali sono le traiettorie dei coefficienti MFCC nel tempo. Si è scoperto quindi, che il calcolo delle traiettorie dell'MFCC e l'aggiunta di questo al vettore delle caratteristiche originale, aumenta le prestazioni del riconoscimento vocale di parecchio (se avessimo 12 coefficienti MFCC, otterremmo anche 12 coefficienti delta, che si combinerebbero per dare un vettore caratteristica di lunghezza 24). Per calcolare i coefficienti delta, viene utilizzata la seguente formula:

$$d_t = \frac{\sum_{n=1}^N n(c_{t+n} - c_{t-n})}{2 \sum_{n=1}^N n^2}$$

dove d_t è un coefficiente delta, dal frame t calcolato in termini di coefficienti statici c_{t+n} e c_{t-n} . Un valore tipico per N è 2. I coefficienti Delta-Delta (accelerazione) sono calcolati allo stesso modo, ma sono calcolati dai delta, non dai coefficienti statici. La

feature delta ci permette di capire qual è l'andamento temporale degli Mfcc mentre il delta-delta contiene informazioni temporali aggiuntive sul delta.

3.4 Pitch

Il pitch chiamato anche frequenza fondamentale è uno degli attributi più importanti per il riconoscimento delle emozioni nel parlato. L'ANSI Acoustical Terminology definisce il pitch come l'attributo uditivo rispetto al quale il suono può essere ordinato su una scala che assume valori da basso ad alto. Essa rappresenta, quindi, l'intonazione della voce. Livelli alti di pitch identificano tutte quelle emozioni, come la gioia, la rabbia e la paura, che mirano a richiamare l'attenzione di uno o più ascoltatori, mentre livelli bassi sono correlati ad emozioni più sobrie, come la tristezza o la calma. Livelli medi di pitch invece esprimono un'attitudine neutrale del parlante.

3.5 Zero-Crossing-Rate (ZCR)

La velocità di attraversamento zero misura la velocità delle variazioni di segno all'interno di un segnale audio. Questa funzione è stata ampiamente utilizzata sia nel riconoscimento vocale che nel recupero delle informazioni musicali.

3.6 Energy

L'energia permette di calcolare l'intensità e quindi la potenza del suono emesso. Il concetto di intensità è strettamente legato a quello di volume, che permette di distinguere i suoni in deboli e forti. È fondamentale analizzare questo parametro dal momento che gli esseri umani trasmettono emozioni diverse urlando o parlando sottovoce. In generale, l'intensità di emozioni come la rabbia, la paura e la gioia cresce durante la pronuncia di una frase, mentre diminuisce nel caso in cui l'emozione espressa sia la tristezza.

Per poter estrarre tutte le features descritte pocanzi è stata utilizzata la libreria di Python Librosa. A causa dell'intrattabilità di matrici di grosse dimensioni per eseguire il task di classificazione e del così detto "problema della dimensionalità", poiché ogni feature utilizzata era a sua volta formata da un vettore di valori, per esse sono stati calcolati diversi valori statistici principali. Qualora si fossero conservati tutti i vettori di ogni feature per intero, i pattern dei dati si sarebbero dispersi nel dataset, rendendo altamente complicato l'individuazione di uno schema generico in grado di descrivere nel migliore dei modi le informazioni. Inoltre, ogni algoritmo di apprendimento automatico avrebbe impiegato più tempo (complessità temporale) e occupato più memoria (complessità spaziale). Le informazioni memorizzate per ogni features sono descritte nella tabella qui sotto.

Nome feature	Attributo del dataset	Breve descrizione
Mfcc	Media Mediana Deviazione standard	Il numero di coefficienti restituiti sono 20. Siccome per ogni audio veniva restituita una matrice fatta da (20 x Num. Frames) e quindi la seconda dimensione dipendeva dalla lunghezza di ogni audio, allora si è deciso di estrapolare da ogni coefficiente solamente questi 3 valori scritti nella colonna di sinistra. In questo modo per ogni audio è stato ottenuto un vettore mfcc che fosse sempre della stessa dimensione. La dimensione ottenuta ogni volta quindi è 60.
Mfcc-delta	Media Mediana	Per lo stesso criterio utilizzato per gli mfcc di

	Deviazione standard	sopra anche in questo caso per ogni audio formato da una matrice (20xNum. Frames) si è deciso di calcolare solamente media, mediana e dev. Standard.
Mfcc-delta-delta	Media Mediana Deviazione standard	Stesso criterio di mfcc e mfcc-delta.
Pitch	Media Min Max Media log Max log Dev standard log	Per ogni audio la matrice del pitch aveva dimensione (1025xNum.frames). Per cercare di diminuire il numero di valori totali restituiti allora tutti i valori statistici ottenuti dal pitch sono stati calcolati sul vettore appiattito.
Zero crossing rate	Media Min Max Deviazione standard	Per ogni audio, la matrice ZCR aveva dimensione (1xNum.frames). Anche in questo caso i valori statistici di sinistra sono stati calcolati sul vettore appiattito.
Energy	Media Min Max Mediana Deviazione standard	Per ogni audio la matrice Energy aveva dimensione (1xNum.frames). I valori di sinistra sono stati calcolati sempre sull'array appiattito.

Capitolo 4 - Classificatori

I dataset nominati nel capitolo 2 sono stati utilizzati per testare vari classificatori in modo da individuare quale fosse quello migliore per il compito di classificazione delle 7 emozioni in base alla valence e arousal.

4.1 Random Forrest

La foresta casuale è un algoritmo di apprendimento automatico flessibile e facile da usare che produce, anche senza l'ottimizzazione degli iperparametri, un grande risultato nella maggior parte dei casi. È anche uno degli algoritmi più utilizzati, per la sua semplicità e diversità (può essere utilizzato sia per compiti di classificazione che di regressione). La foresta casuale è un algoritmo di apprendimento supervisionato. La "foresta" è costituita da un insieme di alberi decisionali, generalmente addestrati con la tecnica chiamata "bagging". L'idea generale di questa tecnica è che addestrando vari modelli su vari sottoinsiemi casuali di dati di un certo dataset si riesce ad ottenere un risultato complessivo migliore. In parole povere: la foresta casuale crea più alberi decisionali e li unisce per ottenere una previsione più accurata e stabile. Nella figura 4 qui sotto, si può notare come apparirebbe una foresta casuale con due alberi:

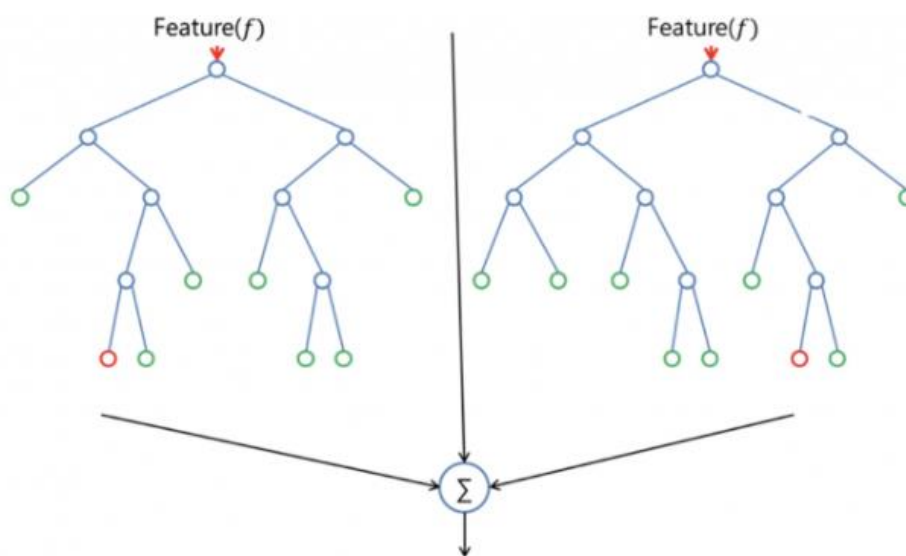


Figura 4: Foresta casuale formata da due alberi di decisione.[26]

La foresta casuale aggiunge ulteriore casualità al modello, mentre fa crescere gli alberi. Invece di cercare la funzionalità più importante durante la divisione di un nodo, cerca la funzione migliore tra un sottoinsieme casuale di funzionalità. Ciò si traduce in un'ampia diversità che generalmente si traduce in un modello migliore.

4.2 Gradient Boosting

Questo classificatore si basa sul concetto di cercare di convertire alberi di decisione deboli in alberi di decisione più forti dal punto di vista predittivo. Per fare ciò ogni nuovo albero si adatta ad una versione modificata del set di dati originale. Il Gradient boosting è composto da 3 elementi principali:

1. Una funzione di perdita da ottimizzare.
2. Diversi alberi di decisione che fungono da modelli più deboli.
3. Un modello additivo (in questo caso sempre albero di decisione) per aggiungere più modelli deboli per ridurre al minimo la funzione di perdita.

L'algoritmo di aumento gradiente può essere spiegato più facilmente introducendo l'algoritmo AdaBoost. L'algoritmo AdaBoost inizia allenando un albero decisionale in cui a ciascuna osservazione viene assegnato un peso uguale. Dopo aver valutato il primo albero, si aumentano i pesi di quelle osservazioni che sono difficili da classificare e si abbassano i pesi per quelli che sono facili da classificare. Il secondo albero viene quindi sviluppato su questi dati ponderati. Qui, l'idea è di migliorare le previsioni del primo albero. Il nuovo modello è quindi un $\text{Tree1} + \text{Tree2}$. Quindi si calcola l'errore di classificazione da questo nuovo modello composto da 2 alberi in modo da sviluppare così un terzo albero per prevedere altri dati. Si ripete questo processo per un numero specificato di iterazioni. Gli alberi successivi aiutano a classificare le osservazioni che non sono ben classificate dagli alberi precedenti. Le previsioni del modello di ensemble finale sono quindi la somma ponderata delle previsioni fatte dai precedenti modelli di alberi. Il Gradient Boosting allena molti modelli in modo graduale, additivo e sequenziale. La principale differenza tra AdaBoost e Gradient Boosting Algorithm è il modo in cui i due algoritmi identificano le carenze dei modelli più deboli (ad es. Alberi di decisioni). Mentre il modello AdaBoost identifica le carenze utilizzando come punti i

pesi più elevati, il Gradient Boosting esegue lo stesso utilizzando però la discesa di gradiente per trovare il valore minimo della funzione di perdita. Tradizionalmente, la discesa del gradiente viene utilizzata per ridurre al minimo una serie di parametri, come i coefficienti in un'equazione di regressione o i pesi in una rete neurale. Dopo aver calcolato l'errore o la perdita, i pesi vengono aggiornati per ridurre al minimo tale errore. Invece di parametri, in questo caso abbiamo sotto-modelli più deboli. Dopo aver calcolato la perdita, per eseguire la procedura di discesa del gradiente, è necessario aggiungere un ulteriore albero al modello in modo tale che si riduca il valore della funzione di perdita (ovvero seguire il gradiente). Lo si fa parametrizzando l'albero, quindi modificando i suoi parametri, in modo da spostarsi nella giusta direzione (riducendo la perdita residua). La funzione di perdita è una misura che indica quanto sono buoni i coefficienti del modello nell'apprendere i dati di addestramento. Una comprensione logica della funzione di perdita dipenderebbe da ciò che stiamo cercando di ottimizzare. Ad esempio, se stiamo cercando di prevedere i prezzi di vendita utilizzando una regressione, la funzione di perdita si baserebbe sull'errore tra i prezzi delle case reali e quelli previsti. Una delle maggiori motivazioni dell'uso del gradiente è che consente di ottimizzare una funzione di costo specificata dall'utente, anziché una funzione di perdita che di solito offre meno controllo e non corrisponde essenzialmente alle applicazioni del mondo reale.

4.3 Support Vector Machine (SVM)

Support Vector Machine, abbreviato in SVM, è un classificatore che può essere utilizzato sia per le attività di regressione che di classificazione. Essa è però ampiamente utilizzata negli obiettivi di classificazione. L'obiettivo dell'algoritmo della macchina vettoriale di supporto è quello di trovare un iperpiano in uno spazio N-dimensionale (N corrisponde al numero di funzioni, chiamate anche features) che classifica distintamente i punti dati. La SVM nasce come classificatore binario ma può essere utilizzato anche per problemi di classificazione multiclasse. Nel caso binario, per separare le due classi di punti dati, ci sono molti possibili iperpiani che possono essere scelti. L'obiettivo della SVM è quello di trovare un iperpiano tra i tanti possibili che abbia il margine massimo, ovvero la distanza massima tra i punti dati di entrambe le classi. Massimizzare la distanza dal

marginale fornisce un certo rinforzo in modo che i punti di dati futuri possano essere classificati con maggiore sicurezza.

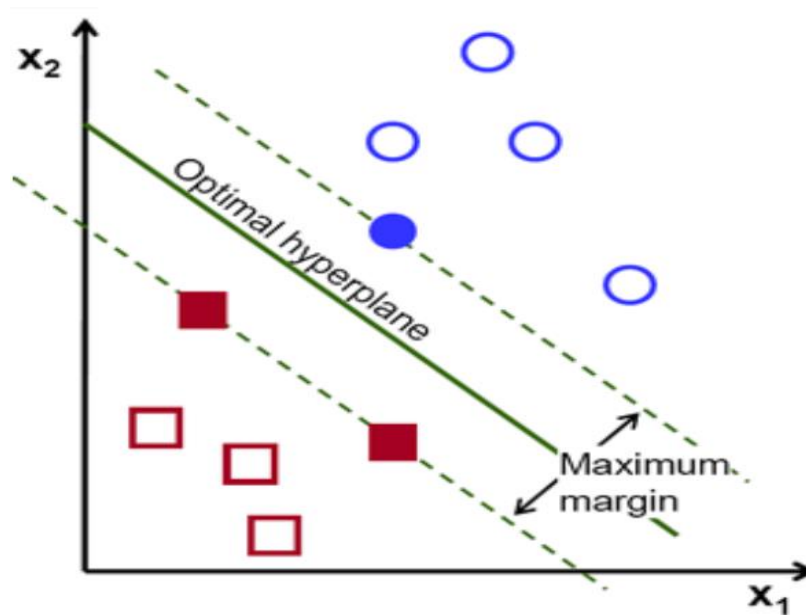


Figura 4.1: Rappresentazione di un iperpiano (in questo caso una retta) in uno spazio a due dimensioni, ove i vari punti rappresentano i dati di un set di allenamento.[27]

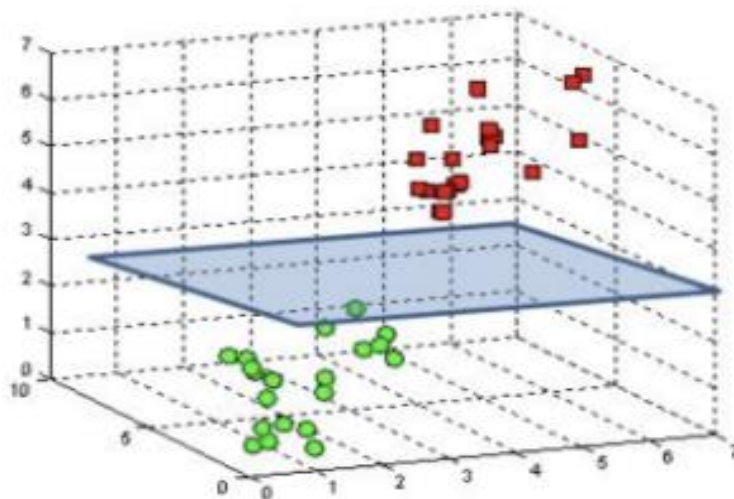


Figura 4.2: Rappresentazione di un iperpiano (in questo caso è un piano) in uno spazio tridimensionale.[28]

Gli iperpiani sono limiti di decisione che aiutano a classificare i punti dati. I punti che ricadono su entrambi i lati dell'iperpiano possono essere attribuiti a classi diverse. Inoltre, la dimensione dell'iperpiano dipende dal numero di funzioni. Se il numero di funzioni di input è 2, l'iperpiano è solo una linea. Se il numero di funzioni di input è 3, l'iperpiano diventa un piano bidimensionale. Diventa difficile immaginarlo quando il numero di funzioni supera 3. Per eseguire la classificazione multiclasse invece, esistono due metodi possibili:

1. Costruire vari classificatori one-versus-all (OVA) e scegliere la classe che classifica il dato di prova con il massimo margine. Con la libreria in Python chiamata **Sklearn**, che è stata utilizzata in questo lavoro di tesi, viene implementato questo metodo per default, ed è stato quello che si è utilizzato in questo caso.
2. Costruire un insieme di classificatori one-versus-one (OVO) e scegliere la classe selezionata dalla maggior parte dei classificatori.

