Dipartimento di Ingegneria e Scienza dell'Informazione

Corso di Laurea in
Informatica

Elaborato finale

Comparazione ed analisi di vari metodi di classificazione delle emozioni umane da sorgente audio

Supervisore	Co-Supervisore	Laureando
Prof. Elisa Ricci	Riccardo Franceschini	Thomas Rigoni

Anno accademico 2020/2021

Ringraziamenti

Elisa Ricci per questa opportunità
Riccardo Franceschini per il supporto e i preziosi consigli

Indice

Sommario

Il riconoscimento delle emozioni umane da sorgente audio è un argomento di ricerca aperto da molti anni e ultimamente ha subito una forte accelerazione dovuta al crescente interesse in questo ambito, supportata dai recenti avanzamenti nell'ambito del Deep Learning.

Questo elaborato si inserisce nel contesto del progetto europeo SPRING [25], acronimo per *Socially Pertinent Robots in Gerontological Healthcare*. Questo progetto ha come obbiettivo la creazione di robot in grado di interagire in maniera naturale con le persone, non solo da un punto di vista relativo al movimento ed alla funzionalità, ma anche dal lato emotivo ed espressivo. I risultati ottenuti verranno impiegati nel prossimo futuro principalmente in ambienti sanitari e residenze per anziani, ma le applicazioni per questa tecnologia sono molto vaste ed è molto probabile che queste tecniche possano essere adottate non solo nella robotica più in generale, ma anche in maniera importante nel campo della interazione uomo-macchina, per esempio nella creazione di sistemi informatici in grado di adattarsi allo stato d'animo dell'interlocutore, arrivando un giorno nella vita di tutti.

Per prima cosa vengono trattati i tipi di **feature extraction** maggiorente utilizzati, ovvero quei processi che vengono applicati ai dati per ottenere le *features* da utilizzare come input per i modelli. Questi processi si possono suddividere in vari modi, vale la pena menzionare la distinzione tra estrazione di features locali e globali, ovvero tra metodi che estraggono varie caratteristiche da una registrazione e metodi che invece generano caratteristiche singole spesso tramite medie o altre operazioni statistiche. Entrambi questi metodi hanno vantaggi e svantaggi, ma ultimamente vengono usate principalmente features locali basate sull'estrazione tramite spettrogrammi, mettendo in secondo piano le cosiddette features continue. Queste ultime vanno ad analizzare varie qualità del segnale, come l'intonazione, l'energia e le frequenze fondamentali e formanti e sono utilizzate principalmente con i modelli di machine learning classici. Per quanto riguarda l'estrazione basata su spettrogrammi viene usata una trasformazione di Fourier sui segmenti temporali che compongono la registrazione per costruire una rappresentazione dei dati basata sulle dimensioni del tempo e delle frequenze. I principali algoritmi utilizzati in questo campo sono quelli dello spettrogramma Mel e degli MFCC, che vengono solitamente abbinati ai modelli di Deep Learning, in quanto questi sono in grado di gestire con facilità i casi in cui si hanno molti dati in input.

In aggiunta alla feature extraction vi sono altre trasformazioni che possono essere effettuate sulla sorgente audio che in alcuni paper consentono di raggiungere risultati molto migliori, tra queste quelle più utilizzate sono la rimozione delle parti in silenzio delle registrazioni e una operazione di "pulitura" degli audio che consiste nel provare a rimuovere il rumore di fondo irregolare. Una ulteriore modifica che si è rivelata molto positiva nei risultati è l'aggiunta di un leggero e costante rumore di fondo, atto a mascherare quello già presente e a diminuire le differenze tra gli attori.

Per quanto riguarda i **modelli** impiegati viene fatta una distinzione tra modelli che classificano una registrazione per intero e modelli che invece attuano la classificazione a livello di segmenti e poi tramite questa valutano la registrazione, questa distinzione vale sia per modelli classici che di Deep Learning. Tra i modelli classici quelli più utilizzati in passato sono gli *Hidden Markov Models* e le *Support Vector Machines*, con a volte l'uso di *K-Nearest Neighbours* per la sua semplicità. Questi

riescono a raggiungere risultati abbastanza soddisfacenti, ma comunque decisamente più bassi rispetto a modelli quali le *Convolutional Neural Networks*, *Recurrent Neural Networks* e basati su attenzione come i *Transformers*.

Vengono successivamente esposti più nel dettaglio due paper che sono, per quanto concerne la nostra ricerca, l'attuale **stato dell'arte** per i dataset RAVDESS [26] e IEMOCAP [4]. Il primo [20] utilizza una CNN a due dimensioni per analizzare gli spettrogrammi generati dall'audio e mostra che le operazioni di preprocessing sono molto importanti per il risultato finale, con un aumento della precisione di più di 10 punti percentuali, arrivando ad una accuracy attorno al 80%. Nel secondo [5] vengono usate una combinazione di CNN e attenzione, con l'uso di alcuni Transformer che compongono una classificazione a partire da dati audio, testuali e visivi, arrivando ad una precisione del 84.5%. É da tenere comunque in forte considerazione il fatto che solitamente non è possibile comparare direttamente due dataset, in quanto essi sono strutturati in maniera diversa, spesso comprendono emozioni differenti e metodi di registrazione ed etichettamento delle emozioni discordi.

Le **strutture** che sono state sperimentate sono una *Temporal Convolutional Network* (TCN) ed una Conv-TasNet, esse sono esposte nel dettaglio nel capitolo 4. La scelta della TCN risiede nel fatto che questo modello è stato proposto per trovare una valida alternativa alla combinazione di CNN e RNN per l'estrazione di features temporali, caratteristica intrinseca delle tracce audio.

Per quanto riguarda i **risultati** ottenuti per prima cosa è stata fatta una comparazione tra i due modelli, osservando che la TCN sembra essere più adatta a questo compito, quindi si è deciso di proseguire solo con questa struttura. La maggior parte delle esecuzioni è stata eseguita con il dataset RAVDESS, per prima cosa si è notato che tra i metodi di feature extraction sia gli MFCC che lo spettrogramma Mel con scala in dB ottengono buoni risultati ed è stato deciso di utilizzare quest'ultimo. Successivamente si nota come le dimensioni della rete non sembrano avere un impatto diretto sulle prestazioni, limitandosi a delle leggere variazioni. Un problema della TCN è l'overfitting: la struttura tende a concentrarsi molto sulle caratteristiche particolari delle registrazioni presenti nel training set, non generalizzando bene le informazioni presenti. Per questo motivo sono stati aggiunti dei layer di Dropout che permettono di ridurre questo problema, migliorando leggermente le prestazioni. Tra le modifiche apportate all'audio l'aggiunta di un leggero rumore di fondo è decisamente posivita per la classificazione, essa permette di migliorare di molto le prestazioni, con un aumento attorno al 8% nel caso speaker independent; altre modifiche che sono state sperimentate, invece, portano a dei peggioramenti.

Il miglior risultato raggiunto arriva ad una accuracy del 81,2% stabilendo il nuovo stato dell'arte per il dataset RAVDESS. Per quanto riguarda i dataset IEMOCAP e CREMA-D, dove comunque sono state eseguite esecuzioni limitate, si raggiungono prestazioni inferiori rispettivamente attorno al 70 e al 65%, che sono piuttosto lontane dallo stato dell'arte del 85 e 81.5%.

Per testare la robustezza della struttura proposta, sono state eseguite delle prove mediante l'utilizzo di dataset diversi in fase di training e testing. In questo contesto si raggiunge l'accuracy migliore tramite la combinazione di IEMOCAP per train e CREMA-D per test con un risultato del 56.4%.

Ricordando le molteplici applicazioni di questa tecnologia, è possibile affermare che la TCN sia un modello molto valido e robusto per questo compito, raggiungendo risultati in generale molto positivi sui vari dataset ed essendo in grado di riconoscere le emozioni in configurazione cross-dataset. É stato verificata inoltre l'importanza delle operazioni di feature extraction e preprocessing, che permettono di migliorare in maniera notevole i risultati ottenuti.

I possibili **sviluppi futuri** dal lato tecnico sono molteplici, partendo dall'integrazione in un sistema multimodale con video e dati testuali, alla generazione di altri speaker, da usare sia come data

augmentation che per ricondurre sistematicamente il caso speaker independent al dependent. Saranno inoltre fondamentali altri test su dataset differenti non composti da attori, con lo scopo di migliorare l'affidabilità dei modelli in situazioni reali.

1 Introduzione

Uno dei canali con cui le persone veicolano molto significato è attraverso l'uso della voce. Questa può essere uno degli strumenti più semplici e immediati per scambiare informazioni tra gli individui, non solamente mediante le parole pronunciate, ma anche dal contesto, dalla intonazione della stessa e dalle emozioni che vengono manifestate. A questo riguardo il riconoscimento dell'emozione umana da sorgente audio è un argomento di ricerca da molti anni ed è stato approcciato in vari modi. Per una persona è un compito del tutto naturale, ma è ancora piuttosto difficile avere delle classificazioni accurate da una macchina. Ultimamente, grazie agli sviluppi nell'ambito del Deep Learning, è possibile costruire dei modelli che approssimano piuttosto bene questo compito, soprattutto quando si usano dei dataset creati in laboratorio, con attori che spesso esagerano i tratti caratteristici di ogni emozione. Il riconoscimento automatico è invece ancora piuttosto difficoltoso quando vengono usati dataset cosiddetti "in the wild", ovvero quelli che ritraggono persone nella vita di tutti i giorni e che quindi esprimono le emozioni in maniera normale, spesso esprimendone più di una nello stesso momento oppure avvalendosi di sarcasmo o di altre manifestazioni che possono dipendere fortemente dalla cultura e dalla lingua parlata.

In questo documento verranno analizzati nel capitolo 2 i metodi più comuni che sono stati adottati in questi anni per risolvere questo problema, sia in termini di modelli utilizzati che di tecniche di preprocessing sperimentate, con una parte dedicata allo stato dell'arte attuale. Successivamente verranno esposti nei capitoli 3 e 4 i dataset utilizzati e i modelli sperimentati. I risultati ottenuti saranno raccolti ed analizzati nel capitolo 5, includendo una comparazione con lo stato dell'arte, mentre le conclusioni alle quali è stato possibile arrivare e gli sviluppi futuri saranno esposti nel capitolo 6.

2 Metodi

In questo capitolo verranno esposti i principali metodi che vengono adottati in bibliografia per trattare il problema del riconoscimento dell'emozione da sorgente audio o *Speech Emotion Recognition*.

Per prima cosa è importante parlare dei vari metodi di *feature extraction*, ovvero quei metodi che permettono di estrarre delle caratteristiche, dette *features*, da successivamente usare per allenare i modelli che verranno usati. Questa parte sarà piuttosto dettagliata in quanto è, come si può vedere in [20], molto importante per il raggiungimento di buone prestazioni e può avere un impatto paragonabile a quello del modello utilizzato, tanto da migliorare le prestazioni dei modelli (nel paper sopra citato) di oltre il 10%.

