



SAPIENZA
UNIVERSITÀ DI ROMA

Un Approccio alla Voice Conversion a Spettro Ridotto attraverso la Sine-Wave Speech

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Laurea Triennale in Informatica

Davide Gabrielli

Matricola 1883616

Relatore

Prof. Danilo Avola

Correlatore

Prof. Luigi Cinque

Dr. Daniele Pannone

Anno Accademico 2021/2022

Un Approccio alla Voice Conversion a Spettro Ridotto attraverso la Sine-Wave Speech

Tesi di Laurea Triennale. Sapienza Università di Roma

© 2022 Davide Gabrielli. Tutti i diritti riservati

Questa tesi è stata composta con L^AT_EX e la classe Sapthesis.

Email dell'autore: gabrielli.1883616@studenti.uniroma1.it

*Ai miei genitori e mia sorella,
che mi hanno da sempre supportato
e dovuto ascoltare ogni mia spiegazione non richiesta di tutto questo.*

*A Fiorella,
che mi ha insegnato a scrivere in italiano, o che almeno ci ha provato,
e mi ha detto sempre le cose giuste per tranquillizzarmi.*

*Ai miei amici,
che mi hanno sostenuto
e dato modo di vedere ogni tanto luci diverse da quelle del monitor.*

*Al mio gatto Luna,
che mi ha sempre fatto compagnia nelle notti insonni a scrivere
(a differenza di Maya che se la dormiva).*

Sommario

L'idea per cui sia possibile disaccoppiare il contenuto linguistico dall'identità vocale è ricorrente all'interno di varie aree dello speech processing. Si può infatti considerare la voce come la somma di due componenti: una acustica, dipendente dall'interlocutore, e una linguistica, indipendente dall'interlocutore.

Questa tesi ha lo scopo di introdurre nel campo della voice conversion un approccio che sfrutta codifiche audio a spettro ridotto al fine di ridurre la componente acustica. Per tale scopo verranno impiegate delle rappresentazioni in sine-wave speech e in vocoded speech, che in letteratura scientifica hanno dimostrato di preservare l'intelligibilità nonostante la forte alterazione dello spettro sonoro.

Indice

1 Introduzione	1
1.1 Ambito e scopo dell'applicazione	1
1.2 Stato dell'arte	2
1.2.1 Dati paralleli	2
1.2.2 Dati non paralleli	2
1.3 Contributo	4
2 Audio	5
2.1 Il suono	5
2.2 Onda sinusoidale	5
2.3 Digitalizzazione segnale audio	6
2.4 Spettrogramma	6
2.5 Spettrogramma mel	7
2.6 Coefficienti mel-frequency cepstrum	8
2.7 Sine-wave speech	9
2.8 Vocoded speech	10
3 Deep Learning	11
3.1 Introduzione	11
3.2 Percettrone	11
3.3 Funzione di attivazione	11
3.4 Rete neurale artificiale	13
3.5 Algoritmi di apprendimento	13
3.6 Reti neurali avanzate	14
3.6.1 CNN	14
3.6.2 GAN	15
3.6.3 CycleGAN	15
3.6.4 MelGAN	16
3.6.5 CycleGAN-VCs	17
4 Architettura proposta	21
4.1 Metodo	21
4.2 Modulo di riduzione dello spettro sonoro	21
4.2.1 Modulo di riduzione a vocoded speech	21
4.2.2 Modulo di riduzione a SWS	22
4.3 Modulo di conversione audio-spettrogrammi	22

4.4	Architettura della rete	23
5	Test e valutazioni	25
5.1	Dataset	25
5.2	Training	25
5.3	Metodi di valutazione	26
5.4	Risultati	26
6	Conclusioni	29
	Bibliografia	31

Capitolo 1

Introduzione

In questo capitolo si farà un'introduzione sull'importanza della voce, ponendo attenzione su cosa si intende per voice conversion e su cosa può comportare lo sviluppo di tali tecniche. Verrà inoltre analizzato lo stato dell'arte e descritto cosa si andrà a realizzare nell'ambito di questa tesi.

1.1 Ambito e scopo dell'applicazione

La voce è uno strumento comunicativo di cruciale importanza nelle relazioni umane in quanto ci consente di esprimere pensieri e di condividere informazioni con altri individui. Tuttavia essa non è unicamente informazione verbale, infatti oltre ad essere un veicolo per il contenuto linguistico che vogliamo esprimere è caratterizzata anche da aspetti acustici, come timbrica e intonazione. Queste ultime sono fondamentali poiché ci permettono di distinguere l'identità a cui corrisponde una determinata voce, tema che anch'esso ha sollevato recentemente molto interesse nel campo della ricerca sulla *speaker recognition*[10].