4.4 Multi-Layer Perceptron (MLP)

Gli esseri umani hanno la capacità di identificare modelli all'interno delle informazioni accessibili con un livello sorprendentemente elevato di accuratezza. Ogni volta che si vede un'auto o una bicicletta l'uomo è in grado riconoscere immediatamente cosa sono. Questo perché ha appreso in un certo periodo di tempo l'aspetto di un'auto e di una bicicletta e quali sono le loro caratteristiche distintive. Le reti neurali artificiali sono sistemi di calcolo che intendono imitare le capacità di apprendimento umano attraverso un'architettura complessa che ricorda il sistema nervoso umano. Il sistema nervoso umano è costituito da miliardi di neuroni. Questi neuroni elaborano collettivamente l'input ricevuto dagli organi sensoriali, elaborano l'informazione e decidono cosa fare in risposta all'input. Le reti neurali artificiali sono ispirate all'architettura della rete neurale umana. La rete neurale più semplice è costituita da un solo neurone ed è chiamata **percettrone**, come mostrato nella figura sottostante 4.3. Un percettrone ha uno strato di input e un neurone. Il livello di input è responsabile della ricezione degli input. Il numero di nodi nel livello di input è uguale al numero di funzioni (features) nel set di dati

di input. Ogni input viene moltiplicato per un peso (che in genere viene inizializzato con un valore casuale) e i risultati vengono sommati. La somma viene quindi passata attraverso una funzione di attivazione. La funzione di attivazione di un perceptrone ricorda il nucleo del neurone del sistema nervoso umano. Elabora le informazioni e produce un output. Nel caso di un perceptrone, questo risultato è il risultato finale. Tuttavia, nel caso di reti neurali multistrato, l'output dai neuroni nello strato precedente funge da input per i neuroni dello strato successivo.

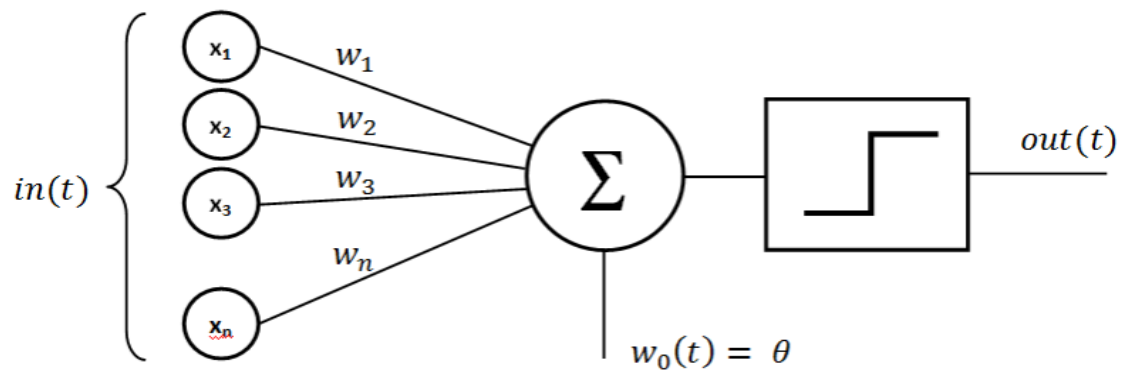


Figura 4.3: Struttura di un perceptrone.[29]

Un perceptrone a singolo strato può risolvere semplici problemi in cui i dati sono separabili linearmente in dimensioni n , dove n è il numero di funzioni nel set di dati. Tuttavia, nel caso di dati non linearmente separabili, l'accuratezza del perceptrone a strato singolo diminuisce in modo significativo. I perceptron multistrato, d'altra parte, possono lavorare in modo efficiente con dati separabili in modo non lineare. I perceptron multistrato, o più comunemente indicati come reti neurali artificiali, sono una combinazione di più neuroni collegati sotto forma di una rete. Una rete neurale artificiale ha uno strato di input, uno o più livelli nascosti e uno di output. Questo è mostrato nell'immagine 4.4.

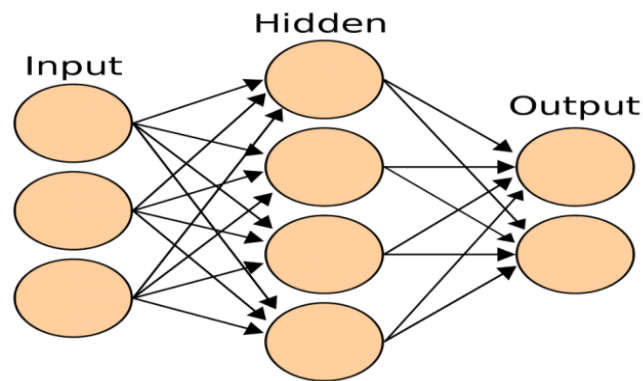


Figura 4.4: Struttura di una rete neurale artificiale.[29]

Una rete neurale viene eseguita in due fasi: Feed-Forward (propagazione in avanti) e Back Propagation (propagazione all'indietro). Di seguito sono riportati i passaggi eseguiti durante la fase di feed-forward:

1. I valori ricevuti nel livello di input vengono moltiplicati per i pesi. Viene aggiunto un bias alla somma degli input e dei pesi per evitare valori nulli.
2. Ogni neurone nel primo strato nascosto riceve valori diversi dallo strato di input in base ai pesi e alla distorsione. I neuroni hanno una funzione di attivazione che opera sul valore ricevuto dal livello di input. La funzione di attivazione può essere di molti tipi, (sigmoid, relu o una funzione tanh). Come regola empirica, la funzione relu viene utilizzata nei neuroni dello strato nascosto e la funzione sigmoide viene utilizzata per il neurone dello strato di uscita.
3. Gli output dei neuroni del primo strato nascosto vengono moltiplicati per i pesi del secondo strato nascosto; i risultati vengono sommati e passati ai neuroni degli strati successivi. Questo processo continua fino a raggiungere lo strato esterno. I valori calcolati sul livello esterno sono gli output effettivi dell'algoritmo.

Tuttavia, l'output restituito non è detto che sia necessariamente corretto immediatamente; può essere sbagliato e quindi bisogna correggerlo. Per migliorare questi risultati previsti, una rete neurale passerà attraverso una fase di propagazione all'indietro. Durante questa fase, i pesi dei diversi neuroni vengono aggiornati in modo tale che la differenza tra l'output desiderato e quello previsto sia la più piccola possibile. La fase di propagazione posteriore consiste nei seguenti passaggi:

1. L'errore viene calcolato quantificando la differenza tra l'output previsto e l'output desiderato. Questa differenza si chiama "perdita" e la funzione utilizzata per calcolare la differenza si chiama "funzione di perdita". Le funzioni di perdita possono essere di diversi tipi, ad esempio errore quadratico medio o funzioni di entropia incrociata.
2. Una volta calcolato l'errore, il passaggio successivo è minimizzare tale errore. Per fare ciò, la derivata parziale della funzione di errore viene calcolata rispetto a tutti i pesi e i biases. L'algoritmo che esegue questo passaggio è chiamato 'Discesa di gradiente'. Le derivate possono essere utilizzate per trovare la pendenza della funzione di errore. Se la pendenza è positiva, i valori dei pesi possono essere ridotti mentre se la pendenza è negativa, i valori dei pesi possono essere aumentati. Ciò riduce l'errore generale. La funzione utilizzata per ridurre questo errore è chiamata funzione di ottimizzazione.

Questo ciclo di propagazione feed-forward e back è chiamato "epoca". Questo processo continua fino a quando non viene raggiunta una ragionevole accuratezza.

4.5 K-Nearest Neighbors

Il **K-Nearest Neighbors** è un algoritmo di classificazione utilizzato nel riconoscimento di pattern per eseguire la classificazione di oggetti basandosi sulle caratteristiche degli oggetti vicini a quello considerato. Quindi sostanzialmente quello che succede è che dato un nuovo esempio, si usano i k esempi più simili per predire la classe a cui quell'esempio di partenza appartiene. Per trovare la similarità tra gli esempi si utilizza una metrica. La metrica che in genere è utilizzata è la **distanza Euclidea (utilizzata in questo progetto)** ma si possono usare anche altre come la distanza Manhattan oppure la distanza di Hamming qualora si dovessero manipolare stringhe e non numeri. In generale la formula per calcolare la distanza Euclidea tra due punti in uno spazio n-dimensiona è la seguente.

Siano $P = (p_1, p_2, p_3, \dots, p_n)$ e $Q = (q_1, q_2, q_3, \dots, q_n)$ due punti in uno spazio n -dimensionale, allora:

$$d(P, Q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Capitolo 5 - Sperimentazione e Risultati

In questo capitolo vengono presentati i test eseguiti sui vari dataset utilizzando le features descritte nel capitolo 2 per etichettare ogni audio secondo la valence e l'arousal. Ogni test è stato eseguito con l'uso della tecnica della **stratified k-fold cross-validation (CVS)** che si differenzia dalla cross-validation normale perché ogni volta che un dataset viene diviso in k partizioni diverse, si cerca di mantenere la stessa percentuale di campioni con la stessa etichetta per ciascuna partizione, in modo da poter ottenere valori di accuratezza che tengano conto di partizioni equilibrate che garantiscono una buona rappresentazione di tutte le classi. Il valore di k scelto è 5. Prima di eseguire il partizionamento di ogni dataset i dati vengono mescolati attraverso il settaggio del parametro Shuffle a **true** della libreria StratifiedKFold di Sklearn. I valori delle features di ogni audio sono stati standardizzati attraverso l'utilizzo del metodo **StandardScaler()** della libreria **Sklearn.preprocessing** prima di darli in input ai vari classificatori. Questa tecnica praticamente prende tutti i valori di ogni colonna del dataset (quindi di ogni feature) e li trasforma in modo tale che la **media sia 0** e la **varianza sia 1**. La trasformazione viene eseguita sottraendo ad ogni valore di ogni feature la media e successivamente eseguendo la divisione tra il risultato ottenuto e la deviazione standard. Ovviamente sia la media che la deviazione standard vengono calcolati all'inizio specificatamente per ogni singola feature. L'utilizzo di questa tecnica si ritiene essere di fondamentale importanza soprattutto per la SVM, il KNN e la MLP perché grazie ad essa **i valori di accuratezza in media aumentano del 10%**, e in alcuni casi anche di più. Per il Random Forrest e il

Gradient Boosting, nella maggior parte dei casi l'utilizzo di questo metodo ha aumentato l'accuratezza dei risultati, altre volte invece la diminuivano, nonostante ciò però si è deciso di mostrare solo i risultati dove viene utilizzata, in quanto, anche senza di essa, la decisione di quale sia il modello migliore rimane invariante.

5.1 Ravdess

CLASS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.73%	0.72%	0.76%	0.78%	0.77%	0.75%
GB	0.76%	0.75%	0.78%	0.80%	0.77%	0.77%
SVM	0.80%	0.83%	0.84%	0.84%	0.85%	0.83%
MLP	0.62%	0.64%	0.59%	0.68%	0.63%	0.63%
KNN	0.74%	0.74%	0.72%	0.76%	0.75%	0.74%

Tabella 5: Risultati della CVS per la valence solo sul dataset Ravdess.

CLASS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.77%	0.78%	0.79%	0.77%	0.77%	0.77%
GB	0.79%	0.81%	0.80%	0.78%	0.79%	0.79%
SVM	0.83%	0.88%	0.81%	0.81%	0.79%	0.82%
MLP	0.62%	0.61%	0.66%	0.65%	0.63%	0.63%
KNN	0.74%	0.78%	0.74%	0.71%	0.73%	0.74%

Tabella 5.1: Risultati della CVS per l'arousal solo sul dataset Ravdess.

5.2 Tess

CLASS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.98%	0.99%	0.99%	0.99%	0.99%	0.99%
GB	0.99%	0.99%	1.0%	0.99%	0.99%	0.99%
SVM	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%

MLP	0.82%	0.83%	0.82%	0.81%	0.80%	0.81%
KNN	0.99%	0.99%	0.99%	0.99%	1.0%	0.99%

Tabella 5.2: Risultati della CVS per la valence solo sul dataset Tess.

CLASS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	1.0%	0.99%	0.99%	0.99%	0.99%	0.99%
GB	0.99%	1.0%	0.99%	0.99%	1.0%	0.99%
SVM	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%
MLP	0.82%	0.83%	0.84%	0.85%	0.82%	0.83%
KNN	1.0%	0.99%	1.0%	0.99%	1.0%	1.0%

Tabella 5.3: Risultati della CVS per l'arousal solo sul dataset Tess.

5.3 Savee

CLASS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.77%	0.74%	0.79%	0.70%	0.73%	0.75%
GB	0.73%	0.74%	0.74%	0.77%	0.71%	0.74%
SVM	0.76%	0.73%	0.78%	0.69%	0.66%	0.73%
MLP	0.42%	0.50%	0.45%	0.46%	0.44%	0.45%
KNN	0.72%	0.72%	0.69%	0.70%	0.62%	0.69%

Tabella 5.4: Risultati della CVS per la valence solo sul dataset Savee.