Vedremo che vi è una certa differenza tra i metodi di feature extraction adottati per modelli di machine learning classico, quali *K-NN* e *Support Vector Machines*, e modelli di Deep Learning, quali *CNN* e *RNN*.

Successivamente verranno esposti alcuni dei modelli più comuni che sono stati adottati, con particolare riguardo agli ultimi sviluppi, che verranno descritti nella sezione 2.3.

2.1 Tipi di feature extraction

Una problema molto importante in tutti gli scenari di machine learning è quello della feature extraction: l'estrazione delle informazioni corrette può fare la differenza tra un buon modello e un risultato piuttosto inutile. Per quanto riguarda questo tema vi sono stati molti tentativi nel corso degli anni per capire quali potessero essere delle caratteristiche adatte ad una classificazione efficace delle emozioni dall'audio. Un sondaggio del 2011 riassume bene molte dele tecniche "classiche" utilizzate in questo campo [7]. Confrontando questo paper con le tecniche adottate più di recente, come anche in uno studio del 2019 [1], vediamo che esso è ancora attuale e che i problemi generali e le considerazioni che li vengono esposte riguardo l'estrazione delle features sono le stesse che troviamo in molti lavori degli ultimi due anni, sebbene la scena si sia spostata verso i modelli di Deep Learning.

2.1.1 Features locali o globali

Nel contesto della estrazione delle features è importante distinguere tra due tipi di estrazione:

- A caratteristiche *locali*: ovvero quei metodi che estraggono caratteristiche multiple per ogni registrazione dividendola in piccoli segmenti temporali chiamati *frames*
- A caratteristiche *globali*: ovvero quei metodi che generano features singole per ogni registrazione, spesso ottenute combinando le features locali.

A questo riguardo i ricercatori non sono tutt'oggi d'accordo su quale sia l'approccio migliore in linea teorica, ma gli ultimi lavori prediligono quasi esclusivamente l'estrazione di features locali, basate spesso su spettrogrammi.

I principali vantaggi dell'uso di caratteristiche locali sono una più completa caratterizzazione della sorgente audio, infatti consentono di avere molti più dati per ogni registrazione. Questo è un punto molto importante quando si adottano modelli di Deep Learning, infatti questi lavorano molto bene

con una grande quantità di dati, a differenza dei modelli tradizionali. Un' altro vantaggio importante è anche la permanenza delle informazioni temporali presenti nei dati, in quanto queste vengono perse quando si ricorre a caratteristiche globali che spesso ricorrono ad una qualche forma di media sulla dimensione temporale.

I principali vantaggi dell'uso caratteristiche globali invece sono le prestazioni migliori ottenute tramite algoritmi semplici quali K-NN, il minor numero di dati generati permette di avere tempi di training molto ridotti, permettendo di eseguire più operazioni di cross-validation e più analisi in ricerca dei parametri migliori.

2.1.2 Features continue

Alcune delle caratteristiche che sono state usate molto nell'ambito dell'analisi di sorgenti audio sono le cosiddette features continue, queste si suddividono principalmente in quattro categorie:

- Analisi della *intonazione* (*Pitch*): questa è una qualità del suono che determina l'altezza della nota percepita, è formata dalla frequenza fondamentale e dalle sue armoniche.

- Analisi dell' *energia*: misura dell'energia compresa in un certo segmento temporale del segnale, quindi associata alla potenza di questo in un determinato istante.

- Analisi della *frequenza fondamentale (F0)*: questa è la frequenza più bassa di un segnale periodico. Insieme alle sue armoniche forma l'intonazione generale di un suono.

- analisi delle *frequenze formanti*: sono le frequenze di un suono (spesso coincidenti con delle armoniche) che presentano un picco in ampiezza del segnale, quindi in potenza dello stesso.

Vi sono vari modi per analizzare queste caratteristiche, quelli più usati come già illustrato si basano su caratteristiche locali o globali: solitamente viene estratto il valore di una certa metrica per piccoli intervalli di tempo (solitamente tra i 10 e gli 80ms) e successivamente si possono rispettivamente utilizzare tutte le misure per allenare i modelli oppure, cosa piuttosto comune fino a qualche anno fa, applicare delle funzioni alle serie di dati, estraendo quindi media, mediana, massimo, minimo, deviazione standard o altro.

2.1.3 Estrazione basata su spettrogrammi

In questa parte si espone una tipologia di estrazione delle features basata su *spettrogrammi*. Questo è l'approccio preferito negli ultimi anni, superando l'uso di caratteristiche continue anche grazie alla capacità dei modelli basati sul Deep Learning di gestire un numero di features molto elevato.

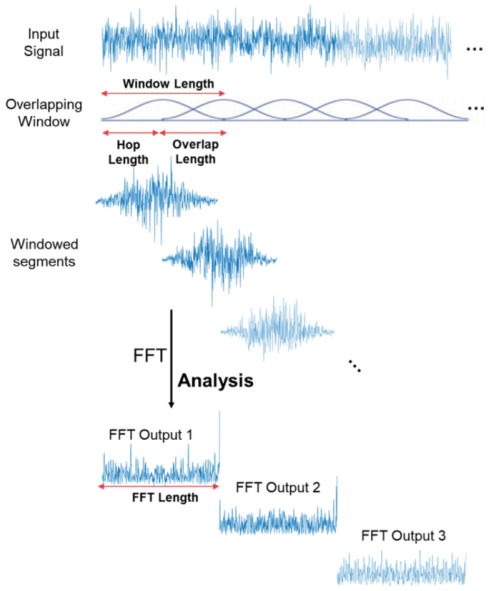
Queste caratteristiche vengono estratte tramite una specifica procedura che adesso andremo a delineare:

1. Il segnale audio in forma d'onda viene diviso anche qui in segmenti (lunghi solitamente tra i 20 e i 30ms) con una tecnica chiamata *windowing*: questa consiste nell'applicare al segnale una sinusoide (in valore assoluto), creando delle *window*, solitamente queste sono sovrapposte in modo da non perdere segnale nei punti in cui la sinusoide assume valori vicini allo zero.

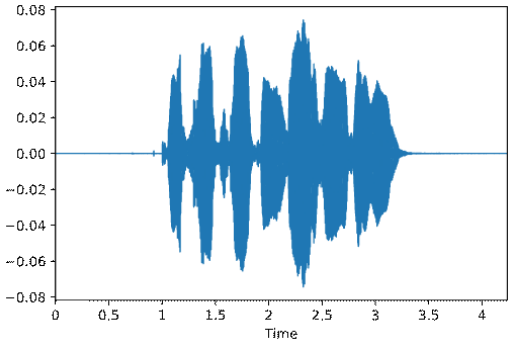
2. Il segnale di ogni window viene trasformato tramite una trasformazione di Fourier, più precisamente la *short time Fourier transform*: questa permette di convertire i dati dal dominio temporale a quello delle frequenze, consentendo una analisi più efficace.

3. Le sequenze così ottenute vengono impilate in modo da creare una rappresentazione del segnale audio in due dimensioni, dove la prima dimensione rappresenta i segmenti temporali in cui è stato diviso il segnale e la seconda rappresenta le frequenze. Questa rappresentazione è chiamata spettrogramma.

4. Successivamente possono venire applicate delle altre trasformazioni che dipendono dallo specifico algoritmo che si sta utilizzando, questi sono esposti in seguito.

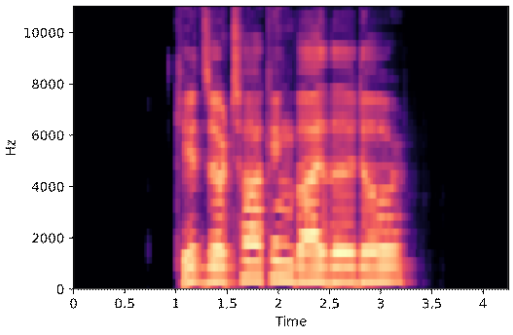


(a) Estrazione dello spettrogramma (da mathworks)

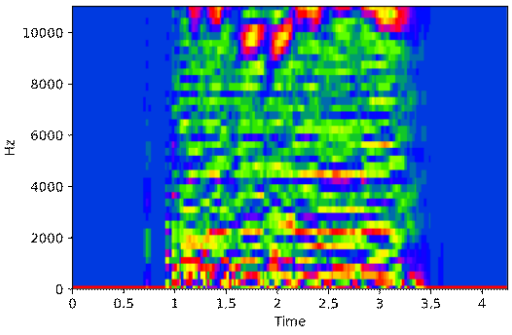


(b) Forma d'onda

Figura 2.1: Estrazione spettrogrammi e forma d'onda iniziale



(a) Spettrogramma Mel



(b) MFCC

Figura 2.2: Risultati del preprocessing

Spettrogramma Mel

Il nome dello *spettrogramma Mel* deriva dal mel, una unità di misura basata sulla percezione dell'orecchio umano alla frequenza fisica del suono, infatti sembra che questa non venga percepita in maniera lineare, ma secondo una scala logaritmica. Una approssimazione per le frequenze superiori a 1KHz può essere espressa come segue:

$$f_{mel} = 2595 \log\left(1 + \frac{f}{700}\right)$$

Dove f esprime la frequenza fisica (in Hz), mentre f_{mel} denota la frequenza percepita.[23]

Per ottenere lo spettrogramma Mel è sufficiente applicare questa trasformazione allo spettrogramma ottenuto precedentemente. Una variante comune che solitamente ottiene risultati migliori è la conversione dei valori da potenza a decibel: questa viene effettuata tramite la formula seguente:

$$I_{dB} = 10 \log_{10}(I_{pow})$$

dove I_{dB} rappresenta l'intensità in decibel, I_{pow} in potenza.

Alcuni lavori che usano questo algoritmo sono [20, 31, 6, 14, 32, 17]

MFCC

I *Mel Frequency Cepstral Coefficients* sono dei coefficienti calcolati dallo spettrogramma Mel applicando una *trasformata discreta del coseno*: questa è una trasformazione simile alla trasformazione di Fourier. Sono molto usati nell'ambito del riconoscimento delle emozioni, infatti molti dei paper analizzati li utilizzano. [18, 19, 13, 14, 2]

2.1.4 Altri tipi di preprocessing

In alcuni lavori [20] prima di calcolare le features da dare come input ai classificatori vengono applicati altri tipi di modificazioni alla sorgente audio. Queste solitamente hanno lo scopo di migliorare la qualità delle registrazioni per raggiungere accuracy migliori.

Le principali variazioni che vengono applicate sono:

- rimozione del rumore di fondo: può essere una operazione molto importante per evitare che il classificatore si concentri su di esso, può essere molto utile soprattutto quando le registrazioni sono state condotte in un ambiente all'aperto o non sotto controllo.

- rimozione delle parti in silenzio: questa operazione permette a volte di evitare di avere sequenze di cui gran parte della registrazione è formata da silenzio. Un problema di applicare questa variazione è quello che a volte le pause trasmettono delle emozioni, quindi rimuovendole si altererà negativamente la registrazione.

- inserimento di rumore casuale: questa tecnica può sembrare in contraddizione rispetto alla rimozione del rumore di fondo, ma l'aggiunta di rumore (con una intensità costante nel tempo) può aiutare nella classificazione riducendo l'overfitting e aiutanto il modello a generalizzare.