Con il termine *voice conversion* si fa riferimento a tutte quelle tecniche che permettono di trasformare la voce di una persona *sorgente* in un'altra voce *target* preservando il contenuto linguistico, ovvero andando a modificare unicamente le caratteristiche dipendenti dall'interlocutore (come formanti, frequenza fondamentale, intonazione, intensità e durata) mantenendo invece quelle indipendenti che rappresentano il contenuto effettivo.

La procedura tipica di VC si può riassumere in tre fasi fondamentali (Fig. 1.1)[23]:

1. **Analisi:** Si ottiene una rappresentazione intermedia del segnale originale che sia più facilmente manipolabile o che renda più semplice l'estrazione di feature.
2. **Trasformazione:** Si crea una mappatura tra le caratteristiche del segnale sorgente e quelle del segnale target e lo si trasforma.
3. **Ricostruzione:** Si inverte la rappresentazione intermedia, a cui è stata applicata la trasformazione, al fine di ottenere il risultato finale come audio riproducibile.

Risulta interessante notare come la VC non sia esclusivamente l'applicazione di una trasformazione ad un segnale ma includa anche delle rappresentazioni intermedie,

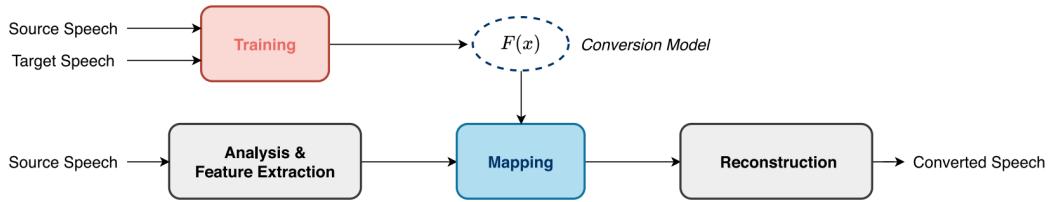


Figura 1.1. Pipeline tipica della voice conversion. Fonte [23].

il cui ruolo è cruciale in quanto potrebbero facilitare le operazioni di mappatura oppure causare artefatti e compromettere il risultato. L'interesse per la VC è dato dall'applicazione in vari campi come: sintesi vocale personalizzata, anonimizzazione voce e mimica vocale. Il tema è oggetto di ricerca nel campo della sintesi vocale e recentemente, grazie all'impiego di tecniche di deep learning, sono stati ottenuti importanti progressi.

1.2 Stato dell'arte

In questa sezione si riporta una classificazione dei metodi per la voice conversion che impiegano tecniche di deep learning, estratta dall'analisi svolta da Sisman et al.[23], distinguendo in due categorie differenti in base alla tipologia del dataset impiegato nella fase di training: impiego di dati paralleli e impiego di dati non paralleli.

1.2.1 Dati paralleli

I primi studi si sono focalizzati sull'impiego di dati paralleli, vale a dire impiegando dataset formati da audio di parole, o frasi, pronunciati da entrambi gli speaker dei quali si vuole la conversione. Il task è piuttosto semplice poiché avendo a disposizione audio corrispondenti e allineati, il problema si riduce a creare una mappatura tra questi.

Tuttavia ciò può risultare complesso e sebbene il problema dell'allineamento degli audio possa essere ovviato mediante l'impiego di encoder-decoder con meccanismo di attention[27], rimane comunque la problematica di avere a disposizione tali dati.

1.2.2 Dati non paralleli

Studi recenti hanno mostrato come siano possibili conversioni di voci anche con dati non paralleli, ovvero impiegando dataset che non presentano alcun audio di parole, o frasi, pronunciati da entrambi gli speaker di cui si vuole la conversione.

In base all'approccio usato per realizzare ciò si possono distinguere quattro categorie principali:

1. Dati non paralleli di una coppia di speaker definita
2. Impiego di sistemi Text-to-Speech
3. Impiego di sistemi di Automated Speech Recognition

4. Separazione del contenuto linguistico dallo speaker

Si procede dunque con un'analisi di quelle che sono le tecniche allo stato dell'arte per ciascuno di essi.

1) Dati non paralleli di una coppia di speaker definita Le tecniche attualmente impiegate per la voice conversion con dati non paralleli di una coppia di speaker definita sono le stesse impiegate nell'*'image-to-image translation'*, il cui obiettivo è trovare una mappatura da un dominio X ad un dominio Y mantenendone la struttura. Infatti come nella image translation si può volere, ad esempio, convertire fotografie di paesaggi in quadri di Monet, mantenendone il contesto originale rappresentato, così nella conversione di voci si vuole trasformare la voce tra due speaker differenti mantenendone il contenuto linguistico.