CLASS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.95%	0.88%	0.88%	0.92%	0.80%	0.89%
GB	0.95%	0.85%	0.85%	0.91%	0.86%	0.88%
SVM	0.89%	0.82%	0.83%	0.84%	0.81%	0.84%
MLP	0.58%	0.62%	0.53%	0.54%	0.55%	0.56%
KNN	0.82%	0.76%	0.73%	0.85%	0.85%	0.80%

Tabella 5.5: Risultati della CVS per l'arousal solo sul dataset Savee.

5.4 Emovo

CLASS:	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.69%	0.78%	0.70%	0.75%	0.76%	0.74%
GB	0.66%	0.74%	0.68%	0.73%	0.74%	0.71%
SVM	0.65%	0.66%	0.75%	0.70%	0.74%	0.70%
MLP	0.38%	0.34%	0.49%	0.39%	0.50%	0.42%
KNN	0.65%	0.62%	0.74%	0.61%	0.67%	0.66%

Tabella 5.6: Risultati della CVS per la valence solo sul dataset Emovo.

CLASS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.80%	0.74%	0.79%	0.74%	0.67%	0.75%
GB	0.83%	0.80%	0.87%	0.86%	0.74%	0.82%
SVM	0.77%	0.73%	0.74%	0.80%	0.74%	0.76%
MLP	0.55%	0.64%	0.62%	0.62%	0.56%	0.60%
KNN	0.76%	0.74%	0.77%	0.80%	0.71%	0.76%

Tabella 5.7: Risultati della CVS per l'arousal solo sul dataset Emovo

5.5 Emofilm

CLAS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.80%	0.82%	0.80%	0.80%	0.80%	0.80%
GB	0.80%	0.82%	0.81%	0.84%	0.82%	0.82%
SVM	0.80%	0.84%	0.79%	0.82%	0.81%	0.81%
MLP	0.77%	0.77%	0.77%	0.75%	0.76%	0.76%
KNN	0.79%	0.80%	0.80%	0.80%	0.80%	0.80%

Tabella 5.8: Risultati della CVS per la valence solo sul dataset Emofilm.

CLASS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.76%	0.71%	0.76%	0.78%	0.80%	0.76%
GB	0.69%	0.73%	0.75%	0.76%	0.78%	0.74%
SVM	0.71%	0.68%	0.73%	0.75%	0.73%	0.72%
MLP	0.63%	0.58%	0.64%	0.58%	0.65%	0.62%
KNN	0.74%	0.69%	0.67%	0.72%	0.63%	0.69%

Tabella 5.9: Risultati della CVS per l'arousal solo sul dataset Emofilm.

5.6 Emodb

CLASS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.79%	0.76%	0.77%	0.78%	0.80%	0.78%
GB	0.85%	0.79%	0.81%	0.82%	0.82%	0.82%
SVM	0.75%	0.81%	0.81%	0.82%	0.79%	0.80%
MLP	0.73%	0.70%	0.68%	0.60%	0.62%	0.67%
KNN	0.74%	0.77%	0.76%	0.80%	0.78%	0.77%

Tabella 5.10: Risultati della CVS per la valence solo sul dataset Emodb.

CLASS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.88%	0.82%	0.87%	0.92%	0.89%	0.88%
GB	0.90%	0.88%	0.88%	0.92%	0.84%	0.89%
SVM	0.92%	0.95%	0.87%	0.91%	0.89%	0.91%
MLP	0.69%	0.66%	0.66%	0.68%	0.67%	0.67%
KNN	0.88%	0.90%	0.80%	0.80%	0.81%	0.84%

Tabella 5.11: Risultati della CVS per l'arousal solo sul dataset Emodb.

5.7 Demos

CLASS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.72%	0.71%	0.72%	0.71%	0.72%	0.72%
GB	0.77%	0.77%	0.77%	0.76%	0.78%	0.77%
SVM	0.79%	0.79%	0.79%	0.79%	0.81%	0.79%
MLP	0.67%	0.68%	0.68%	0.66%	0.67%	0.67%
KNN	0.70%	0.70%	0.71%	0.71%	0.73%	0.71%

Tabella 5.12: Risultati della CVS per la valence solo sul dataset Demos.

CLASS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.71%	0.72%	0.72%	0.69%	0.71%	0.71%
GB	0.75%	0.76%	0.74%	0.74%	0.73%	0.74%

SVM	0.78%	0.78%	0.80%	0.78%	0.77%	0.78%
MLP	0.59%	0.61%	0.61%	0.58%	0.61%	0.60%
KNN	0.67%	0.67%	0.66%	0.68%	0.66%	0.67%

Tabella 5.13: Risultati della CVS per l'arousal solo sul dataset Demos.

5.8 Unico dataset con audio del bilanciamento

Un ulteriore test è stato eseguito su un unico dataset formato da Ravdess, Tess, Savee, Emofilm, Emovo e Emodb completi, utilizzando Demos solo per riuscire ad equilibrare il numero di campioni di tutte le classi. Il numero di file audio complessivi iniziali è 7210, poiché è stata applicata la tecnica di data augmentation (spiegata nella sezione 6.8), il numero di file complessivi raddoppia, arrivando a 14420 audio del dataset bilanciato. I file iniziali sono distribuiti in questo modo:

Emozione	Numero audio
Disgusto	1030
Gioia	1030
Neutrale	1030
Paura	1030
Rabbia	1030
Sorpresa	1030
Tristezza	1030

CLASS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.84%	0.85%	0.83%	0.83%	0.84%	0.84%
GB	0.85%	0.86%	0.85%	0.85%	0.85%	0.85%
SVM	0.95%	0.92%	0.92%	0.94%	0.93%	0.93%
MLP	0.65%	0.67%	0.66%	0.66%	0.67%	0.66%
KNN	0.87%	0.85%	0.86%	0.86%	0.84%	0.85%

Tabella 5.14: Risultati della CVS per la valence sul dataset bilanciato.

CLASS.	PARTIZIONE 1	PARTIZIONE 2	PARTIZIONE 3	PARTIZIONE 4	PARTIZIONE 5	MEDIA
RDF	0.85%	0.85%	0.84%	0.84%	0.84%	0.84%
GB	0.86%	0.86%	0.86%	0.85%	0.84%	0.85%
SVM	0.93%	0.92%	0.93%	0.93%	0.92%	0.93%
MLP	0.73%	0.72%	0.72%	0.72%	0.71%	0.72%
KNN	0.86%	0.87%	0.88%	0.86%	0.86%	0.87%

Tabella 5.15: Risultati della CVS per l'arousal sul dataset bilanciato.

5.9 Analisi dei risultati:

Dall'analisi degli esperimenti eseguiti nel capitolo 5 si può notare come la SVM sia il classificatore in grado di dare i risultati mediamente migliori per quanto riguarda l'accuratezza su quasi tutti i dataset, sia per quanto riguarda la valence che l'arousal. Da notare il grande livello di accuratezza che si ha sul dataset bilanciato. Per queste ragioni si è deciso di utilizzarla come modello finale da comparare con la rete neurale che verrà presentata nel capitolo successivo. I parametri utilizzati per la SVM (della libreria Sklearn) sono i seguenti:

- $C = 10$ (parametro di regolarizzazione)
- $\text{Gamma} = 0.01$ (coefficiente del Kernel)
- Kernel = 'rbf' (specifica il tipo di Kernel utilizzato, quello usato in questo caso è chiamato kernel esponenziale quadrato)

Capitolo 6 Convolutional Neural Network (CNN)

La **Convolutional Neural Networks (CNN)** rappresenta un'architettura di rete neurale artificiale di grande successo nelle applicazioni di visione artificiale e ampiamente utilizzate anche in applicazioni che processano media come audio e video.

Un'architettura di rete neurale convoluzionale può essere formata da vari livelli, che in genere sono i seguenti:

- **Livello di input:** rappresenta l'insieme di numeri che rappresenta, per il computer, l'immagine da analizzare. Essa è rappresentata come un insieme di pixel. Ad esempio, 32 x 32 x 3 indica la larghezza (32), altezza (32) e profondità (3, i tre colori Red, Green e Blue nel formato RGB) dell'immagine.
- **Livello convoluzionale (Conv):** è il livello principale della rete. Il suo obiettivo è quello di individuare **schemi**, come ad esempio curve, angoli, circonferenze o quadrati raffigurati in un'immagine con elevata precisione. Sono più di uno, e ognuno di essi si concentra nella ricerca di queste caratteristiche nell'immagine iniziale. Maggiore è il loro numero e maggiore è la complessità della caratteristica che riescono ad individuare.
- **Livello ReLU (Rectified Linear Units):** si pone l'obiettivo di annullare valori negativi ottenuti nei livelli precedenti e solitamente è posto dopo i livelli convoluzionali.
- **Livello Pool:** permette di identificare se la caratteristica di studio è presente nel livello precedente. Semplifica e rende più grezza l'immagine, mantenendo la caratteristica utilizzata dal livello convoluzionale.
- **Livello FC (o Fully connected, completamente connesso):** connette tutti i neuroni del livello precedente al fine di stabilire le varie classi identificative visualizzate nei precedenti livelli secondo una determinata probabilità. Ogni classe rappresenta una possibile risposta finale che il computer darà.

6.1 Convolutional Layer

Per riuscire a capire meglio come lavora il livello convoluzionale di una CNN supponiamo di dare alla rete in input un'immagine rappresentante un 7. Nel linguaggio macchina tale figura si può rappresentare tramite un array di 28x28x3 pixel.



Figura 6: Rappresentazione della cifra sette a sinistra su un foglio e a destra secondo una sfumatura del colore rosso con valori compresi nel range $[0,1]$. [30]

Prima di continuare occorre spiegare cosa sono i filtri e cosa rappresenta il passo. Per filtro generalmente, si intende una piccola matrice di poche righe e colonne che rappresenta una caratteristica (feature) che il livello convoluzionale vuole identificare, ad esempio le curve o una linea retta. Inizialmente per i primi livelli si dice che il filtro rappresenta una caratteristica di **basso livello** perché identifica semplici oggetti come appunto curve o linee. Per un livello convoluzionale il filtro identificherà le curve, per un altro linee orizzontali, per un altro ancora circonferenze, e così via negli ultimi livelli, fino a formare figure complesse che rappresenteranno oggetti più complicati. In quest'ultimo caso si dice che il filtro rappresenta una caratteristica di **alto livello** perché identifica oggetti complessi, come ad esempio una mano o un volto. Si può supporre che il filtro sia un rivelatore di curve. Una volta identificata la caratteristica che il filtro identificherà nel livello convoluzionale, si decide la **dimensione del filtro** e il **numero di filtri** da utilizzare nel livello. Si supponga ad esempio di utilizzare un filtro dalle

dimensioni 3x3 (ossia 3 righe e 3 colonne), che per il primo livello convoluzionale assume i valori casuali (detti anche **pesi**).

FILTRO		
0.979	0.278	0.940
0.713	0.048	0.564
0.604	0.327	0.853

Figura 6.1: Esempio di filtro di dimensione 3x3.[30]

Si ipotizza per semplicità un solo filtro anche se in realtà sono diversi per ogni livello. A questo punto si parte ad analizzare il cosiddetto **campo ricettivo**, che ha la stessa dimensione del filtro (quindi in questo caso 3x3). Esso viene inizialmente rappresentato dal primo blocco di pixel 3x3 in alto a sinistra del livello di input (quindi dell'immagine data in ingresso alla rete neurale). Il risultato, che si otterrà in alto a sinistra nel livello successivo (**Conv1**) della rete neurale, si ottiene facendo un **prodotto scalare** dei valori del filtro con i valori di questo primo blocco (cioè facendo moltiplicazioni a livello di elemento tra i pixel 3x3 del campo ricettivo e i pesi dei neuroni del filtro 3x3, e infine il risultato sarà la somma dei vari prodotti così ottenuti). Questo darà un numero unico: tale valore sarà più alto in prossimità di curve, e più in basso nel caso contrario.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD	BE	BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ	BR	BS	BT	BV	BW	BX	BY	BZ	CA	CB	CC	CD	CE	CF	CG	CH	CI	CJ	CK	CL	CM	CN	CO	CP	CQ	CR	CS	CT	CU	CV	CW	CX	CY	CZ	DA	DB	DC	DD	DE	DF	DG	DH	DI	DJ	DK	DL	DM	DN	DO	DP	DQ	DR	DS	DT	DU	DV	DW	DX	DY	DZ	EA	EB	EC	ED	EE	EF	EG	EH	EI	EJ	EK	EL	EM	EN	EO	EP	EQ	ER	ES	ET	EU	EV	EW	EX	EY	EZ	FA	FB	FC	FD	FE	FF	FG	FH	FI	FJ	FK	FL	FM	FN	FO	FP	FQ	FR	FS	FT	FU	FV	FW	FX	FY	FZ	GA	GB	GC	GD	GE	GF	GG	GH	GI	GJ	GK	GL	GM	GN	GO	GP	GQ	GR	GS	GT	GU	GV	GW	GX	GY	GZ	HA	HB	HC	HD	HE	HF	HG	HH	HI	HJ	HK	HL	HM	HN	HO	HP	HQ	HR	HS	HT	HU	HV	HW	HX	HY	HZ	IA	IB	IC	ID	IE	IF	IG	IH	II	IJ	IK	IL	IM	IN	IO	IP	IQ	IR	IS	IT	IU	IV	IW	IX	IY	IZ	JA	JB	JC	JD	JE	JF	JG	JH	JI	IJ	JK	JL	JM	JN	JO	JP	JQ	JR	JS	JT	JU	JV	JW	JX	JY	JZ	KA	KB	KC	KD	KE	KF	KG	KH	KI	KJ	KK	KL	KM	KN	KO	KP	KQ	KR	KS	KT	KU	KV	KW	KX	KY	KZ	LA	LB	LC	LD	LE	LF	LG	LH	LI	LJ	LK	LL	LM	LN	LO	LP	LQ	LR	LS	LT	LU	LV	LW	LX	LY	LZ	MA	MB	MC	MD	ME	MF	MG	MH	MI	MJ	MK	ML	MM	MN	MO	MP	MQ	MR	MS	MT	MU	MV	MW	MX	MY	MZ	NA	NB	NC	ND	NE	NF	NG	NH	NI	NJ	NK	NL	NM	NN	NO	NP	NQ	NR	NS	NT	NU	NV	NW	NX	NY	NZ	OA	OB	OC	OD	OE	OF	OG	OH	OI	OJ	OK	OL	OM	ON	OO	OP	OQ	OR	OS	OT	OU	OV	OW	OX	OY	OZ	PA	PB	PC	PD	PE	PF	PG	PH	PI	PJ	PK	PL	PM	PN	PO	PP	PQ	PR	PS	PT	PU	PV	PW	PX	PY	PZ	QA	QB	QC	QD	QE	QF	QG	QH	QI	QJ	QK	QL	QM	QN	QO	QP	QQ	QR	QS	QT	QU	QV	QW	QX	QY	QZ	RA	RB	RC	RD	RE	RF	RG	RH	RI	RJ	RK	RL	RM	RN	RO	RP	RQ	RR	RS	RT	RU	RV	RW	RX	RY	RZ	SA	SB	SC	SD	SE	SF	SG	SH	SI	SJ	SK	SL	SM	SN	SO	SP	SQ	SR	SS	ST	SU	SV	SW	SX	SY	SZ	TA	TB	TC	TD	TE	TF	TG	TH	TI	TJ	TK	TL	TM	TN	TO	TP	TQ	TR	TS	TT	TU	TV	TW	TX	TY	TZ	UA	UB	UC	UD	UE	UF	UG	UH	UI	UJ	UK	UL	UM	UN	UO	UP	UQ	UR	US	UT	UU	UV	UW	UX	UY	UZ	VA	VB	VC	VD	VE	VF	VG	VH	VI	VJ	VK	VL	VM	VN	VO	VP	VQ	VR	VS	VT	VU	VV	VW	VX	VY	VZ	WA	WB	WC	WD	WE	WF	WG	WH	WI	WJ	WK	WL	WM	WN	WO	WP	WQ	WR	WS	WT	WU	WV	WW	WX	WY	WZ	XA	XB	XC	XD	XE	XF	XG	XH	XI	XJ	XK	XL	XM	XN	XO	XP	XQ	XR	XS	XT	XU	XV	XW	XX	XY	XZ	YA	YB	YC	YD	YE	YF	YG	YH	YI	YJ	YK	YL	YM	YN	YO	YP	YQ	YR	YS	YT	YU	YV	YW	YX	YY	YZ	ZA	ZB	ZC	ZD	ZE	ZF	ZG	ZH	ZI	ZJ	ZK	ZL	ZM	ZN	ZO	ZP	ZQ	ZR	ZS	ZT	ZU	ZV	ZW	ZX	ZY	ZZ
1	Input																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												</																																																																																																																																																																																																