- altri tipi di *data augmentation*: questi sono più rari, in quanto i metodi più semplici (come modifica della tonalità, della velocità e altre) tendono a peggiorare le prestazioni in quanto modificano il contenuto emotivo della registrazione, come vedremo anche nel capitolo 5. Alcune tecniche sembrano comunque essere efficaci, come quella adottata da [9] che combina una operazione di oversampling e perturbazione del segnale, oppure quella di [33] che utilizzano delle *Generative Adversarial Networks* (GAN)

2.2 Modelli utilizzati

Una distinzione da fare nei modelli utilizzati è sicuramente tra i modelli cosiddetti classici (K-NN, SVM) e quelli legati al Deep Learning, quindi che utilizzano reti neurali. Negli ultimi 10 anni c'è stata una progressiva ma decisa tendenza a preferire questi ultimi, non solo per il fatto permettono di raggiungere prestazioni migliori, ma anche perché semplificano l'operazione di feature extraction tramite l'uso di spettrogrammi, non rendendo necessaria una estrazione manuale.

Una differenza da tenere in considerazione, sia per quanto riguarda i modelli classici che quelli di Deep Learning, è il fatto che alcuni approcci tendono a classificare le registrazioni per intero, mentre altri classificano ogni segmento in base alle caratteristiche locali che sono state estratte, poi solo in un secondo momento calcolano una classificazione complessiva della traccia audio. A questo riguardo la maggior parte dei lavori analizzati utilizza il primo approccio, ma non mancano quelli che prediligono il secondo, come [12, 10, 17].

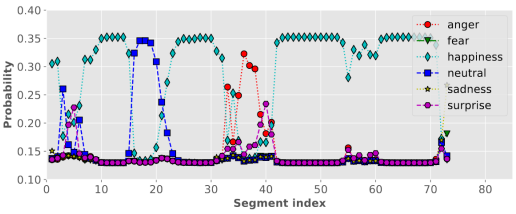


Figura 2.3: Classificazione sui segmenti in [17]

2.2.1 modelli classici

Come abbiamo visto i modelli classici sono ormai poco utilizzati, uno degli ambiti in cui vengono ancora adottati è quello della comparazione tra i metodi di feature extraction, come in [2], dove vengono utilizzate delle SVM.

K-NN

Uno dei modelli più semplici in assoluto e il K-Nearest Neigbours: questo si basa sull'idea di classificare un oggetto ignoto in base alla classe di appartenenza dei k oggetti più vicini, adottando generalmente come misura di distanza la distanza euclidea. Questi modelli hanno il vantaggio di non necessitare propriamente di un allenamento, questo però al costo di avere dei tempi di predizione piuttosto lunghi.

SVM

Le Support Vector Machines sono un tipo di modello molto diffuso, si basano sull'idea di trovare un piano (solitamente un iperpiano) che permetta di ottenere il massimo margine tra le classi, ovvero di massimizzare la distanza tra i punti appartenenti a classi diverse che si trovano più vicini al piano. Questo implica che il modello formato sia lineare: infatti riesce ad adattarsi senza errori solo a dati linearmente separabili. Per risolvere questo problema si utilizzano delle funzioni cosiddette di kernel: queste servono a mappare i dati in uno spazio con più dimensioni rispetto a quello di partenza, permettendo di dividere in maniera lineare dei dati che precedentemente non erano divisibili con questo metodo.

HMM

Gli Hidden Markov Models sono una classe di modelli utilizzata principalmente nel riconoscimento delle parole, ma sono stati applicati con successo anche al riconoscimento delle emozioni, infatti tra i modelli classici sono quelli più utilizzati. Questi modelli si basano su di una catena di stati nascosti

non osservabili, dove ogni stato dipende esclusivamente dallo stato precedente. Ogni stato nascosto può generare un evento osservabile che poi costituisce l'output del modello.

2.2.2 Deep Learning

In questa parte verranno esposti i metodi usati più di recente, ovvero quelli che riguardano il Deep Learning: questi sono spesso, ma non sempre, migliori (a livello di performance) dei metodi tradizionali, ma spesso le metodologie di confronto e i dataset impiegati sono diversi e quindi una comparazione diretta risulta piuttosto difficile [28].

Le Reti Neurali Artificiali sono fondamentalmente dei modelli costituiti da un livello di input, uno o più livelli nascosti e uno di output. I livelli sono composti da nodi: il numero di nodi nello strato di input e di output dipende rispettivamente dalla rappresentazione dei dati che viene usata e dal numero di classi, il numero dei nodi nei livelli nascosti, come il numero stesso di questi strati, è variabile. Ogni livello è connesso al successivo tramite dei pesi inizializzati in maniera casuale. Quando un campione viene estratto dal training set, i suoi valori sono caricati nel livello di input e poi propagati fino al livello di output. A questo punto tramite l'algoritmo di *backpropagation* i pesi vengono aggiornati. Alla fine dell'allenamento ci si aspetta che la rete sia in grado di classificare nuovi dati. Uno schema di una semplice rete neurale è presente nella figura 2.4a.

CNN

Le *Convolutional Neural Networks* sono un tipo particolare di rete neurale che è molto usata nella classificazione e nel riconoscimento di oggetti nelle immagini, sono dei modelli molto buoni nel catturare delle caratteristiche locali dei dati, rendendo meno importante la posizione di queste caratteristiche nel quadro generale . Sono state applicate con successo al riconoscimento delle emozioni dall'audio, ma spesso mancano della capacità di catturare le caratteristiche temporali intrinseche in questo problema.

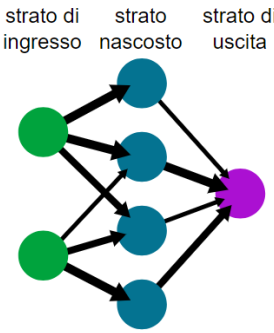
Le CNN si basano su di una operazione detta convoluzione: questa si basa sull'applicare una matrice detta kernel ai dati in input. Ogni valore della matrice in input viene moltiplicato per il corrispondente valore della matrice kernel, poi i valori generati vengono sommati per andare a formare uno dei valori della matrice contenente i risultati. Questo procedimento è schematizzato nella figura 2.4b.

RNN

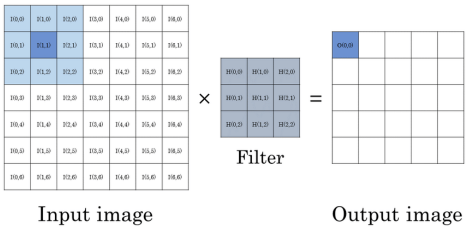
Le *Recurrent Neural Networks* sono un tipo di rete che, a differenza delle cosiddette *Feed Forward Neural Networks*, includono nella loro struttura dei cicli. In particolare, i dati in input vengono passati secondo una sequenza temporale, quindi l'output di un nodo ad un tempo t può diventare uno degli input dello stesso nodo nel tempo $t + 1$. Questi tipi di reti sono molto usate quando non si conosce a priori la dimensione dei dati in ingresso oppure in uscita, come per esempio nella traduzione del linguaggio naturale.

Un vantaggio rispetto alle CNN per il problema che affrontiamo è quello che le RNN sono in grado di modellare in maniera naturale le caratteristiche temporali delle registrazioni, questo però al prezzo di una non così buona caratterizzazione di features locali nei dati e di un tempo maggiore necessario per l'allenamento in quanto permettono una minore parallelizzazione.

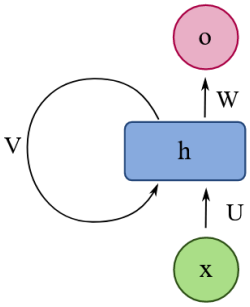
Esistono vari tipi di celle riguardanti le RNN, questi dipendono dalla struttura della cella stessa. I più comuni sono le *LSTM* (Long-Short Term Memory networks) e i *GRU* (Gated Recurrent Unit).



(a) Rete neurale (da wikipedia)



(b) Operazione di convoluzione (da [3])



(c) Struttura di una RNN (da wikipedia)

Figura 2.4: Vari tipi di rete neurale

2.2.3 modelli basati su attenzione

Negli ultimi anni stanno prendendo piede alcuni modelli basati su un meccanismo di attenzione, in particolare sui *Transformer*, una struttura introdotta nel 2017 [27] per il problema della traduzione di sequenze. Sebbene alcuni paper come [6] usino direttamente questa struttura per la classificazione delle emozioni, l'approccio più comune risulta essere quello di usare questo tipo di architettura nella fusione delle modalità: quindi nel unire i dati provenienti dall'audio, video e talvolta dalle trascrizioni testuali.

2.3 Stato dell'arte

In questa sezione tratteremo più in dettaglio qualche pubblicazione che sembra avere i risultati migliori, è tuttavia da tenere in forte considerazione il fatto che una comparazione diretta tra i vari lavori è spesso molto complicata, se non impossibile. Questo è dovuto al fatto che i vari modelli vengono testati su dataset che, anche se possono essere gli stessi, spesso utilizzano metodi di suddivisione o addirittura di selezione delle registrazioni che possono essere molto diversi, per non parlare dei tipi di preprocessing utilizzati.

Le strutture che sembrano ottenere i risultati migliori al giorno d'oggi sembrano essere quelle che utilizzano più modalità per la classificazione mediante delle CNN, a volte combinate con delle RNN o con dei meccanismi di attenzione, quest'ultimi a volte sfruttati come già detto per l'unione delle modalità.

In *A CNN-Assisted Enhanced Audio Signal Processing for Speech Emotion Recognition* [20] vengono usati i dataset RAVDESS [26] e IEMOCAP [4] con solamente i dati relativi all'audio. Questi vengono applicati ad una CNN composta da sette livelli convoluzionali e due livelli lineari, schematizzata nella figura 2.5. Una caratteristica peculiare di questo paper è l'utilizzo di convoluzioni in due dimensioni quando la maggior parte dei lavori le applica in una dimensione, quindi i dati generati dallo spettrogramma vengono trattati a tutti gli effetti come delle immagini, non come dati unidimensionali su più canali. I dataset vengono divisi con l'80% delle registrazioni affidata al training e il 20% al testing in maniera casuale, quindi si tratta di una configurazione *speaker dependent* (concetto esposto meglio nel capitolo 5). Il tipo di estrazione features che è stato applicato prevede l'utilizzo dello spettrogramma Mel, con in particolare una operazione di rimozione delle parti rumorose e silenziose dagli audio che viene dimostrata avere una importanza fondamentale nel miglioramento delle

prestazioni, con un incremento della precisione di più di 10 punti percentuali. Questo paper racchiude il risultato migliore ottenuto per il dataset RAVDESS, con una accuracy del 80%. É da notare però che questo dataset non è diffuso quanto IEMOCAP, quindi sono stati condotti meno studi su di esso.