Figura 6.2: Al centro delle due immagini è presente il filtro (figura 6.1), in alto a sinistra del livello di input (immagine di sinistra) è presente il campo ricettivo mentre il risultato del prodotto scalare tra il filtro e il campo ricettivo è presente come primo valore del livello successivo chiamato Conv1 (immagine di destra). [30]

Con riferimento alla figura 6.2, siccome nell'immagine sopra il primo risultato non è in prossimità di curve esso assumerà valore 0 (inoltre ogni pixel del volume di input assume valore nullo, quindi il risultato del prodotto scalare è pari a zero). L'operazione descritta pocanzi va ripetuta per tutti i blocchi che l'immagine di input può contenere. Di conseguenza, il campo ricettivo viene fatto spostare di un determinato passo (o **stride**) verso destra. Ad esempio, ipotizzando di avere un volume di input 7x7 (numero di righe e di colonne del livello di input), un filtro 3x3 e un passo 1, il campo ricettivo (sempre 3x3) si sposterà di un'unità verso destra per analizzare il secondo blocco (e così via per tutti gli altri, fino a ricoprire l'intero volume di input). Tutto questo è rappresentato in Figura 6.3.

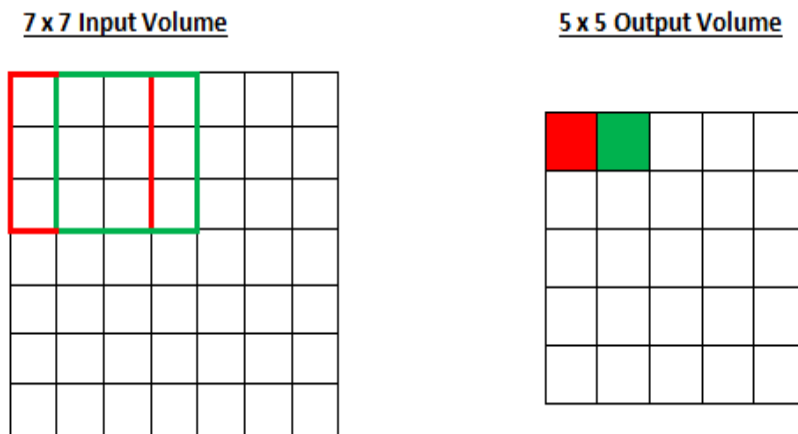


Figura 6.3: A sinistra è rappresentato il volume di input su cui il filtro verrà applicato e a destra il volume di output.[30]

Dopo aver fatto scivolare il campo ricettivo su tutte le posizioni del livello di input presente nella figura 6.2, si ottiene una matrice di numeri $26 \times 26 \times 1$. La ragione per cui è 26×26 è che si ottengono 686 differenti pixel che un filtro di 3×3 può analizzare in un'immagine 28×28 , come era quella proposta all'inizio. La profondità, invece, risulta pari a 1 perché in questo esempio è stato utilizzato un solo filtro. L'insieme dei valori che si ottengono seguendo la procedura appena enunciata si dice mappa di attivazione (rappresentata in questo caso dalla matrice $26 \times 26 \times 1$). In presenza di più filtri, invece, la stessa procedura andava ripetuta, per un numero di volte pari al numero dei filtri del livello (n), ottenendo n mappe di attivazione.

Il quarto ed ultimo parametro che influenza il comportamento di un livello convoluzionale è **detto riempimento zero**, o **zero-padding** dall'inglese: esso identifica uno strato da apporre al volume di input iniziale al fine di evitare di perdere alcune informazioni al passaggio da un livello a un altro. Se infatti ad esempio si applicassero tre filtri $5 \times 5 \times 3$ a un volume di input $32 \times 32 \times 3$ si otterrebbe un volume di output che sarebbe $28 \times 28 \times 3$. Il motivo è sempre che si ottengono 784 differenti pixel che un filtro 5×5 può coprire in un'immagine 32×32 . La profondità è data dal numero di filtri, quindi 3. Come si può notare le dimensioni spaziali diminuiscono. Man mano che si continuano ad applicare i livelli convoluzionali, la dimensione del volume diminuirà più velocemente di quanto si vorrebbe. Nei primi strati della nostra rete, però, è bene conservare il maggior numero di informazioni sul volume di input originale in modo da poter estrarre tali caratteristiche di

basso livello, che altrimenti andrebbero perse e sarebbe poi impossibile recuperarle nei livelli successivi. Quindi quello che si vuole è che il volume di output rimanga $32 \times 32 \times 3$. Per fare ciò, si applica uno spessore zero di dimensione 2 al primo livello. In altre parole, si riempie il volume di input di zeri attorno al bordo, in modo tale da ottenere così un volume di input $36 \times 36 \times 3$.

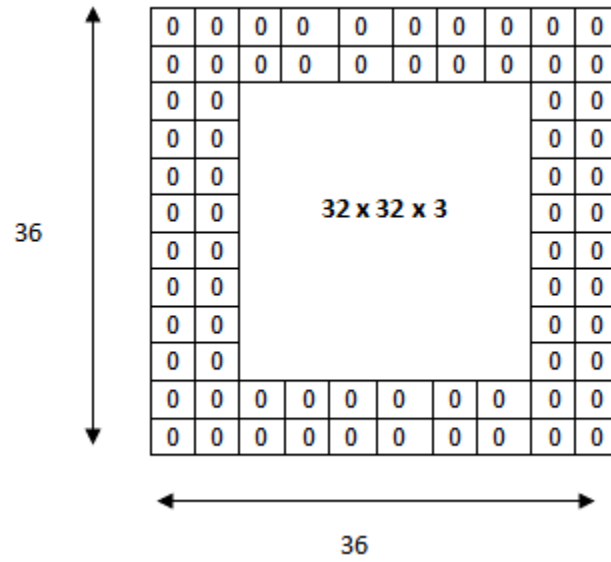


Figura 6.4: Rappresentazione dello zero-padding di dimensione 2.[30]

Per scegliere gli iperparametri di una rete neurale convoluzionale, non esiste uno standard stabilito che viene utilizzato da tutti i ricercatori, in quanto la rete dipende in gran parte dal tipo di dati a disposizione.

6.2 Rectified Linear Unit

Quando si attraversa un altro livello convoluzionale, l'output del primo livello convoluzionale diventa l'input del secondo livello. Di conseguenza, l'output del livello convoluzionale diventa l'input del livello ReLU, che solitamente è collocato subito dopo il livello convoluzionale. Il livello ReLU rappresenta un livello non lineare, il cui scopo è quello di introdurre la non linearità a un sistema che sostanzialmente sta calcolando operazioni lineari durante i livelli convoluzionali (tramite il prodotto scalare tra il filtro e il campo ricettivo). I ricercatori hanno scoperto che con questi livelli, le reti neurali convoluzionali funzionano molto meglio, perché la rete è in grado di allenarsi molto più velocemente (a causa dell'efficienza computazionale) senza impattare significativamente sull'accuratezza dei risultati. Il livello ReLU applica la funzione $f(x) = \max(0, x)$ a tutti i valori nel volume di input. In parole semplici, questo livello annulla tutti i valori negativi, aumentando le proprietà non lineari del modello e della rete globale senza influenzare i campi ricettivi del livello convoluzionale.

6.3 Pooling Layer

Dopo alcuni livelli ReLU, in genere si può scegliere se inserire il livello di pooling. Questo livello può essere eseguito secondo diversi risultati: il massimo è quello che è stato utilizzato per la rete neurale che verrà presentata nei prossimi capitoli, e per questo si considererà quello. Secondo questa classificazione il livello richiede un filtro (normalmente di dimensione 2×2) e un passo della stessa lunghezza. Quindi lo applica al volume di input del livello precedente e genera il numero massimo in ogni campo ricettivo attorno al quale il filtro ruota.

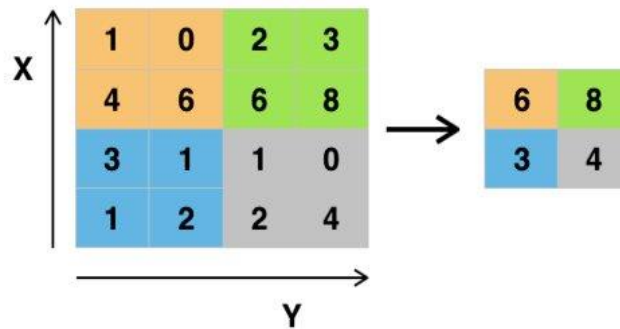


Figura 6.5: Esempio di applicazione del livello di pooling massimo.[30]

Ad esempio, per il primo riquadro in rosa della figura 6.5: 6 risulta il massimo tra il primo blocco (1, 0, 4 e 6), e così viene riportato nella prima posizione. E lo stesso per le altre 3. Il ragionamento intuitivo alla base di questo livello è che una volta che si sa che una caratteristica specifica è nel volume di input originale (ci sarà un alto valore di attivazione) e la sua posizione esatta non è importante quanto la sua posizione relativa rispetto alle altre caratteristiche. Questo livello riduce drasticamente la dimensione spaziale (l'altezza e la larghezza cambiano ma non la profondità) del volume di input ed i requisiti computazionali per i livelli futuri.

6.4 Fully Connected Layer

Solitamente è l'ultimo livello di una rete neurale convoluzionale. Questo livello prende fondamentalmente un volume di input (qualunque sia l'output del livello convoluzionale o del ReLU o del livello pool che lo precede) e genera un vettore N dimensionale in cui N è il numero di classi tra cui il programma deve scegliere. Ad esempio, se si desidera un programma, come in questo caso, di classificazione della valence composta da 3 classi, allora N sarà 3. Ogni numero in questo vettore di dimensione N rappresenta la probabilità di una certa classe. Se il vettore risultante per un programma di classificazione di cifre è:

[0.25 0.25 0.50]

allora questo rappresenta una probabilità del 25% che l'immagine dello spettrogramma appartenga alla classe 0 (valence-positiva), una probabilità del 25% che l'immagine appartenga alla classe 1 (valence-neutrale), una probabilità del 50% che l'immagine appartenga invece alla classe 2 (valence-negativa). Il modo in cui questo livello completamente connesso funziona è che guarda l'output del livello precedente (che dovrebbe rappresentare le mappe di attivazione di caratteristiche di alto livello) e determina quali caratteristiche sono maggiormente correlate a una particolare classe. Ad esempio, se il programma prevede che un'immagine sia un cane, avrà valori elevati nelle mappe di attivazione che rappresentano caratteristiche di alto livello come una zampa o 4 zampe, o il muso, e così via. Analogamente, se il programma prevede che un'immagine sia un uccello, avrà valori alti nelle mappe di attivazione che rappresentano caratteristiche di alto livello come le ali o un becco, e così via. Fondamentalmente, un livello FC guarda quali caratteristiche di alto livello sono maggiormente correlate ad una particolare classe e calcola i prodotti tra i pesi e il livello precedente per ottenere le probabilità corrette per le diverse classi.

6.5 Softmax Unit

Il livello softmax viene solitamente utilizzato come uscita dello strato completamente collegato. In questo livello viene utilizzata la funzione Softmax (esponenziale normalizzata) per rappresentare la distribuzione di probabilità delle k classi, nel nostro caso il numero delle classi sia per la valence che per l'arousal sono 3 quindi avremo k=3. La funzione è la seguente:

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum e^x}$$

6.6 Cross-Entropy

Per valutare la differenza che c'è tra le predizioni restituite dalla rete neurale e i valori reali si cerca di minimizzare una funzione di perdita. Esistono molte funzioni di perdita disponibili, in questo caso ho utilizzato la **categorical cross-entropy** della libreria Keras di Python, la quale è utilizzata nei problemi multiclasse. La funzione è la seguente:

$$D_{avg}(S(x), L) = -\frac{1}{N} \sum \sum_i L_i \log(\text{Softmax}(x_i))$$

ove la x rappresenta il vettore restituito dalla rete neurale che contiene le probabilità per ciascuna classe, mentre L rappresenta il vettore (in codifica one-hot) delle classi reali dove solamente uno dei suoi elementi avrà valore 1. L'obiettivo durante la fase di allenamento è quello di trovare il miglior set di pesi e bias in grado di minimizzare questa funzione, per fare ciò in ogni epoca si eseguono questi due passaggi (già spiegati nella sezione 4.4):

- **Calcolo dell'output previsto \hat{y} , noto come feedforward**
- **Aggiornamento dei pesi e dei biases, noto come backpropagation**

6.7 Dropout Layer

Le reti neurali, durante il loro processo di apprendimento, sfruttano, come spiegato nella sezione 6.6 la funzione di perdita per decidere come sistemare i propri parametri, ovvero pesi e bias. Un altro grande problema è quello dell'overfitting. Sono state sviluppate alcune tecniche che, operando sulla funzione costo, aiutano a ridurre gli effetti del sovrallenamento di una rete neurale. Tali tecniche prendono il nome di **tecniche di regolarizzazione**. Una delle più utilizzate è la tecnica di dropout. Questa tecnica non

modifica la funzione di costo ma bensì la rete stessa. Essa per ogni epoca di allenamento sceglie (casualmente) quali neuroni tenere e quali scartare continuando ad addestrare in questo modo la rete così ottenuta. Si ripete quindi il procedimento, tenendo e scartando neuroni diversi ad ogni epoca: una volta che si ritiene che la rete sia pronta, si prende la rete originale e si aggiustano i pesi uscenti dai neuroni nascosti: abbiamo ottenuto una rete pronta a svolgere il proprio compito. In poche parole, è come se venissero utilizzate tante reti diverse e poi si prendesse come risultato la media di tutti i risultati di queste reti. Va tenuto ben presente che questo procedimento è applicato solo in fase di allenamento: durante il funzionamento vero e proprio, la rete è considerata nella sua interezza.