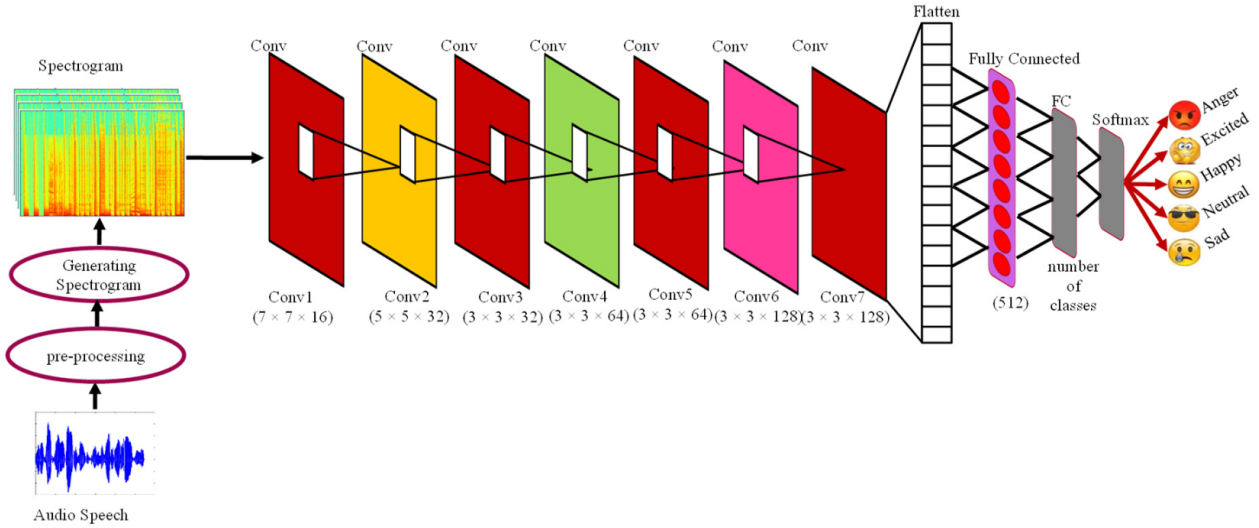


Figura 2.5: Struttura della rete usata in [20]

In *Multimodal End-to-End Sparse Model for Emotion Recognition* [5] vengono composte delle caratteristiche multimodali a partire da dati audio, visivi e testuali. I dataset presi in considerazione sono IEMOCAP e CMU-MOSEI [30], anche qui la divisione delle registrazioni è casuale, con il 70% in training, 10% in validation e 20% in testing. Il modello adottato consiste di un insieme di CNN e meccanismi di attenzione, con l'uso dei Transformer, mostrato in figura 2.6. Qui si impiegano delle cosiddette *sparse CNN*, dove solo alcuni punti della convoluzione sono attivi. Il tipo di feature extraction adottato comprende vari elementi, tra questi vengono utilizzati degli MFCC a 22 canali e 108 features statistiche non meglio identificate. Questo modello stabilisce, per quanto concerne la mia ricerca, lo stato dell'arte per il dataset IEMOCAP, con una accuracy del 84.5% (con medoto multimodale).

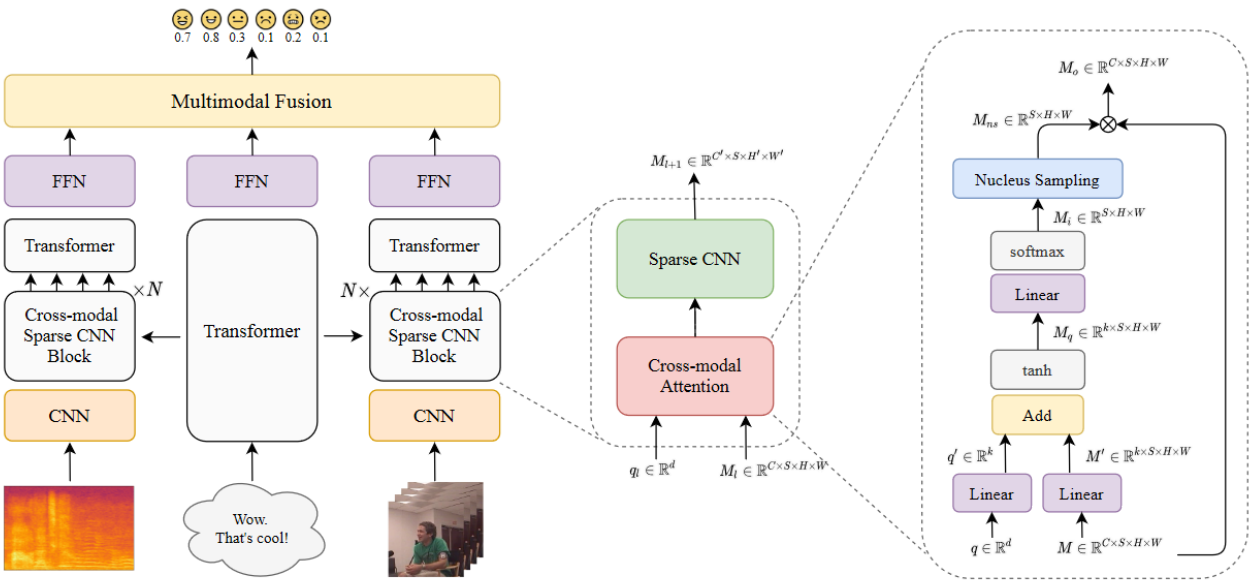


Figura 2.6: Struttura della rete usata in [5]

3 Dataset utilizzati

In questo capitolo verranno esposti i dataset considerati in questo documento, alcuni di questi sono composti da dati multimodali quali audio, video (a volte inclusivi di landmark facciali precalcolati) e trascrizione testuale delle parole pronunciate. Molti studi pubblicano risultati che utilizzano più di una modalità, noi ovviamente ci concentreremo principalmente su quelli riguardanti l'audio.

3.1 RAVDESS

Il *Ryerson Audio-Visual Database of Emotional Speech and Song*[26] è un dataset piuttosto recente pubblicato nel 2018. Esso contiene registrazioni video e audio da 24 attori professionisti, 12 uomini e 12 donne, che pronunciano due frasi in inglese (americano) con un accento neutro.

Le frasi sono ripetute sia in parlato che in cantato. Le emozioni espresse nelle registrazioni del parlato sono catalogate come **neutral, calm, happy, sad, angry, fearful, disgust e surprise**, mentre quelle del cantato sono **neutral, calm, happy, sad, angry, e fearful**. Ogni emozione è registrata a due livelli di intensità (normale, forte), con l'eccezione di **neutral**, che dispone solo dell'intensità normale.

Tutte le registrazioni sono disponibili in audio (16bit, 48kHz .wav), audio-video (720p H.264, AAC 48kHz, .mp4), o solo video. Non sono presenti i file cantati per l'attore 18.
In totale il dataset si compone di 2452 file audio, di cui 1440 relativi al parlato e 1012 relativi al cantato. In questo lavoro verranno considerati entrambi, in quanto il numero di registrazioni è piuttosto limitato e questo ci permette di allenare modelli più solidi.

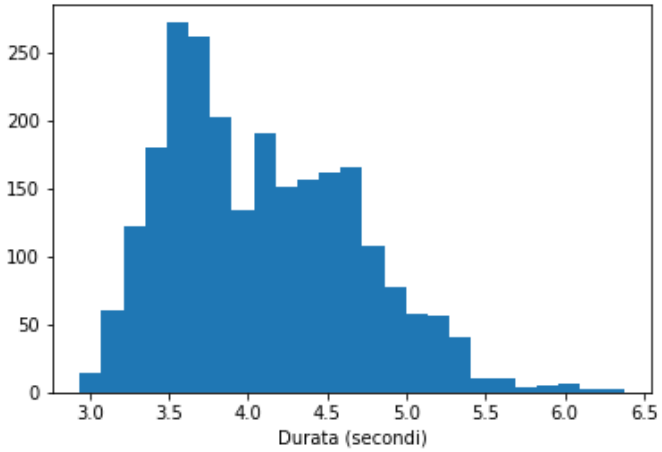


Figura 3.1: Durata delle registrazioni in RAVDESS

La suddivisione dei dati per classe in totale è quindi di 188 registrazioni per la classe *neutral*, 192 per le classi *disgust* e *surprised* e di 376 per le altre.

Le due frasi pronunciate sono:

1. *Kids are talking by the door*
2. *Dogs are sitting by the door*

Il dataset RAVDESS è di gran lunga quello analizzato maggiormente in questo documento.

3.2 IEMOCAP

L' *Interactive Emotional Dyadic Motion Capture*[4] è un dataset sceneggiato multimodale comprensivo di 10 attori, 5 uomini e 5 donne. Esso contiene circa 12 ore di dati audiovisivi comprendenti video,

audio, cattura del movimento del viso e trascrizioni testuali. Consiste in 5 sessioni dove compaiono due attori che si cimentano in scenari improvvisati o sceneggiati precedentemente, specificatamente scelti per suscitare espressioni relative a delle emozioni.

Il database è annotato da più persone e le etichette si dividono in categoriche (**angry, happy, sad, neutral, frustrated, excited, fearful, surprised, disgusted, other**) e di dimensione quali valence, activation e dominance, queste ultime verranno ignorate per questo lavoro.

La distribuzione delle registrazioni per emozione è la seguente:

etichetta		angry	excited	frustrated	happy	sad	neutral	surprise
improvvisati	# segmenti	289	663	971	284	608	1099	60
	durata (minuti)	22.15	42.14	79.94	19.62	50.23	74.54	3.37
totali	# segmenti	1103	1041	1849	595	1084	1708	107
	durata (minuti)	82.96	92.95	145.16	43.05	99.30	111.08	5.54

Tabella 3.1: Distribuzione delle registrazioni per classe in IEMOCAP (da [13])

Visto che il numero di registrazioni è molto basso per le etichette *fear*, *disgust* e *other* (45 in tutto), queste vengono eliminate in quasi tutti i lavori che abbiamo incontrato, quindi saranno ignorate anche qui.

Per quanto riguarda le altre emozioni, la maggior parte dei lavori in bibliografia usa solamente **4 emozioni**: *happy, sad, neutral, angry*, con a volte l'emozione *excited* inclusa in *happy*, quest'ultimo è l'approccio che verrà usato in questo lavoro. Altri paper compiono scelte diverse e meno comuni, come quelle di usare 5,6 o 8 emozioni.

Il dataset IEMOCAP è piuttosto importante in bibliografia, infatti una gran parte dei paper sull'argomento utilizza questo comprensorio.

3.3 CREMA-D

Il *Crowd-sourced Emotional Multimodal Actors Dataset*[11] è un dataset composto da 7442 registrazioni originali da 91 attori. Queste registrazioni sono state create da 48 uomini e 43 donne tra i 20 e i 74 anni, appartenenti a diverse etnie: Afroamericana, Asiatica, Caucasica, Ispanica e non specificata.

Gli attori hanno interpretato 12 frasi presentate con una tra sei emozioni diverse: **Anger, Disgust, Fear, Happy, Neutral e Sad** e quattro diversi livelli di intensità: (Low, Medium, High, Unspecified).