6.8 Data Augmentation

L'aumento dati è una tecnica molto potente utilizzata per creare artificialmente variazioni nei dati esistenti (features, immagini, ecc..), per espandere un set di dati con l'aggiunta di nuove informazioni che permette di rappresentare meglio il dominio su cui un algoritmo di machine learning o deep learning dovrà addestrarsi. Le reti neurali convoluzionali (CNN) ad esempio, necessitano di un numero enorme di immagini per una formazione efficace del modello. **Ciò aiuta ad aumentare le prestazioni del modello generalizzando meglio e riducendo così l'overfitting.** I set di dati più diffusi per gli algoritmi di classificazione e il rilevamento di oggetti contengono da alcune migliaia a milioni di immagini. **Una CNN, grazie alla sua proprietà di varianza, può classificare gli oggetti anche quando sono visibili in diverse dimensioni, orientamenti o differenti illuminazioni.** Quindi, si può prendere il piccolo set di dati di immagini e trasformare gli oggetti in dimensioni diverse ingrandendoli o rimpicciolendoli, ruotandoli verticalmente o orizzontalmente, modificando la luminosità qualunque cosa abbia senso per l'oggetto. In questo modo si crea un set di dati ricco e diversificato con variazioni. In questa tesi è stata utilizzata la tecnica di **aggiunta del rumore Gaussiano** ai file audio che compongono il dataset bilanciato, in modo da poter avere più esempi a disposizione su cui addestrare la SVM e la CNN, in modo tale che soprattutto quest'ultima riesca a generalizzare meglio anche su dati rumorosi. Il rumore gaussiano, o rumore bianco, ha

una media 0 e una deviazione standard di 1, e può essere generato secondo necessità usando un generatore di numeri pseudocasuali.

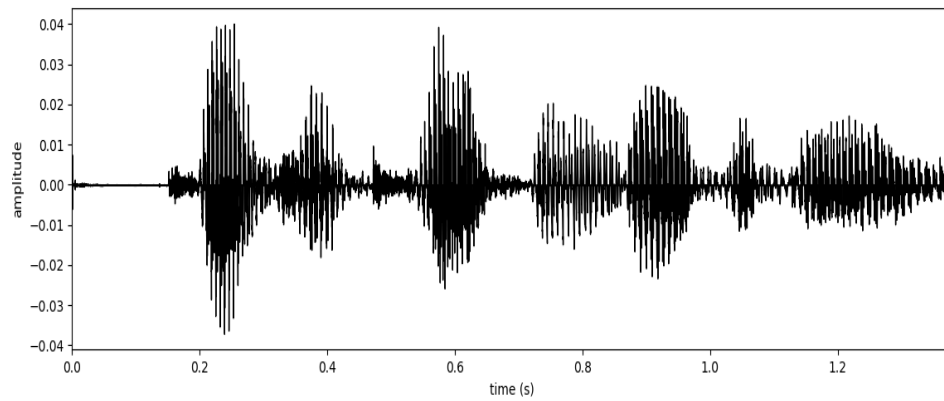


Figura 6.6: Rappresentazione di un file audio di neutrale di Ravdess senza aggiunta di rumore Gaussiano.

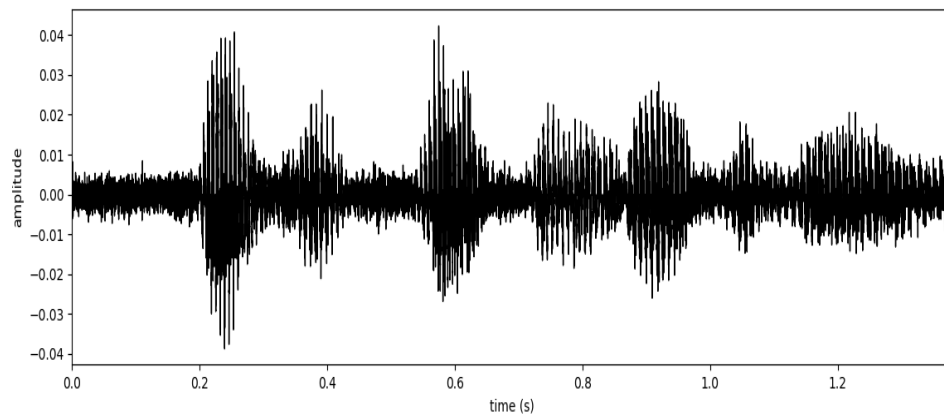


Figura 6.7: Rappresentazione dello stesso file audio della figura 6.6 ma non l'aggiunta di rumore Gaussiano.

6.9 Spettrogramma

Uno spettrogramma è la rappresentazione grafica dell'intensità di un suono in funzione del tempo e della frequenza. Di solito gli spettrogrammi sono rappresentati nel modo seguente:

- Sull'asse delle ascisse è riportato il tempo in scala lineare
- Sull'asse delle ordinate è riportata la frequenza in scala lineare o logaritmica
- A ciascun punto di data ascissa e data ordinata è assegnata una tonalità di grigio, o un colore, rappresentante l'intensità del suono in un dato istante di tempo e a una data frequenza; la relazione fra l'intensità del suono e la scala di grigi o di colori può essere lineare o logaritmica.

I filtri di convoluzione 2D della CNN aiutano a catturare mappe caratteristiche in due dimensioni per ogni esempio di input. Funzioni così ricche non possono essere estratte e applicate quando il discorso viene convertito in testo e/o fonemi. Gli spettrogrammi, che contengono informazioni extra non disponibili solo nel testo, offrono ulteriori capacità nei tentativi di migliorare il riconoscimento delle emozioni. La CNN utilizzata in questo lavoro di tesi utilizza il Mel-Frequency Spectrogram che praticamente è come uno spettrogramma normale solamente che sull'asse y la frequenza è riportata in scala Mel. Avere l'asse delle frequenze in scala Mel pone maggiore attenzione sull'estremità inferiore dello spettro delle frequenze rispetto a quello superiore, imitando così le capacità percettive dell'udito degli umani. Uno spettrogramma si ottiene, di solito, suddividendo l'intervallo di tempo totale (cioè quello relativo all'intera forma d'onda da analizzare) in sotto intervalli uguali (detti finestre temporali) di durata da 5 a 10 ms e calcolando la trasformata di Fourier della parte di forma d'onda contenuta in ciascuna finestra (solitamente si usa la trasformata veloce di Fourier, o FFT), che fornisce l'intensità del suono in funzione della frequenza. Le trasformate di Fourier, relative alle diverse finestre temporali, vengono poi assemblate a formare lo spettrogramma.

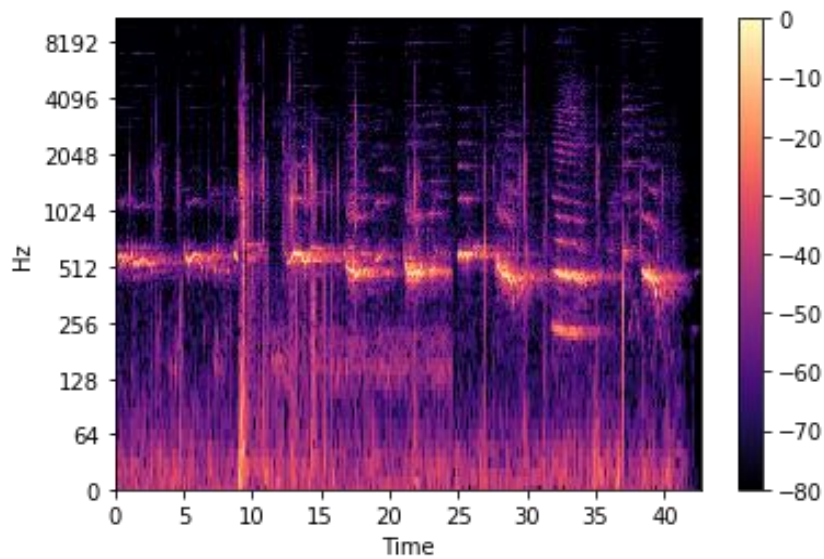


Figura 6.8: Rappresentazione di un Mel-Frequency Spectrogram.[31]

6.10 Architettura

L'architettura di base della rete neurale profonda implementata in questo studio era una rete che veniva utilizzata per il riconoscimento di 10 suoni ambientali attraverso lo spettrogramma in scala Mel. Tuttavia, a seguito di numerosi test eseguiti anche con altre reti neurali i risultati ottenuti con questa architettura si sono rivelati essere i migliori.

L'architettura della rete è composta da 3 strati convoluzionali 2D, i primi due sono seguiti da uno strato di pooling massimo e successivamente dalla funzione di attivazione ReLU mentre il terzo non ha lo strato di pooling ma è seguito direttamente dalla stessa funzione di attivazione appena citata. Dopodichè è presente uno strato completamente connesso che permette di appiattire i risultati in un'unica dimensione, successivamente è presente un livello di dropout con $p = 0.5$. Successivamente è presente uno strato di 64 neuroni seguiti da una funzione di attivazione ReLU e poi da un livello di dropout con $p = 0.5$. Infine è collocato un ultimo strato di 3 neuroni (ognuno per ciascuna classe sia della valence che dell'arousal) seguito dalla funzione di attivazione Softmax per effettuare la distribuzione di probabilità delle classi. L'unica modifica che è stata eseguita su questa

rete, è stato cambiare il numero di neuroni dello strato finale poiché nella versione originale ne erano 10. L'ottimizzatore utilizzato per addestrare la rete è Adam.

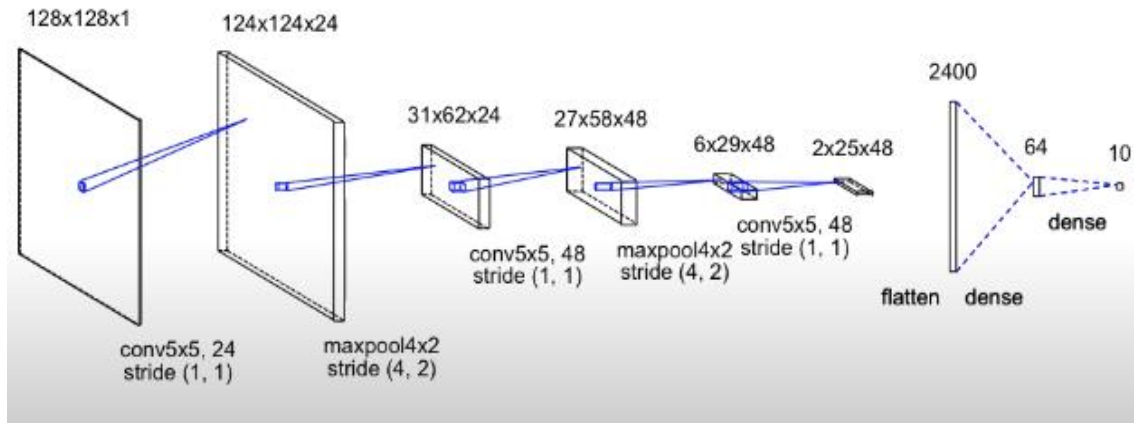


Figura 6.9: Rappresentazione complessiva della CNN spiegata pocanzi.[17]

6.11 Pre-processing

Ogni audio utilizzato negli esperimenti con gli spettrogrammi è stato pre-elaborato come spiegato già nella sezione 2.1.

Utilizzando diversi dataset uno dei problemi principali che si ha è la differenza in lunghezza dei file audio. Per avere spettrogrammi tutti della stessa lunghezza è stata applicata la tecnica dello zero-padding. Questo metodo consiste nell'estrarre prima lo spettrogramma da ogni file audio in base alla propria lunghezza, e successivamente aggiungere degli zeri in coda qualora l'audio fosse troppo corto e non sia quindi possibile estrapolare il numero necessario di frames. Quindi in questo caso è stata suddivisa la banda delle frequenze in 128 componenti e sono stati estratti tutti i frames possibili con un frame-length di 23 ms, ottenendo così una matrice $(128, x)$, successivamente qualora x (numero frames) fosse minore di 128 allora vengono aggiunti gli zeri fino a raggiungere la dimensione completa $(128, 128)$. 128 frames, ciascuno di lunghezza 23 ms, corrisponde

ad un audio di circa 3 secondi. Se invece x fosse maggiore di 128 allora vengono presi solamente i primi 128 frames scartando tutti gli altri.

6.12 Risultati ottenuti

I risultati presentati in questa sezione fanno riferimento al dataset spiegato nella sezione 5.8.

Il modello di rete neurale è stato valutato prima partizionando il dataset in train set e test set (75% per il train e 25% per il test) per riuscire ad individuare il numero di epoche di addestramento necessarie per non far andare il modello in overfitting.

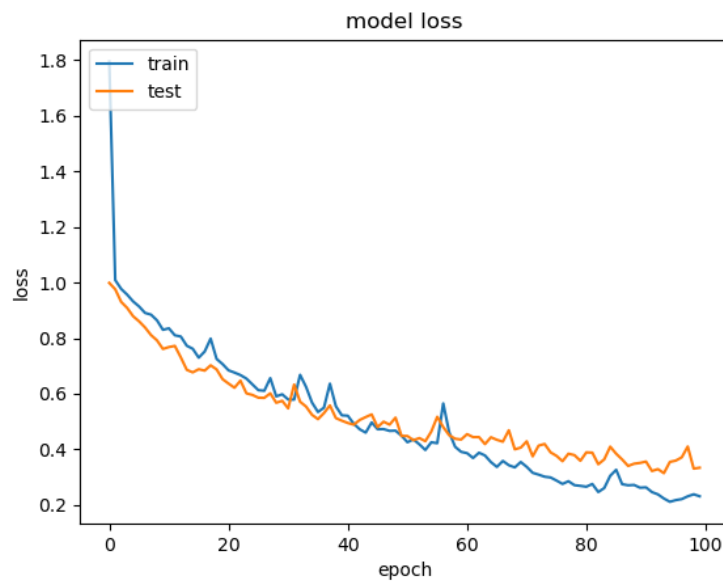


Figura 6.10: Rappresentazione della funzione di costo per la valence

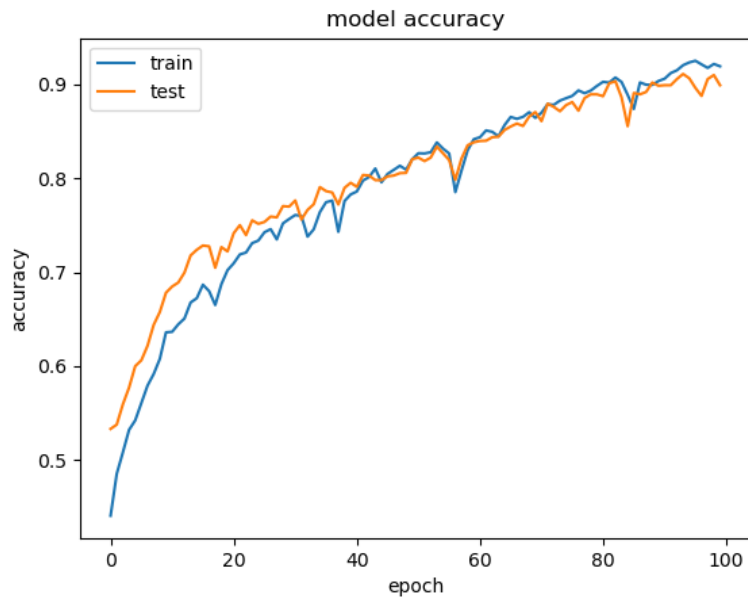


Figura 6.11: Rappresentazione dei valori di accuratezza per la valence

Come si può notare dalla figura 6.10, dalle 60 epoche in poi le due curve tendono ad allontanarsi, ciò sta a significare che il modello da quel punto in poi sta andando in overfitting, per queste ragioni per la valence il numero delle epoche di addestramento per eseguire la cross validation stratified è stato fissato a 60.

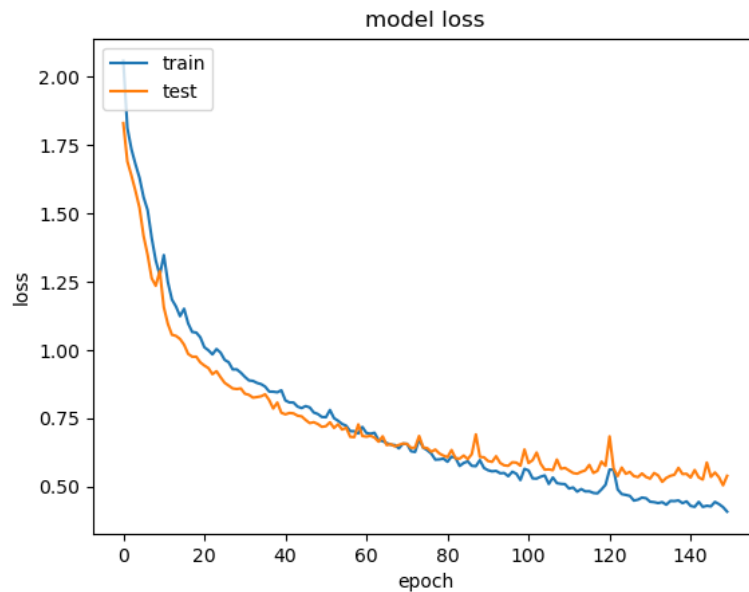


Figura 6.12: Rappresentazione della funzione di costo per l'arousal

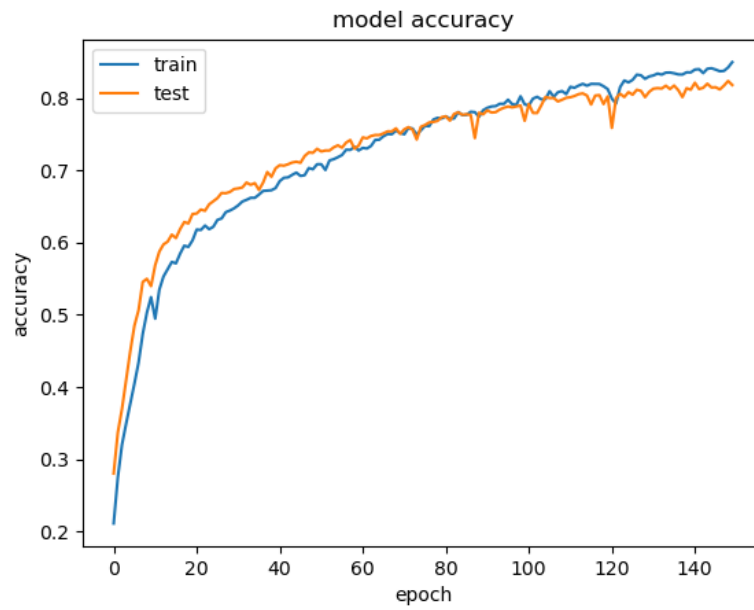


Figura 6.13: *Rappresentazione dei valori di accuratezza per l'arousal*

In questo caso, osservando la figura 6.12, si può notare come dopo le 100 epoche il modello inizi a soffrire del sovradattamento, per queste ragioni il numero di epoche su cui addestrare il modello per eseguire la cross validation stratified sull'arousal è 100.

	VALENCE	AROUSAL
CNN	0.80%	0.87%

Tabella 6.1: *Risultati ottenuti dalla Cross validation stratified sia per la valence che per l'arousal sul dataset bilanciato.*

Capitolo 7 - Sperimentazione finale in the wild

In questo capitolo verranno presentati i risultati ottenuti su un piccolo dataset formato da 209 file audio da me estratti, direttamente da film.

7.1 Risultati ottenuti

Ogni file audio è stato campionato ad una frequenza di 16Khz ed è stato rimosso il silenzio. Ogni file di test è stato etichettato (ovvero per ognuno di essi, ogni utente, in maniera autonoma, diceva quale emozione, tra le 7 definite nell'introduzione di questa tesi, rappresentasse per lui) da 7 persone diverse. Poiché su alcuni audio erano presenti forti discrepanze tra gli utenti, allora questi file sono stati eliminati dal dataset. Gli audio complessivi ritenuti utili sono 209. Essi sono distribuiti nel seguente modo:

Emozione	Numero esempi
Disgusto	30
Gioia	30
Neutrale	29
Paura	30
Rabbia	38
Sorpresa	22
Tristezza	30

Modelli	Valence	Arousal
SVM	0.51%	0.63%
CNN	0.64%	0.59%

Tabella 7: Risultati finali ottenuti sia per la valence che per l'arousal sul dataset di 202 file audio.




Osservando la tabella 7 si può notare come la CNN abbia incrementato la percentuale di accuratezza sulla valence **del 13%**, mentre la SVM rimane ancora il modello migliore per quanto riguarda l'arousal, con un incremento del 4% rispetto alla CNN. Per queste ragioni nell'applicazione EmoFEATURES è stato utilizzato il modello CNN per riconoscere la valence e la SVM per il riconoscimento dell'arousal. Come si può vedere i valori di accuratezza ottenuti non sono alti, questo perché il dataset è formato da file audio registrati in un ambiente non controllato e senza apparecchiature professionali, senza considerare inoltre, tutti gli altri problemi dovuti all'ambito del riconoscimento delle emozioni dalla voce, già spiegate nell'introduzione di questa tesi.

Capitolo 8 - Applicazione EMOFEATURES per Windows

In questo capitolo verrà presentata l'applicazione chiamata EmoFEATURES che permette di eseguire il riconoscimento delle emozioni: solo voce, solo volto e voce e volto insieme.

8.1 Interfaccia grafica

All'avvio, il software presenta una schermata semplice ed essenziale che, oltre a mostrare il titolo del programma, evidenzia anche le possibili scelte effettuabili dagli utenti. Come mostrato nella figura in basso, ciascuno di questi può scegliere tra 3 diversi tipi di analisi delle emozioni:

-  dalla voce
-  dal volto
-  dalla voce e dal volto contemporaneamente

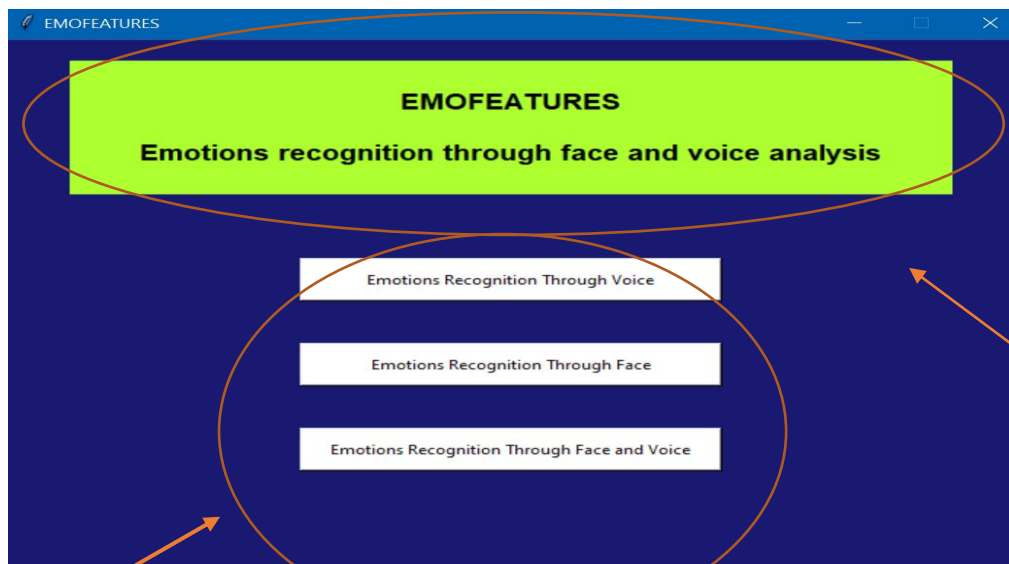


Figura 8: Schermata principale dell'interfaccia.

Possibili
Scelte
per
l'utente

Dopo aver selezionato una scelta, l'utente verrà indirizzato alla rispettiva nuova finestra.

1 Emotions Recognition through Voice

Nel caso in cui l'utente scelga la prima opzione ("emotions recognition through voice") della finestra principale, comparirà una schermata, come mostrato nella figura in basso, dove l'utente potrà effettuare un'analisi della sua voce per rilevare il proprio stato d'animo. Il tasto "Start" consentirà l'acquisizione vocale; una volta cliccato, apparirà al suo posto un tasto "Stop" per terminare la registrazione. Nel box situato più in basso, apparirà il risultato dell'analisi della voce (es. "happy"). Terminata questa fase, l'utente, cliccando sul tasto "Back" tornerà alla schermata iniziale.

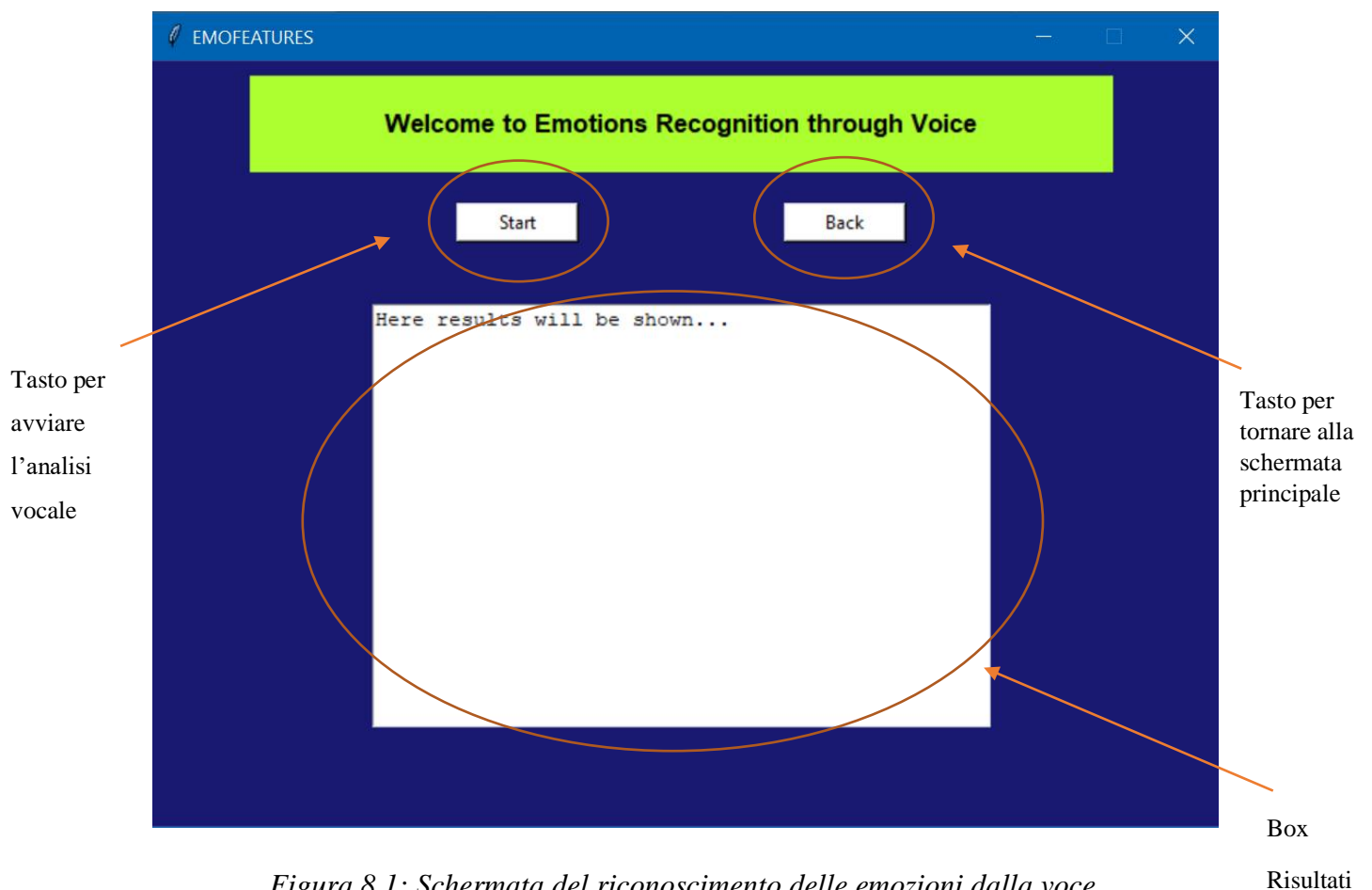


Figura 8.1: Schermata del riconoscimento delle emozioni dalla voce.