I partecipanti hanno dato una valutazione delle emozioni in base al video con audio, al video da solo e all' audio solamente. Dato il grande numero di valutazioni necessarie, questo sforzo è stato attuato con un meccanismo di crowd sourcing, dove un totale di 2443 partecipanti hanno valutato 90 clip uniche a testa, 30 audio, 30 video e 30 audio-video. Il 95% delle registrazioni ha più di 7 valutazioni. Il numero delle registrazioni è di 1087 per la classe neutral, 1271 per le altre classi.

Il dataset CREMA-D è piuttosto recente e poco diffuso, infatti non sono presenti molti paper che lo utilizzano. É stato deciso di utilizzarlo in quanto presenta delle caratteristiche simili a RAVDESS quali il numero limitato delle frasi e il fatto che queste vengono pronunciate da attori.

4 Architetture sperimentate

4.1 TCN

Le *Temporal Convolutional Networks* (**TCN**) sono un tipo di struttura introdotta nel 2016 [15] con lo scopo di trovare una valida alternativa alla combinazione di CNN e RNN per l'estrazione di feature temporali, questo sia in termini di performance a livello qualitativo del modello utilizzato, sia in termini di tempo di calcolo necessario per l'allenamento della rete.

Inizialmente questo modello è stato proposto per la classificazione di video e dati provenienti da sensori ed è stato mostrato essere in grado di modellare la struttura temporale dei dati molto bene, raggiungendo precisioni competitive se non migliori rispetto alla combinazione di CNN e RNN, ad una frazione del tempo di esecuzione. Questo lo rende un valido candidato per il nostro lavoro: la struttura temporale dell'audio è infatti risultata essere molto importante per il riconoscimento efficace delle emozioni.

La struttura che ho usato è leggermente diversa da quella esposta nell'articolo sopra citato: per prima cosa i dati in input vengono fatti passare per una cosiddetta *bottleneck*, ovvero uno strato di normalizzazione seguito da una convoluzione in 1 dimensione con dimensione del kernel uguale a 1 atta a modificare il numero di canali interni alla rete, successivamente si applica un dropout per ridurre l'overfitting. A questo punto si arriva alla struttura principale della rete: Questa è composta da vari blocchi ripetuti: Ognuno di questi blocchi di alto livello è formato da un insieme di blocchi convoluzionali simili, ma con dilatazione differente (potenze di 2): questo rende possibile la cattura di caratteristiche a vari livelli di ingrandimento. Questi blocchi sono ulteriormente suddivisi al loro interno come possiamo vedere nella figura 4.1b. L'output di ogni blocco viene successivamente sommato al vettore che era stato usato come input dello stesso e questo risultato viene usato come input per il blocco successivo. Una volta superati per tutti questi blocchi si applica un ulteriore dropout, seguito da una media sulla dimensione temporale che permette di avere i dati di dimensione prefissata, un ulteriore dropout + PReLU precedono l'ultimo strato lineare, che si occupa della classificazione finale.

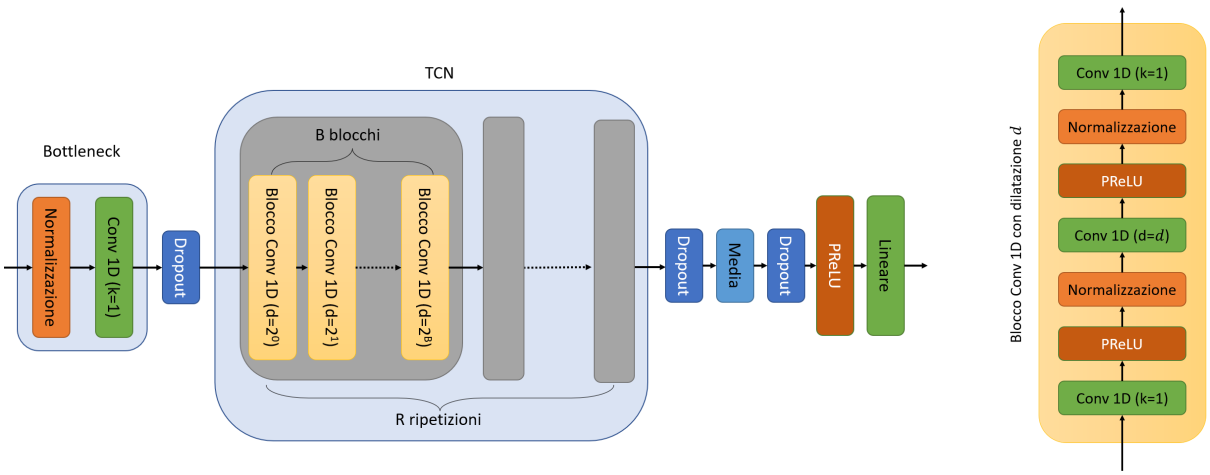
La struttura della rete è schematizzata nella figura 4.1

4.2 Conv-TasNet

La Conv-TasNet è una struttura simile alla TCN, infatti ne include una al suo interno. Questa architettura è stata proposta nel 2019 [16] per risolvere il problema noto come *speaker separation*, ovvero il dividere le tracce audio di due o più persone che parlano in contemporanea, ottenendo buoni risultati.

Ho provato ad applicare questa struttura al problema del riconoscimento delle emozioni, con qualche modifica: infatti il tipo di risultato che vogliamo ottenere è radicalmente diverso, prima era necessario generare delle tracce audio da una sorgente, ora viene richiesto di attuare una classificazione dei dati in input.

Il modello che ho utilizzato è diverso da quello nell'articolo per la mancanza di un decoder ed è così strutturato: per prima cosa si incontra un encoder formato da una convoluzione e una ReLU, successivamente il risultato viene dato come input ad una TCN, che a differenza della precedente ha lo



(a) Struttura generale

(b) Struttura di un blocco Conv1D

Figura 4.1: Struttura della TCN

scopo di generare una maschera. Questa maschera viene poi moltiplicata per il risultato dell'encoder, successivamente si passa per uno strato lineare e dropout, viene applicata una media nella dimensione temporale e un ultimo strato lineare si occupa della predizione. Uno schema della rete è presente nella figura 4.2

Una differenza piuttosto importante tra le due strutture è quella che la TCN utilizza come input dei dati preprocessati tramite spettrogramma Mel o MFCC, mentre la Conv-TasNet utilizza direttamente i dati in forma d'onda.

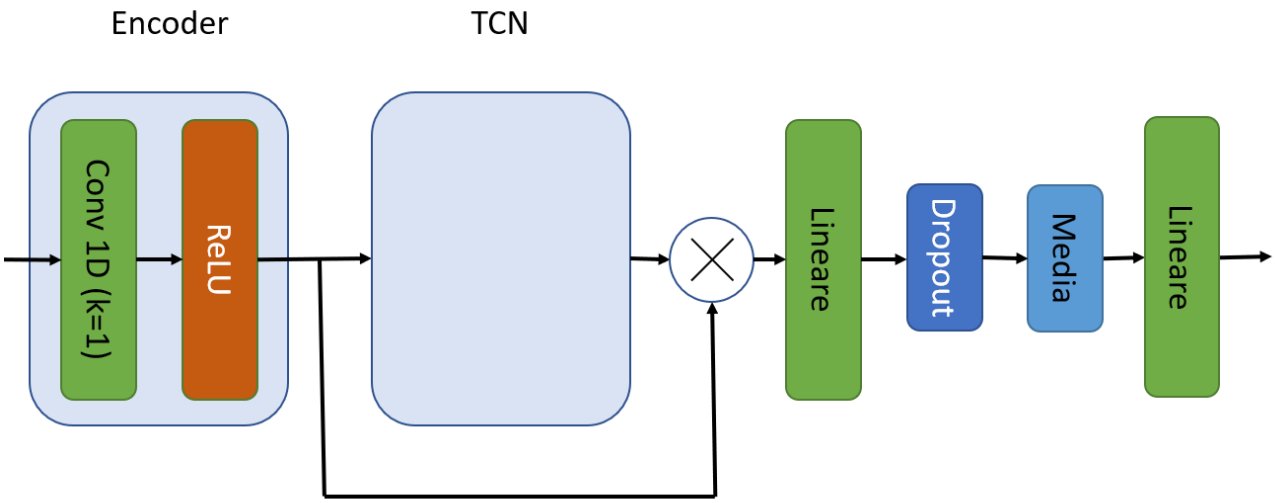


Figura 4.2: Struttura della Conv-TasNet

5 Risultati

In questa sezione verranno esposti i risultati ottenuti tramite le strutture e i dataset esposti precedentemente. La combinazione che è stata sperimentata maggiormente utilizza il modello TCN e il dataset RAVDESS, soprattutto per una questione di disponibilità dei dati e tempo a disposizione per il training.

Nella valutazione dei risultati che coinvolgono registrazioni di persone nella modalità audio vi sono due categorie distinte molto importanti, che nella classificazione delle emozioni riguardano la suddivisione dei dati nella fase di allenamento dei modelli:

- **Speaker Dependent** (dipendenti dall'oratore): sono tutte quelle analisi che in un modo o nell'altro dipendono dalla persona registrata, nel campo del riconoscimento delle emozioni in genere indica l'allenamento di un modello tramite registrazioni che, seppur diverse, appartengono allo stesso oratore/cantante che viene (spesso tra gli altri) utilizzato per la valutazione del modello.

- **Speaker Independent** (indipendenti dall'oratore): sono tutte quelle analisi che non dipendono dalla persona che appare nelle registrazioni, nel campo analizzato solitamente indica il fatto che i modelli vengono allenati tramite registrazioni appartenenti ad alcuni individui, mentre le fasi di validazione e test vengono effettuate su registrazioni di altri individui, a volte mescolati tra queste ultime.

In questo documento le metriche utilizzate per la misura dei risultati ottenuti sono conformi con molti dei paper presenti in bibliografia, ovvero tramite *Unweighted Accuracy* e *Weighted Accuracy*. Queste due misure indicano rispettivamente la percentuale di registrazioni catalogate in maniera esatta rispetto al totale e la media delle percentuali di classificazione corrette calcolate per classe. Le formule seguenti esprimono in maniera rigorosa questi concetti:

$$Acc_{unweighted} = \frac{1}{N} \sum_{n=0}^{N} 1(y_n = p_n)$$

dove N è il numero di elementi, 1() è una funzione indicatrice, y_n è l'etichetta corretta e p_n è la predizione.

$$Acc_{weighted} = \frac{1}{|C|} \sum_{c \in C} A_c$$

$$A_c = \frac{TP_c}{TP_c + FN_c}$$

dove C è l'insieme delle classi, $|C|$ è la sua cardinalità, TP_c indica i veri positivi per una classe c e FN_c indica i falsi negativi per la classe.

I risultati che verranno esposti sono stati ottenuti mediante una operazione di preprocessing sui dati: in particolare questi sono stati trattati con una operazione di *trim* e *padding*, che consistono nel rispettivamente togliere ed aggiungere dello spazio vuoto all'inizio ed alla fine delle registrazioni, con il fine di arrivare ad avere dati della stessa lunghezza.