2 Emotions Recognition through Face

Nel caso in cui l'utente scelga la seconda opzione ("emotions recognition through face") della finestra principale, comparirà una schermata, come mostrato nella figura in basso, dove l'utente potrà effettuare un'analisi del proprio volto per rilevare lo stato d'animo. Il tasto "Start" consentirà l'acquisizione video; dopo aver avviato il riconoscimento facciale, apparirà una seconda finestra che mostrerà la webcam. Nel box situato più in basso, apparirà un suggerimento per la chiusura della videata della webcam. Terminata questa fase, l'utente, cliccando sul tasto "Back" tornerà alla schermata iniziale.

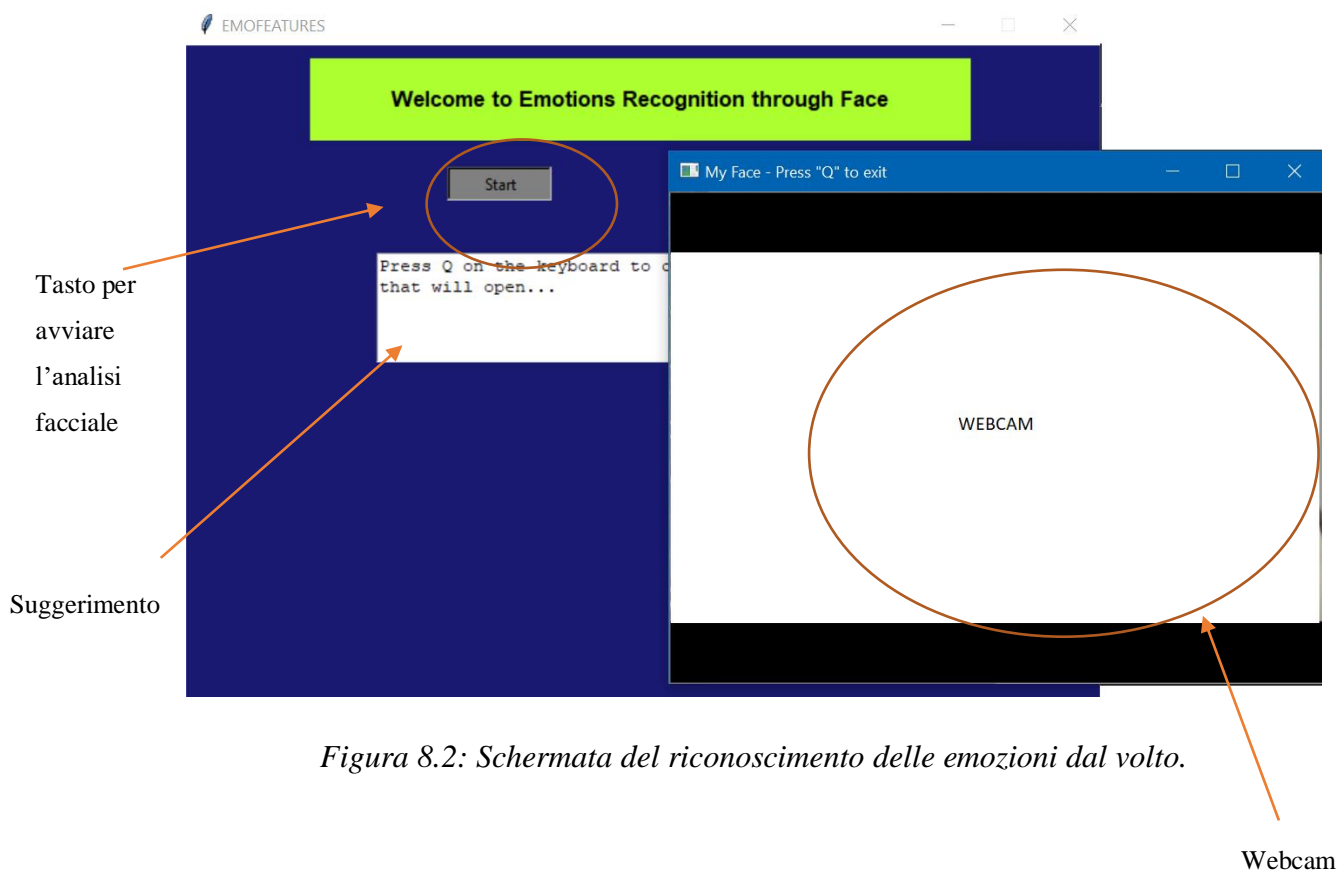


Figura 8.2: Schermata del riconoscimento delle emozioni dal volto.

3 Emotions Recognition through Face and Voice

Nel caso in cui l'utente scelga la terza opzione ("emotions recognition through face and voice") della finestra principale, comparirà una schermata, come mostrato nella figura in basso, dove l'utente potrà effettuare un'analisi del proprio volto e della propria voce contemporaneamente. Il tasto "Start" consentirà l'acquisizione audio-video; dopo aver avviato il riconoscimento, apparirà una seconda finestra che mostrerà la webcam. Nel box situato più in basso, sarà presente un suggerimento per la chiusura della videata della webcam e appariranno i risultati del riconoscimento. Terminata questa fase, l'utente, cliccando sul tasto "Back" tornerà alla schermata iniziale.



Figura 8.3: Schermata del riconoscimento delle emozioni dalla voce e dal volto Webcam contemporaneamente.

Chiusura del programma

Dopo aver terminato le operazioni, l'utente avrà la possibilità di chiudere il programma semplicemente cliccando sul tasto "X" situato in alto a destra della finestra. Apparirà, dunque, il messaggio di conferma chiusura e l'utente potrà decidere di abbandonare definitivamente la sessione in corso cliccando il tasto "Sì" oppure proseguire cliccando il tasto "No".

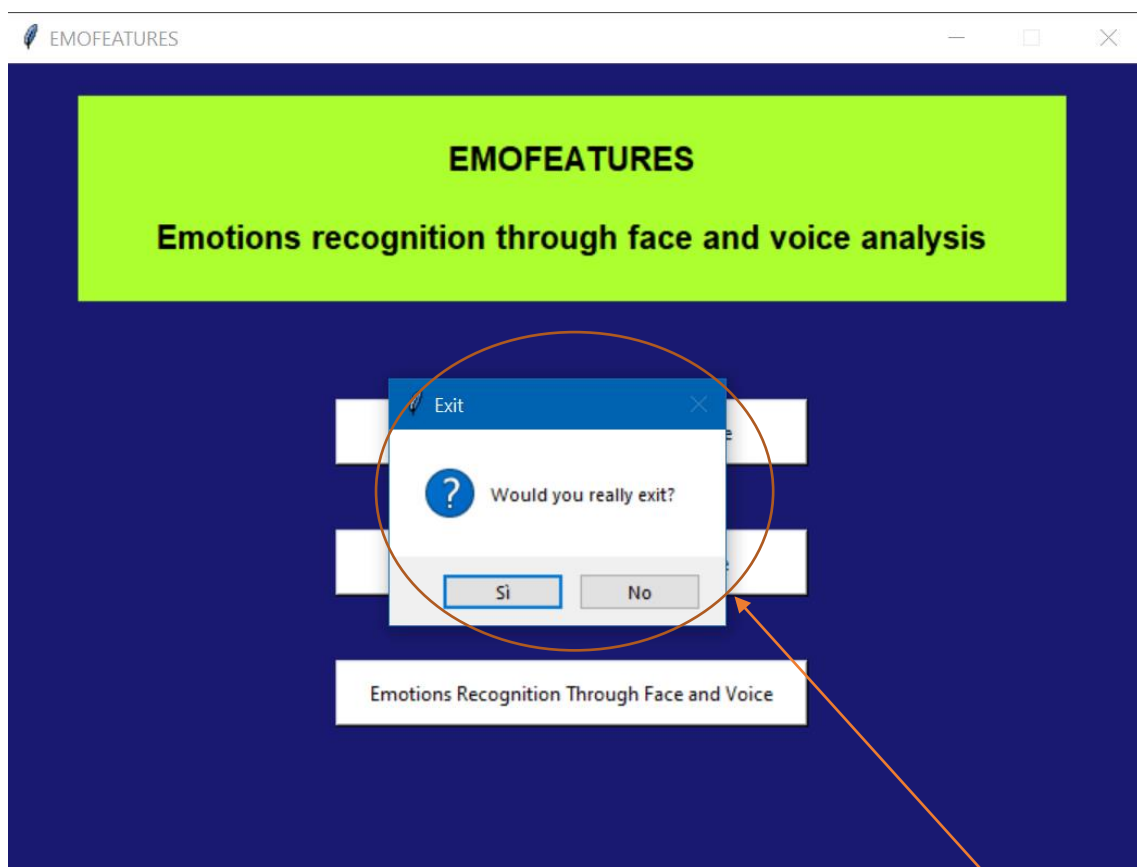


Figura 8.4: Schermata di chiusura dell'interfaccia.

Finestra di
Conferma della chiusura

8.2 Esempi di funzionamento del programma

Riconoscimento delle emozioni attraverso l'analisi della voce:

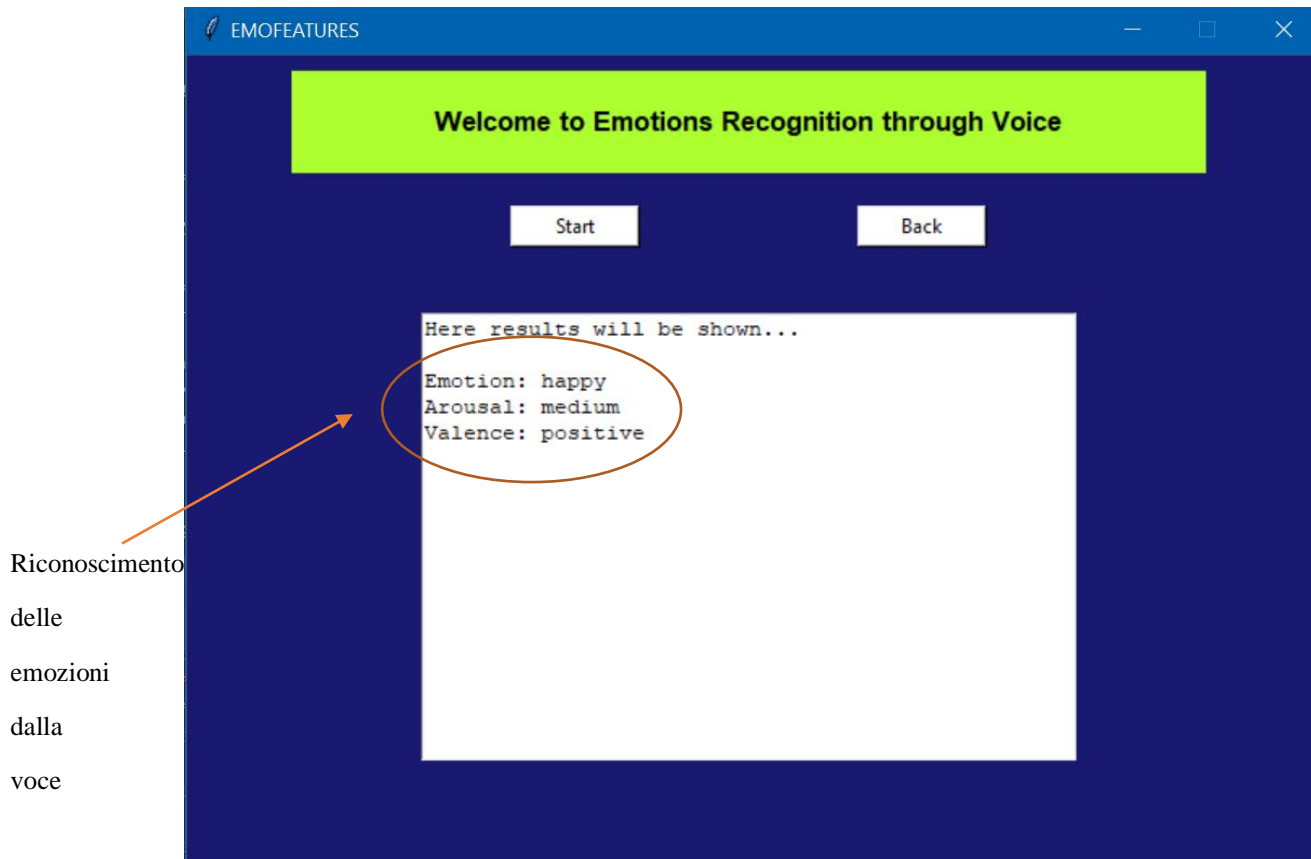


Figura 8.5: Esempio di funzionamento della funzione

“Emotions recognition through voice”

Riconoscimento delle emozioni attraverso l'analisi del volto:

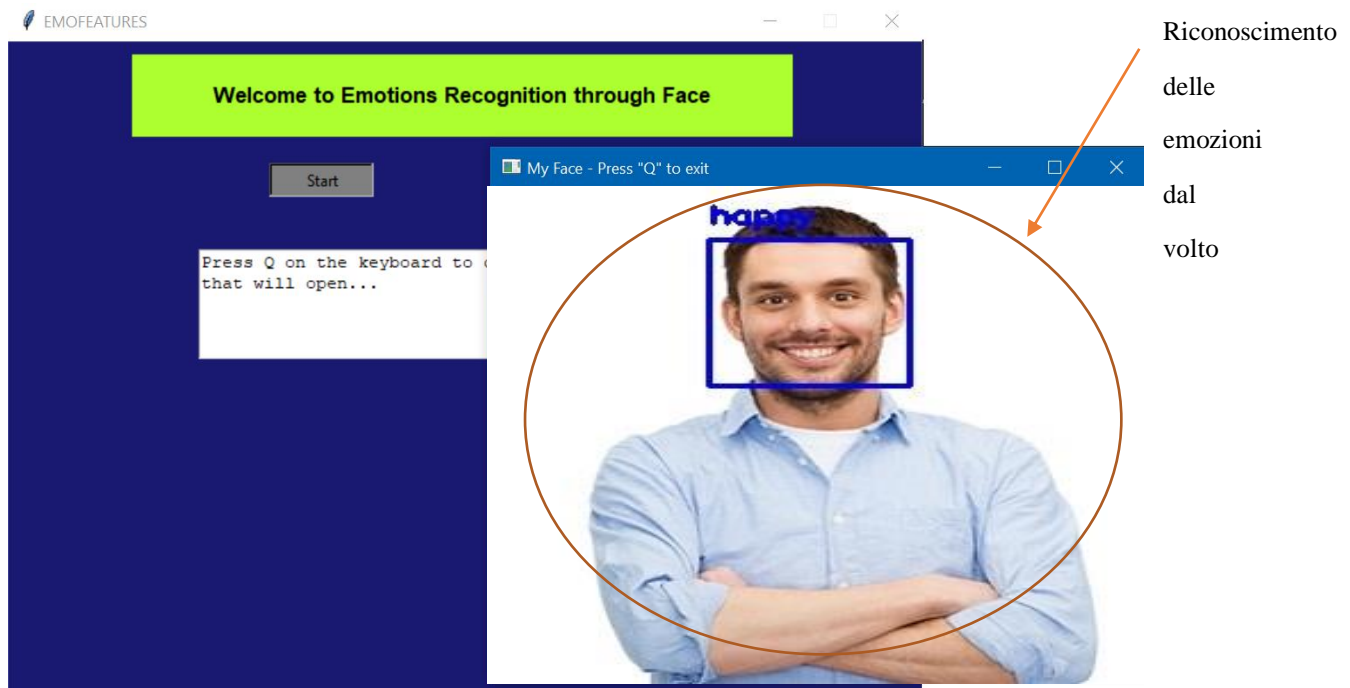


Figura 8.6: esempio di funzionamento della funzione

“Emotions recognition through face”