Dal punto di vista della programmazione il linguaggio utilizzato è python, dove la libreria sfruttata per la maggior parte delle trasformazioni sui dati audio è stata Librosa [8], mentre per la parte riguardante il machine learning è stata utilizzata PyTorch [22]. La repository Github del progetto comprendente tutto il codice utilizzato è disponibile all'indirizzo [24]

I modelli sono stati allenati tramite cross-entropy loss e mediante l'uso dell'optimizer Adam, con un learning rate di 0.001. É stato utilizzato uno scheduler (StepLR) con step ogni 10 epoch e $gamma = 0.9$. Visto che i dataset sono generalmente non bilanciati, ovvero il numero di registrazioni per ogni classe non è costante, è stato adottato un sampler (WeightedRandomSampler) nel caricamento dei dataset per far sì che i modelli non andassero in overfit sulle classi con numeri più alti.

Vista la dimensione decisamente ridotta dei dataset utilizzati e nel tentativo di avere risultati più stabili riguardo all'accuratezza dei modelli, i dati di seguito presentati sono il risultato di una media tra un minimo di 5 esecuzioni per ogni configurazione.

5.1 TCN vs Conv-TasNet

Per prima cosa possiamo confrontare i due modelli per avere un'idea delle prestazioni di ognuno e del tempo e memoria necessari per allenarli. Questo confronto è stato condotto tramite il dataset RAVDESS.

Possiamo vedere nella tabella 5.2 i risultati migliori ottenuti dopo alcune esecuzioni: la Conv-TasNet è stata allenata tramite i parametri mostrati nella tabella 5.1, il significato di questi è esposto in [16], con la differenza che il parametro C ora indica il numero di classi presenti, quindi 8. É stato necessario adottare un batch size piuttosto ridotto (32) in quanto la rete utilizza molta memoria GPU. La TCN è stata allenata utilizzando la configurazione con 5 blocchi e 2 ripetizioni, mediante l'uso di spettrogrammi Mel a 40 canali.

N	32	P	3
L	20	X	5
B	256	R	2
H	32	C	8

Tabella 5.1: Parametri usati per la Conv-TasNet

Modello	Speaker Dependent		Speaker Independent	
	U. Accuracy	W. Accuracy	U. Accuracy	W. Accuracy
Conv-TasNet	51.2	50.7	44.4	43.6
TCN	**75.9**	**75**	**55.8**	**55**

Tabella 5.2: risultati Conv-TasNet e TCN

Come si può notare, la TCN raggiunge risultati decisamente migliori, inoltre
il tempo di training è molto ridotto rispetto alla Conv-TasNet (10 minuti rispetto a più di 40). Per questi motivi nel proseguo adotteremo il modello TCN come architettura migliore, proseguendo con ulteriori analisi.

5.2 RAVDESS

5.2.1 Preprocessing

Come abbiamo visto, una delle prime cose da stabilire per allenare un modello è il tipo di feature extraction da utilizzare. In accordo con i paper analizzati ho deciso di provare 3 possibilità: MFCC, Mel Spectrogram e Mel Spectogram con scala trasformata in decibel.

Come si vede dalla tabella 5.3, vengono raggiunti buoni risultati sia tramite MFCC che tramite lo spettrogramma mel con scala convertita in dB, attorno al 75% di accuracy nel test set per quanto riguarda le esecuzioni *speaker dependent*, mentre attorno al 58% per quanto riguarda le esecuzioni *speaker independent*. Gli spettrogrammi mel appaiono più stabili, ovvero esibiscono una minore (seppur lieve) disparità tra i risultati raggiunti tra le esecuzioni, quindi verranno utilizzati per le successive iterazioni nella variante che utilizza 40 features per blocco temporale.

metodo	# features	Speaker Dependent		Speaker Independent	
		U. Accuracy	W. Accuracy	U. Accuracy	W. Accuracy
mfcc	40	74	73,7	57,3	55,9
	128	75,5	74,6	57,3	**58,8**
spettrogramma mel	40	60,4	58,4	51,5	50,8
	128	55,1	55,7	51,9	50,7
spettrogramma mel con stala in dB	40	**75,9**	**75**	55,8	55
	128	74,3	73,6	**58,5**	56,9

Tabella 5.3: Risultati legati al tipo di preprocessing su RAVDESS

5.2.2 Dimensioni della rete

Un altro fattore che solitamente è piuttosto importante nel raggiungere buone prestazioni è quello delle dimensioni del modello utilizzato. Questo può influenzare molto la capacità di una architettura di generalizzare delle caratteristiche dei dati, oltre ad impattare in maniera importante sull'utilizzo di risorse quali la memoria necessaria e il tempo di allenamento. Nel grafico 5.1 vengono riassunti i risultati ottenuti al variare delle dimensioni della TCN nella configurazione speaker dependent: con *# ripetizioni* si intende il numero di volte che ogni blocco al cui interno troviamo le varie dilatazioni è ripetuto all'interno della rete, *# blocchi* indica il numero di blocchi convoluzionali dilatati (denominati con Blocco Conv1D nella struttura della TCN) presenti in ogni blocco menzionato prima.

Come possiamo vedere dal grafico, non sembra esserci una particolare correlazione tra le dimensioni del modello e la precisione raggiunta, ma solamente delle piccole variazioni. La stessa conclusione si può trarre per il caso speaker dependent. Nel proseguo verrà utilizzata una rete di dimensioni intermedie (la stessa utilizzata finora), formata da 5 blocchi e 2 ripetizioni.

5.2.3 Dropout

Nelle varie esecuzioni abbiamo notato un problema persistente: il modello tende sempre ad andare incontro ad *overfitting*. Questa situazione si verifica quando vi è una grande discrepanza tra i risultati ottenuti nel training set e quelli del validation/test set, spesso con un peggioramento di questi ultimi nel tempo. Questo indica che il modello non sta più imparando caratteristiche generali del problema

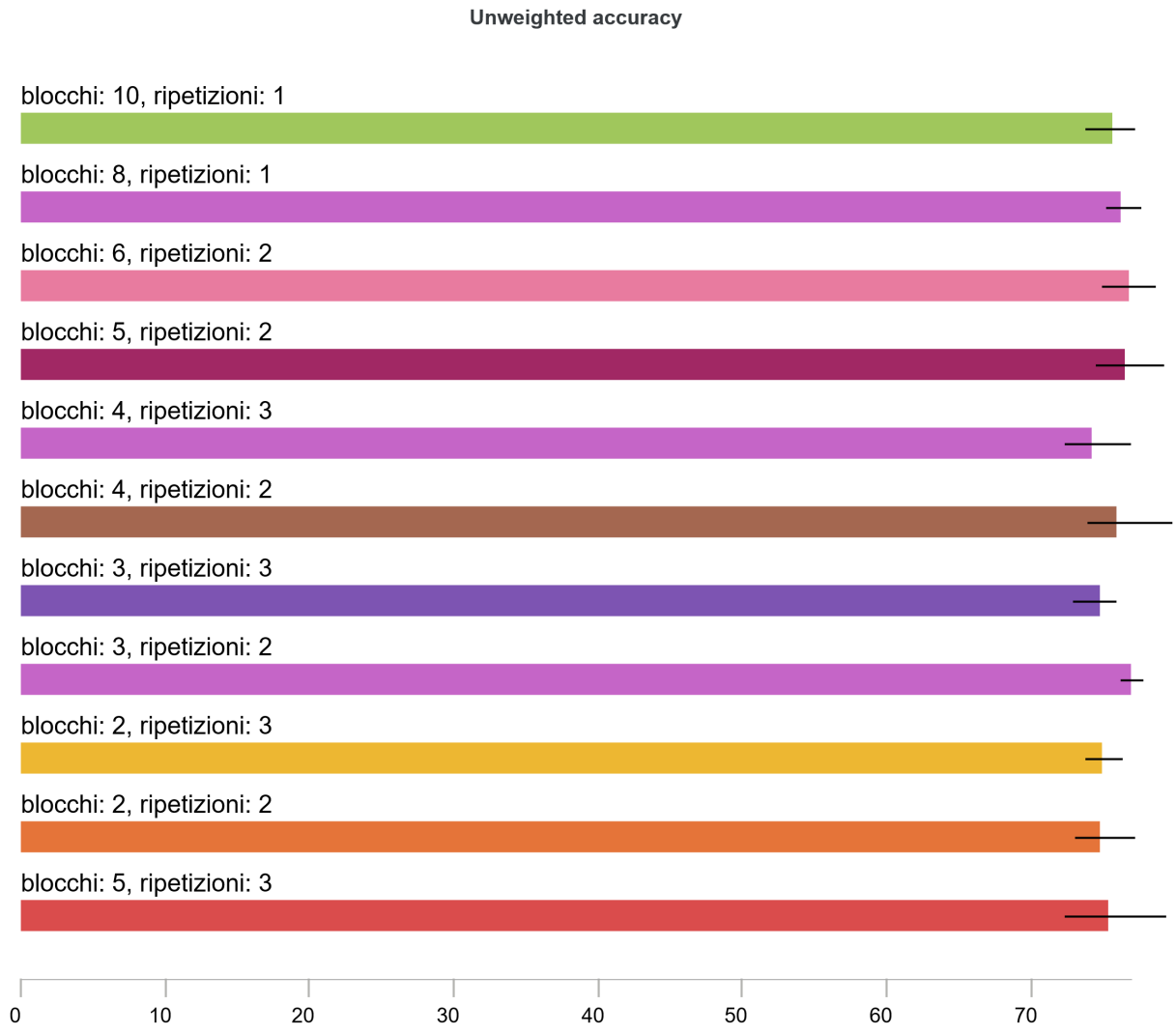


Figura 5.1: Unweighted Accuracy con varie dimensioni della rete

che stiamo analizzando ma che esso si sta specializzando a riconoscere caratteristiche particolari dei singoli esempi dati come input.

Per cercare di ridurre questo problema ed avere un risultato migliore, o perlomeno più stabile ho applicato due metodi:

- dropout: questa tecnica consiste nell'inserire nel modello alcuni strati che andranno ad azzerare casualmente alcuni output per lo strato precedente, con un certa probabilità. Questo permette di diminuire l'overfitting e sembra condurre ad una miglior generalizzazione.

- caricamento casuale dei segmenti: applicando questa variazione i segmenti che vengono dati come input alla rete sono più piccoli rispetto alle registrazioni complete e l'inizio del segmento viene generato in modo casuale. Questo permette di allenare la rete su segmenti diversi, riducendo leggermente l'overfitting.

Nella tabella 5.4 vengono riassunti i risultati ottenuti applicando vari livelli di dropout al modello utilizzato (TCN), inoltre i grafici 5.2 e mostrano l'andamento di accuracy e loss per le esecuzioni con e senza dropout.

Come si può vedere, il valore ideale di dropout per questo dataset e configurazione appare essere $p = 0.2$, ovvero ogni unità in un determinato strato precedente al dropout ha una probabilità del 20%

livello di dropout (p)	Speaker dependent		Speaker Independent	
	U. Accuracy	W. Accuracy	U. Accuracy	W. Accuracy
0.1	76,7	74,8	59,4	58,4
0.2	**77,6**	**77,2**	**60,7**	58,9
0.3	76,4	76,1	60,1	**59,1**
0.4	73,1	73,3	59	57,6
0.5	71,7	72,1	57,2	54,9
0.6	68,3	69	57,6	56,2

Tabella 5.4: Risultati legati a vari livelli di dropout

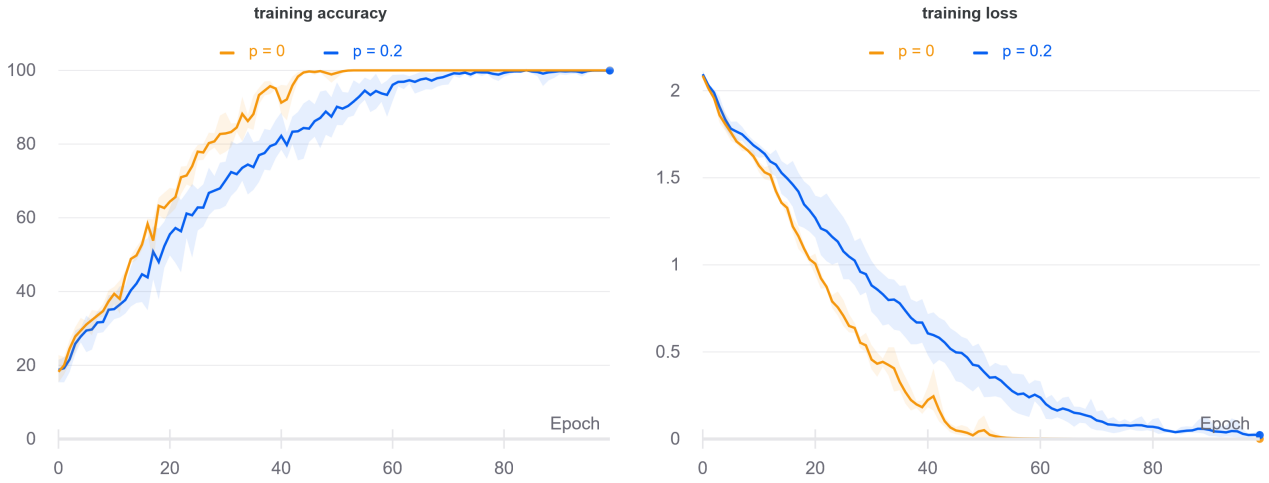


Figura 5.2: Accuracy e loss nel training set con e senza dropout

di essere azzerato. Questo è un risultato coerente con quello che ci si può aspettare, in quanto livelli troppo alti di dropout, come si può vedere nel caso $p = 0.6$, influiscono negativamente sulle prestazioni complessive, rendendo difficile l'allenamento.

Per quanto riguarda il caricamento casuale dei segmenti, questo sembra influire in maniera marginale sulle performance, come pure limitato è il miglioramento riguardo all'overfitting.

5.2.4 Aggiunta di rumore

Come già esposto brevemente nella sezione 2.1, uno dei modi in cui a volte è possibile migliorare la qualità delle classificazioni è tramite l'aggiunta di rumore di fondo. Questo è diverso dal rumore che è già presente nelle registrazioni, in quanto ha un'intensità costante nel tempo, aiutando a nascondere dell'eventuale rumore residuo nelle registrazioni, che non essendo regolare potrebbe influire sulla classificazione. Nella tabella 5.5 è possibile vedere i risultati ottenuti applicando diversi livelli di rumore nelle registrazioni, con l'utilizzo della configurazione con $p_{dropout} = 0.2$.

Come è possibile notare confrontando le tabelle 5.5 e 5.4, l'aggiunta di un leggero rumore di fondo permette di migliorare di molto le prestazioni nel caso speaker independent, raggiungendo risultati migliori di 8 punti percentuali. Questo è probabilmente dovuto al fatto che il rumore aiuta a "mascherare" le caratteristiche peculiari di ogni attore uniformando le registrazioni. Inoltre migliorano anche le prestazioni nella configurazione speaker dependent, ottenendo un aumento di circa 3-4 punti

livello di rumore	Speaker Dependent		Speaker Independent	
	U. Accuracy	W. Accuracy	U. Accuracy	W. Accuracy
0.001	**81,2**	**80,5**	**68,8**	**67,9**
0.002	79,8	78,5	68,3	67,7
0.005	75,3	73,9	68,6	67,5
0.01	72.0	69,8	63,4	63,2

Tabella 5.5: Risultati legati all'aggiunta di vari livelli di rumore

percentuali.

5.2.5 altri tipi di data augmentation

Altre tecniche possono essere adottate per cercare di ottenere risultati migliori, quelle che ho deciso di indagare sono:

- *shift* dei segmenti: spostare il punto di inizio di un segmento, a volte aggiungendo parti vuote
- modifica del tono percepito
- modifica della velocità di riproduzione
- aggiunta di riverbero
- *spec augment*: consiste nel rimuovere parti dello spettrogramma in modo casuale (bande orizzontali e verticali) per poter generare modelli più generali. Segue lo stesso ragionamento del dropout.

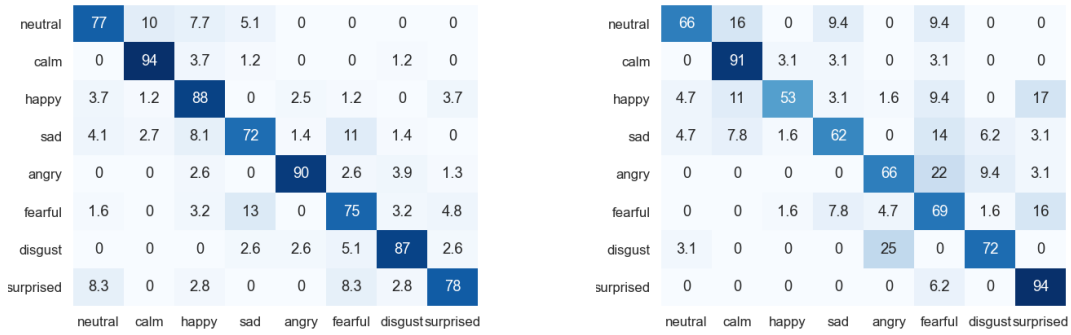
Modifica	Speaker Dependent		Speaker Independent	
	U. Accuracy	W. Accuracy	U. Accuracy	W. Accuracy
Shift	75.0	74.1	64.8	63.9
Tono	74.5	73.3	61.7	62.9
Velocità	71.3	70.1	60.9	59.2
Riverbero	66.0	64.8	54.2	52.9
Spec augment	65.3	64.0	54.3	53.8

Tabella 5.6: Risultati riguardanti altre modifiche

Come si può vedere dalla tabella 5.6, applicando queste modifiche non si nota alcun miglioramento nelle prestazioni della TCN, ma anzi vi è un sostanziale peggioramento, soprattutto per quanto riguarda il riverbero e lo spec augment.

5.2.6 Confronto con lo stato dell'arte

Nella figura 5.3 sono presenti le matrici di confusione per il modello migliore che è stato sviluppato nei casi speaker dependent e speaker independent, dove nell'asse delle ascisse troviamo le predizioni e



(a) Speaker Dependent

(b) Speaker Independent

Figura 5.3: Matrici di confusione per i modelli migliori con RAVDESS

su quello delle ordinate le classi corrette; le tabelle sono normalizzate secondo quest'ultime, quindi in senso orizzontale. Possiamo notare che solamente la classe *calm* sia classificata in maniera piuttosto solida in entrambe le tabelle, con una notevole differenza per le classi *happy* e *angry*.

Il grafico 5.4 mostra una comparazione dei risultati migliori con altri paper e con la precisione umana riportata, come si può vedere la struttura TCN proposta ha prestazioni al livello dell'attuale stato dell'arte, inoltre in questo documento sono presenti entrambe le accuracy per il caso speaker dependent e independent, che spesso non sono presenti insieme in bibliografia. I tre paper riportati sono dall'alto al basso [14, 29, 20]

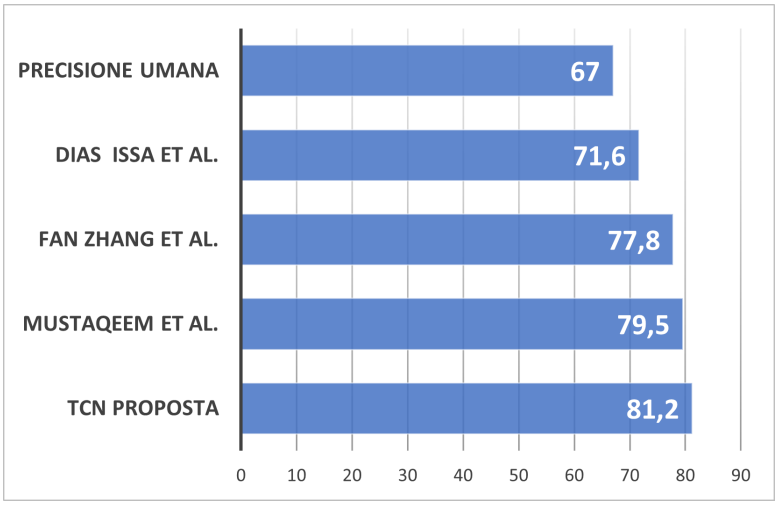


Figura 5.4: Confronto con lavori passati su RAVDESS

5.3 IEMOCAP

Per quanto riguarda il dataset IEMOCAP sono state eseguite solo delle esecuzioni limitate, per il modello la configurazione utilizzata è quella che ha ottenuto i risultati migliori con RAVDESS descritta nella sezione precedente. Il dataset è stato suddiviso per la configurazione speaker independent in 4 sessioni di train e una di validation/test, mentre con la stessa divisione 70-10-20% per la configurazione speaker independent. Come viene evidenziato in alcuni paper come [13, 14, 32], le prestazioni del modello sono decisamente migliori quando si utilizzano solo le registrazioni marcate come improvvisate. Le emozioni che sono state considerate sono *happy*, *sad*, *neutral*, *angry* ed *excited*, con *happy* ed *excited* unite a formare una singola classe.

La precisione del 70% è piuttosto lontana dallo stato dell'arte per questo dataset, che come detto si aggira attorno al 85% in [5], tuttavia dimostra che la struttura proposta è in grado di essere estesa ad altri comprensori con successo.

Dataset	Speaker Dependent		Speaker Independent	
	U. Accuracy	W. Accuracy	U. Accuracy	W. Accuracy
intero	62.7	64.3	57.8	**61.7**
solo improvvisati	**70.2**	**70**	**60.5**	61.1

Tabella 5.7: Riassunto risultati con dataset IEMOCAP

5.4 CREMA-D

I risultati per il dataset CREMA-D sono presenti nella tabella che segue (5.8), anche questi sono stati ottenuti mediante la configurazione che è stata trovata essere la migliore per RAVDESS, la divisione delle registrazioni è analoga per quanto riguarda il caso speaker dependent, mentre per il caso speaker independent vengono adottati 65 attori per il training, 8 per la validation e 18 per il test.