Riconoscimento delle emozioni attraverso l'analisi della voce e del volto:

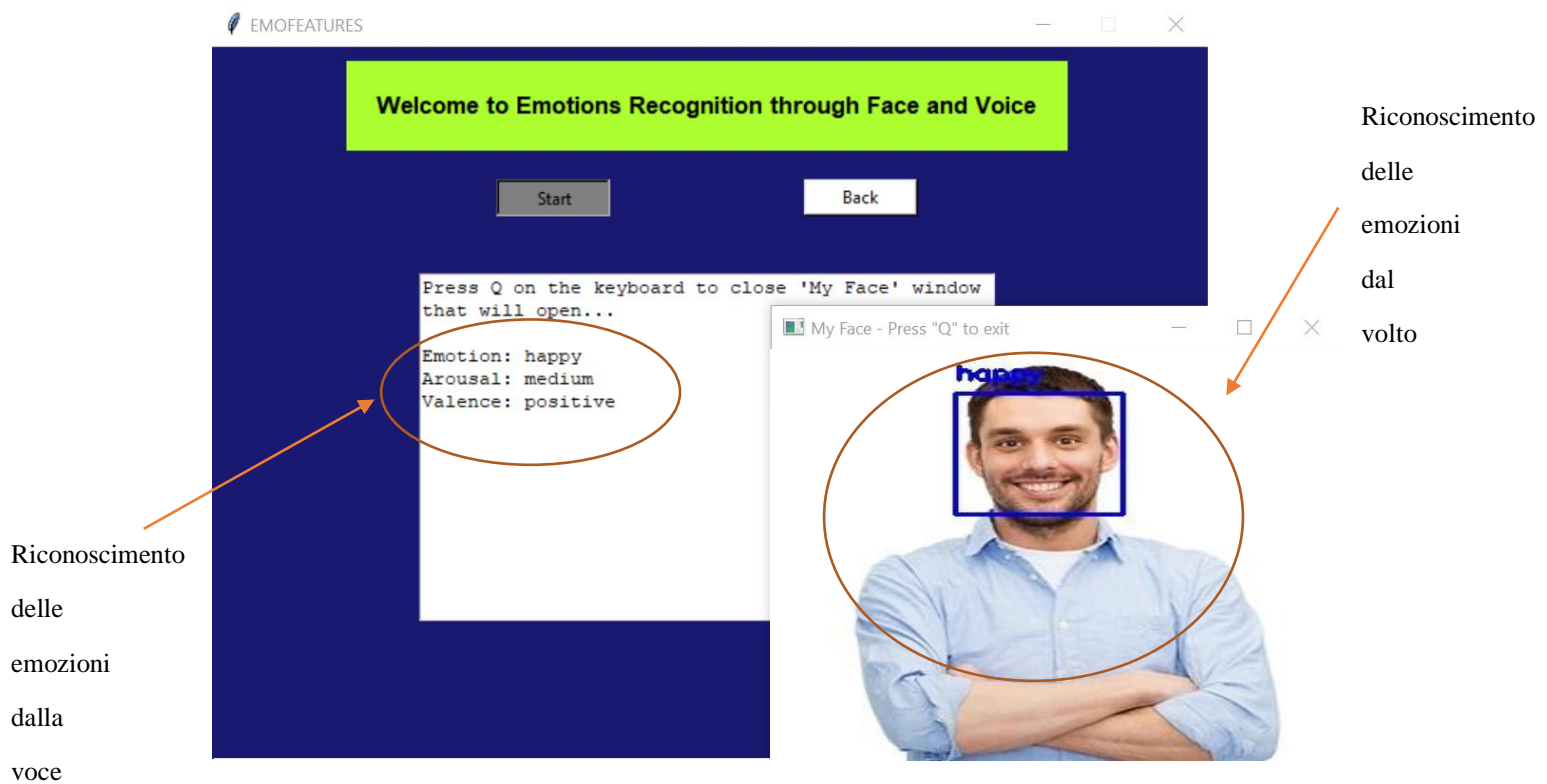


Figura 8.7: esempio di funzionamento della funzione

"Emotions recognition through face and voice"

Capitolo 9 - Conclusioni e sviluppi futuri

In questo lavoro di tesi sono stati utilizzati e testati diversi dataset per addestrare e comparare tra loro diversi classificatori (utilizzando particolari features mirate per il riconoscimento delle emozioni) e successivamente è stata addestrata una CNN avente in input spettrogrammi con frequenza in scala Mel di dimensione 128x128 pixel. Dall'analisi dei risultati il miglior classificatore si è rivelata essere la SVM riuscendo a superare in termini di accuratezza per la CVS (cross-validation stratified) tutti gli altri classificatori in quasi ogni test eseguito sia sulla valence che sull'arousal. Per quanto riguarda la CNN invece, anche se raggiungeva un'accuratezza minore (sul dataset bilanciato) con la CVS rispetto alla SVM, sia sulla valence che per l'arousal, durante la fase di sperimentazione in the wild si è rivelata essere addirittura superiore in termini di accuratezza sulla valence (0.62%) rispetto alla SVM (0.51%) con un incremento del 13% percento. Questo mette in mostra le grandi capacità di generalizzazione della CNN che l'hanno resa celebre soprattutto negli ultimi anni grazie ai miglioramenti che ci sono stati per quanto riguarda l'hardware. Per l'arousal invece, la CNN si posiziona poco dopo la SVM con un'accuratezza del 59% contro il 63%. Questo risultato può essere considerato comunque abbastanza buono perché permette di poter dire che la rete neurale convoluzionale in alcuni casi (come questo) o riesce a raggiungere praticamente quasi lo stesso livello di generalizzazione del classificatore migliore sulle emozioni (SVM), senza essere troppo inferiore in termini di accuratezza, oppure, come nel caso della valence, riesce a migliorare notevolmente i risultati. Sicuramente un grande contributo ad entrambi i modelli è stato dato dalla tecnica di data augmentation che ha permesso di aumentare considerevolmente i dati e quindi permettere ad essi una migliore generalizzazione. Un'ulteriore considerazione deve essere fatta: come si può notare dai risultati ottenuti dalla CVS i due modelli riescono a raggiungere ottimi risultati quando vengono testati sui dataset preparati in condizioni ambientali ottimali. Questo accade anche grazie ad una strumentazione di registrazione audio molto più avanzata rispetto ad un semplice microfono di un pc o di un cellulare. Come infatti già detto nell'introduzione, i dispositivi audio giocano un ruolo fondamentale per quanto riguarda l'ambito del riconoscimento

delle emozioni dalla voce. Per quanto riguarda i risultati ottenuti sul piccolo dataset da me creato, si capisce come le CNN possano rappresentare a mio avviso il futuro per quanto riguarda questo ambito, anche se quest'ultimo, risulta essere molto complesso e variabile, e quindi difficilmente generalizzabile, visto che il modo di esprimere un'emozione risulta essere molto soggettiva e dipendente fortemente dal genere (maschile,femminile), dall'etnia e fascia d'età. Questo fatto apre molte strade su possibili modi alternativi di affrontare il problema: potrebbe essere infatti testato l'eventuale miglioramento che potrebbe esserci dividendo i dataset principali per genere, razza ed età (oppure solamente in base al primo) cercando di avere comunque un buon numero di campioni. In questo caso sarebbe interessante notare il comportamento della CNN rispetto alla SVM su queste configurazioni. Questi potrebbero essere dei possibili sviluppi futuri legati al lavoro svolto, insieme ad una validazione più approfondita eseguita utilizzando altri dataset.

Ringraziamenti

In questa parte vorrei ringraziare tutte quelle persone che mi sono state vicino durante questi tre anni. Ringrazio in primis la mia relatrice la prof. De Carolis per avermi accompagnato durante la sperimentazione e la stesura di questa tesi. Un ringraziamento particolare va a tutta la mia famiglia: mio padre, mio modello di vita, per avermi sempre incoraggiato nei momenti più difficili che ho dovuto affrontare in questo percorso, a mia madre, per avermi guidato e confortato in ogni momento di difficoltà, a mia sorella Maria Chiara, mia migliore amica, per l'affetto e i consigli datomi, e a tutti i miei fratelli Antonio, Giovanna e Serena per essere riusciti, in alcuni momenti, a farmi distrarre e divertire in questo lungo periodo di preparazione di questa tesi. Vorrei ringraziare anche tutti i miei zii e i miei cugini per avermi insegnato i valori fondamentali della famiglia, che non si limita al solo nucleo familiare ma è qualcosa di molto più esteso. Ringrazio la donna della mia vita Agnese, per avermi supportato e soprattutto sopportato in questi tre anni, per avermi reso un uomo migliore, per avermi dato preziosi consigli e affetto nei periodi più complicati da affrontare. Un ringraziamento particolare va a tutti i miei nonni: Dante, Clotilde, Michele e Teresa che con il loro spirito mi sono stati sempre vicino in

ogni momento di difficoltà, di paura, di ansia dovuti agli esami e alla preparazione di questa tesi, seppur lontani hanno saputo farsi sentire, senza farmi mai sentire solo. Ringrazio i miei compagni di viaggio di questi tre anni: Claudio (con il quale ho avuto il piacere di condividere il lavoro sulla combinazione tra voce e volto), Gianfranco, Pietro, Nicola e Matteo perché grazie a loro il peso degli esami e dello studio intenso sono stati più leggeri e spensierati. Ringrazio i miei coinquilini e amici veri: Cosimo, Benedetto, Gabriele, Fabio e Alessandro per avermi regalati giorni di risate e divertimento anche solo rimanendo in casa o prendendo un caffè. Volevo ringraziare inoltre, i miei amici di una vita, Alessio e Antonio, per aver creduto in me sin da l'inizio e per avermi regalato momenti di rilassatezza e risate durante tutti questi tre anni. Ringrazio il mio amico Germano per aver sopportato la mia ansia durante la preparazione di quasi ogni singolo esame di questa triennale. Ringrazio infine, tutti i miei amici che mi sono stati vicino, non solo in questi tre anni, per avermi regalato giorni di festa e divertimento indimenticabili.

Bibliografia

- [1] <https://zenodo.org/record/1188976#.Xqbe22j7TIU>
- [2] <https://www.kaggle.com/ejlok1/toronto-emotional-speech-set-tess>
- [3] <http://kahlan.eps.surrey.ac.uk/savee/>
- [4] EMOVO Corpus: an Italian Emotional Speech Database Giovanni Costantini^{1,2}, Iacopo Iadarola³, Andrea Paoloni³, Massimiliano Todisco^{1,3} ¹ Department of Electronic Engineering, University of Rome “Tor Vergata”, Rome, Italy ² Institute of Acoustics “O. M. Corbino”, Rome, Italy ³ Fondazione “Ugo Bordon”, Rome, Italy.
- [5] Emilia Parada-Cabaleiro, Giovanni Costantini, Anton Batliner, Alice Baird e Björn Schuller (2018), Percezione categorica vs dimensionale del discorso emotivo italiano , in Proc. di Interspeech, Hyderabad, India, pagg. 3638-3642.
- [6] <http://emodb.bilderbar.info/index-1024.html>
- [7] <https://zenodo.org/record/2544829#.XqboTmj7TIU>
- [8] <https://online-audio-converter.com/it/>

- [9] [https://it.wikipedia.org/wiki/Coclea_\(anatomia\)](https://it.wikipedia.org/wiki/Coclea_(anatomia))
- [10] https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-73003-5_775
- [11] <https://builtin.com/data-science/random-forest-algorithm>
- [12] <https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/>
- [13] <https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab>
- [14] <https://scikitlearn.org/stable/modules/generated/sklearn.svm.NuSVC.html#sklearn.svm.NuSVC>
- [15] <https://lorenzogovoni.com/support-vector-machine/>
- [16] https://it.wikipedia.org/wiki/K-nearest_neighbors
- [17] Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification, Justin Salamon and Juan Pablo Bello
- [18] <https://lorenzogovoni.com/architettura-di-rete-neurale-convoluzionale/>
- [19] <https://www.thebigsmoke.com.au/>
- [20] <https://opentextbc.ca/introductiontopsychology/chapter/10-1-the-experience-of-emotion/>
- [21] <https://it.cleanpng.com/cleanpng-7jwlha/>
- [22] <https://www.thestreamart.com/il-progetto/>
- [23] <http://tesi.cab.unipd.it/48771/1/tesiBressan.pdf>
- [24] https://it.wikipedia.org/wiki/Scala_mel
- [25] http://Sicurezza/04_SPEAKER-20VERIFICATION-20-20NEW.pdf
- [26] <https://builtin.com/data-science/random-forest-algorithm>
- [27] <https://www.pinterest.it/pin/192388215317840422/>
- [28] <https://www.quora.com/What-is-a-hyperplane-in-machine-learning>
- [29] <https://stackabuse.com/introduction-to-neural-networks-with-scikit-learn/>
- [30] <https://lorenzogovoni.com/architettura-di-rete-neurale-convoluzionale/>
- [31] Deep Learning based Emotion Recognition System Using Speech Features and Transcriptions, Suraj Tripathi¹, Abhay Kumar¹, Abhiram Ramesh¹, Chirag Singh¹, Promod Yenigalla¹
- [32] “Six Basic Emotions,” *Management Mania*, 2016. [Online]. Available: <https://managementmania.com/en/six-basic-emotions>. [Accessed: 27-Feb-2020].

- [33] R. Firth-Godbehere, “AI believes we express emotions the same six ways – That is a problem @ www.thebigsmoke.com.au,” 2019. [Online]. Available: <https://www.thebigsmoke.com.au/2019/04/28/ai-believes-we-express-emotions-same-six-ways-that-problem-emotion/>.
- [34] Wikipedia, “Robert Plutchik,” 2016. [Online]. Available: https://it.wikipedia.org/wiki/Robert_Plutchik. [Accessed: 27-Feb-2020].
- [35] K. Wiggers, “Amazon’s AI improves emotion detection in voices @ venturebeat.com,” 2019. [Online]. Available: <https://venturebeat.com/2019/05/21/amazons-ai-improves-emotion-detection-in-voices/>. [Accessed: 27-Feb-2020].