Speaker Dependent		Speaker Independent	
U. Accuracy	W. Accuracy	U. Accuracy	W. Accuracy
64.8	64.4	61.6	61.2

Tabella 5.8: Risultati con CREMA-D

Anche qui la accuracy è piuttosto bassa rispetto allo stato dell'arte riscontrato in *X-vectors meet emotions: A study on dependencies between emotion and speaker recognition* [21] che ottiene un risultato del 81.5%.

5.5 Dataset incrociati

Molti dei paper riguardanti il riconoscimento delle emozioni applica le metodologie sviluppate ad un singolo dataset, ma nelle applicazioni reali è necessario avere dei modelli robusti che siano in grado di superare alcune sfide come la varietà degli oratori, delle culture e delle lingue parlate. Per questi motivi può essere utile valutare le prestazioni della metodologia proposta in modalità *cross dataset*, quindi allenando le reti su di un dataset e testandole su di un altro.

La tabella 5.9 riassume i risultati per questo tipo di esecuzioni, le emozioni considerate sono quelle comuni, quindi 4 per run comprendenti IEMOCAP e 6 per le altre.

train\test	RAVDESS	IEMOCAP	CREMA-D
RAVDESS	/	42.0	35.3
IEMOCAP	41.1	/	**56.4**
CREMAD	32.2	47.6	/

Tabella 5.9: Risultati cross-dataset

5.6 Dataset "in the wild"

I dataset di cui abbiamo parlato sono tutti stati creati da attori, che quindi tendono ad esagerare il contenuto emotivo delle registrazioni. Vi sono anche altri comprensori che invece utilizzano registra-

zioni da altre fonti, come per esempio CMU-MOSEI [30]. Applicando la struttura proposta a questo dataset essa non sembra essere in grado di generalizzare bene le informazioni presenti.

Inoltre questo dataset è etichettato in maniera *multi-label*, quindi ogni registrazione può essere associata a più di una emozione, come a nessuna. Questo riconduce il task ad una classificazione binaria, con il problema che essendo il dataset sbilanciato la accuracy base predicendo tutte le emozioni come non presenti è attorno al 80%. Per avere delle metriche più significative è stato usato l'F1 score, che risulta essere attorno al 40% nel test set.

La mancanza di generalizzazione è probabilmente dovuta in parte anche alla relativa semplicità del modello e alla forte tendenza all'overfitting di quest'ultimo.

6 Conclusioni

In questo lavoro viene proposto di applicare una struttura detta *Temporal Convolutional Network* (TCN) per il riconoscimento delle emozioni dall'audio. Abbiamo visto che questo modello risulta essere molto valido per questo incarico, raggiungendo addirittura lo stato dell'arte per il dataset RAVDESS con una precisione superiore al 80%, mentre esso raggiunge risultati leggermente inferiori per gli altri dataset sperimentati, ovvero IEMOCAP e CREMA-D. É però doveroso ricordare che sono state svolte solo esecuzioni limitate con questi ultimi, quindi è molto probabile che si possano raggiungere prestazioni migliori. É possibile affermare che il modello sia piuttosto robusto, infatti esso è in grado di riconoscere, sebbene con precisioni inferiori, emozioni espresse in un dataset diverso da quello usato per l'allenamento. Abbiamo verificato inoltre come la fase di preprocessing e feature extraction sia molto importante in questo tipo di task, permettendo tramite l'aggiunta di un leggero rumore di fondo di migliorare di molto le prestazioni e di ridurre considerevolmente il divario tra esecuzioni speaker dependent e independent.

Ricordiamo l'importanza di questa tecnologia, con molteplici applicazioni dirette in ambienti sanitari e residenze per anziani, inoltre questa potrà essere applicata ad un vasto spettro della robotica e in generale nel campo della interazione uomo-macchina.

6.1 Sviluppi futuri

Questo è un campo di ricerca aperto e ci si può aspettare che molte innovazioni possano ancora venire attuate nei prossimi anni. Limitandosi alle strutture e metodologie adottate, alcuni aspetti che potrebbe valere la pena indagare sono:

- L'integrazione con un sistema multimodale è sicuramente uno dei prossimi passi nella ricerca in questo campo, infatti è possibile notare come nella grande maggioranza dei casi questo sia un fattore incisivo che permette di migliorare notevolmente le prestazioni, riducendo in maniera significativa gli errori e permettendo di avere una ridondanza di informazioni utili nei casi in cui alcune modalità non siano disponibili.

- La struttura Conv-TasNet potrebbe rivelarsi una valida alternativa alla TCN tramite l'utilizzo di alcuni livelli differenti, in particolare potrebbe risultare positivo l'utilizzo di strati convoluzionali nella parte successiva alla TCN, dove al momento sono presenti strati lineari.

- Potrebbero venire applicate delle tecniche per la generazione di altri speaker, questo potrebbe essere usato in più modi: per prima cosa come data augmentation, quindi per avere dei dati per l'allenamento dei modelli che siano più vari, dall'altro lato potrebbero essere adottati per ricondurre in maniera sistematica il caso speaker independent a quello speaker dependent, tramite la trasformazione delle registrazioni in tracce audio che sembrano essere pronunciate tutte dalla stessa persona. Quest'ultima tecnica potrebbe essere adottata in tutti i sistemi che lavorano con dati audio e non solo in questo campo specifico.

- Modifiche come quelle proposte nella sezione 5.2.5 o simili potrebbero rivelarsi positive se attuate in maniera diversa, oppure potrebbero rivelarsi efficaci anche in questo caso quelle adottate in [9] o [33].
- Altre sperimentazioni possono venire condotte sia per IEMOCAP e CREMA-D, sia per altri dataset come CMU-MOSEI, MELD, BAUM2 o altri.

Bibliografia

[1] Mehmet Berkehan Akçay and Kaya Oğuz. Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. *Speech Communication*, 116:56–76, 2020.

[2] J. Ancilin and A. Milton. Improved speech emotion recognition with mel frequency magnitude coefficient. *Applied Acoustics*, 179:108046, 2021.

[3] Chaim Baskin, Natan Liss, Avi Mendelson, and Evgenii Zheltonozhskii. Streaming architecture for large-scale quantized neural networks on an fpga-based dataflow platform. *CoRR*, abs/1708.00052, 2017.

[4] C. Bussoand M. Bulut, C.C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J.N. Chang, S. Lee, and S.S. Narayanan. Iemocap: Interactive emotional dyadic motion capture database. *Journal of Language Resources and Evaluation*, December 2008.

[5] Wenliang Dai, Samuel Cahyawijaya, Zihan Liu, and Pascale Fung. Multimodal end-to-end sparse model for emotion recognition, 2021.

[6] Jean-Benoit Delbrouck, Noé Tits, Mathilde Brousmiche, and Stéphane Dupont. A transformer-based joint-encoding for emotion recognition and sentiment analysis. *Second Grand-Challenge and Workshop on Multimodal Language (Challenge-HML)*, 2020.

[7] Moataz El Ayadi, Mohamed S. Kamel, and Fakhri Karray. Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern Recognition*, 44(3):572–587, 2011.

[8] Brian McFee et al. Librosa: audio and music processing in python, https://librosa.org/.

[9] Caroline Etienne, Guillaume Fidanza, Andrei Petrovskii, Laurence Devillers, and Benoit Schmauch. Speech emotion recognition with data augmentation and layer-wise learning rate adjustment. *CoRR*, abs/1802.05630, 2018.

[10] Haytham M. Fayek, Margaret Lech, and Lawrence Cavedon. Evaluating deep learning architectures for speech emotion recognition. *Neural Networks*, 92:60–68, 2017. Advances in Cognitive Engineering Using Neural Networks.

[11] Cao H, Cooper DG, Keutmann MK, Gur RC, Nenkova A, and Verma R. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing*, 2014.

[12] Kun Han, Dong Yu, and Ivan Tashev. Speech emotion recognition using deep neural network and extreme learning machine. *INTERSPEECH-2014*, pages 223–227, 2014.

[13] Ngoc-Huynh Ho, Hyung-Jeong Yang, Soo-Hyung Kim, and Gueesang Lee. Multimodal approach of speech emotion recognition using multi-level multi-head fusion attention-based recurrent neural network. *IEEE Access*, 8:61672–61686, 2020.

[14] Dias Issa, M. Fatih Demirci, and Adnan Yazici. Speech emotion recognition with deep convolutional neural networks. *Biomedical Signal Processing and Control*, 59:101894, 2020.

[15] Colin Lea, René Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks: A unified approach to action segmentation. *CoRR*, abs/1608.08242, 2016.

[16] Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8):1256–1266, Aug 2019.

[17] Shuiyang Mao, P. C. Ching, and Tan Lee. Enhancing segment-based speech emotion recognition by deep self-learning, 2021.

[18] Seyedmahdad Mirsamadi, Emad Barsoum, and Cha Zhang. Automatic speech emotion recognition using recurrent neural networks with local attention. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2227–2231, 2017.

[19] Trisha Mittal, Uttaran Bhattacharya, Rohan Chandra, Aniket Bera, and Dinesh Manocha. M3er: Multiplicative multimodal emotion recognition using facial, textual, and speech cues. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02):1359–1367, Apr. 2020.

[20] Mustaqeem and Soonil Kwon. A cnn-assisted enhanced audio signal processing for speech emotion recognition. *Sensors*, 20(1), 2020.

[21] Raghavendra Pappagari, Tianzi Wang, Jesus Villalba, Nanxin Chen, and Najim Dehak. X-vectors meet emotions: A study on dependencies between emotion and speaker recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7169–7173, 2020.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[23] K. Sreenivasa Rao, V. Ramu Reddy, and Sudhamay Maity. *Language Identification Using Spectral and Prosodic Features*. Springer, 2015. Appendix B, page 89.

[24] Thomas Rigoni. Audio emotion recognition ravdess, https://github.com/thomasrigoni7/audio-emotion-recognition-ravdess.

[25] Spring: Socially pertinent robots in gerontological healthcare, https://spring-h2020.eu/.

[26] Livingstone SR and Russo FA. Ravdess: Ryerson audio-visual database of emotional speech and song. https://zenodo.org/record/1188976, April 2018.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[28] Thurid Vogt. Real-time automatic emotion recognition from speech, 2010.

[29] Mingke Xu, Fan Zhang, and Wei Zhang. Head fusion: Improving the accuracy and robustness of speech emotion recognition on the iemocap and ravdess dataset. *IEEE Access*, 9:74539–74549, 2021.

[30] AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2236–2246, 2018.

[31] Jianfeng Zhao, Xia Mao, and Lijiang Chen. Speech emotion recognition using deep 1d & 2d cnn lstm networks. *Biomedical Signal Processing and Control*, 47:312–323, 2019.

[32] Ziping Zhao, Qifei Li, Zixing Zhang, Nicholas Cummins, Haishuai Wang, Jianhua Tao, and Björn W. Schuller. Combining a parallel 2d cnn with a self-attention dilated residual network for ctc-based discrete speech emotion recognition. *Neural Networks*, 141:52–60, 2021.

[33] Xinyue Zhu, Yifan Liu, Zengchang Qin, and Jiahong Li. Data augmentation in emotion classification using generative adversarial networks. *CoRR*, abs/1711.00648, 2017.