Allo stato dell'arte abbiamo la CycleGAN-VC[31] proposta da Kaneko et al., in particolare con le sue varianti CycleGAN-VC2[12], CycleGAN-VC3[13] e MaskCycleGAN-VC[14] che verranno approfondite nel Capitolo 3.

2) Impiego di sistemi Text-to-Speech Uno degli aspetti importanti della voice conversion è preservare il contenuto linguistico dalla voce sorgente a quella di destinazione, la quale è una caratteristica in comune con i sistemi Text-to-Speech (TTS) capaci di generare audio sintetico partendo da trascrizioni date. Questi ultimi si basano principalmente su modelli encoder-decoder e pertanto risulta possibile sfruttarli per la conversione attraverso tecniche di transfer learning[30], condividendo la parte del decoder. Tuttavia la maggior parte di questi approcci richiede molti dati per la fase di addestramento del modello di TTS che non sempre sono disponibili.

3) Impiego di sistemi Automated Speech Recognition Gli approcci basati sul deep learning per la voice conversion richiedono grandi quantità di dati al fine di poter creare una rappresentazione latente che descriva il sistema fonetico.

Tuttavia sappiamo che la maggior parte dei sistemi di riconoscimento automatico del parlato (ASR) sono già stati addestrati su grandi dati e descrivono correttamente il sistema fonetico, pertanto risulta interessante sfruttare la rappresentazione latente di essi nella conversione di voci.

Un approccio particolarmente di successo consiste nel costruire un modello che sfrutti i phonetic posteriogram (PPG) estratti da un sistema di SI-ASR (speaker-independent ASR) al fine di creare una mappatura verso una voce target[26].

4) Separazione del contenuto linguistico dallo speaker Per separare il contenuto linguistico dallo speaker sono possibili vari approcci, tra i più efficaci si menzionano l'impiego di instance normalization[3], vector quantization[29] e l'utilizzo di auto-encoder[17].

Un auto-encoder impara a riprodurre l'output come il suo input, e per tale scopo deve costruire una rappresentazione latente intermedia. Questa codifica interna può essere vista come una forma compressa dell'input che mantiene tutte le informazioni necessarie per ricostruire il segnale originale in output. Questi approcci tuttavia tendono a generare audio di bassa qualità per via dell'eccessiva rimozione di informazione.

1.3 Contributo

Questo lavoro ha lo scopo di esplorare la conversione di voci con riduzione dello spettro sonoro. Partendo dall'idea per cui è possibile da un audio di una voce disaccoppiare la componente linguistica (indipendente dallo speaker) da quella acustica (dipendente dallo speaker), si vuole trovare una rappresentazione che riduca le caratteristiche di quest'ultima senza l'impiego di ulteriori dati o modelli pre-trained.

Per fare ciò sono state impiegate tecniche tradizionali di speech processing per il pre-processamento dei dati, ottenendo così delle rappresentazioni in *sine-wave speech*, *buzz vocoded* e *noise vocoded* degli stessi. Questi sono poi stati impiegati per addestrare un modello di tipo MaskCycleGAN-VC[14], stato dell'arte per quanto riguarda la voice conversion con dati non paralleli di coppie di speaker, e ne sono stati confrontati i risultati.

Capitolo 2

Audio

In questo capitolo verrà descritto il suono, le sue caratteristiche e le sue rappresentazioni digitali, partendo dalle più comuni fino ad arrivare a quelle utilizzate per questo lavoro.

2.1 Il suono

Il suono è un segnale acustico prodotto dalle vibrazioni di un corpo e dalla trasmissione di queste attraverso un mezzo, ad esempio l'aria. Di questo segnale è possibile misurare l'intensità, ovvero la variazione di pressione, l'unità di misura più comunemente utilizzata sono i decibel (dB).

Possiamo dividere i suoni in due classi: periodici e aperiodici. Per quanto riguarda i suoni periodici questi possono essere semplici, come delle onde sinusoidali, o complessi, risultato di più onde sinusoidali combinate. Mentre i suoni aperiodici non presentano pattern ripetitivi e possono essere continui, come il rumore, o transienti, come impulsi.

2.2 Onda sinusoidale

Al fine di poter descrivere al meglio cos'è il suono, è necessario definirlo nella sua forma più semplice: l'onda sinusoidale. Essa consiste in un suono periodico semplice, descrivibile con la seguente formula:

$$y(t) = A \cdot \sin(2\pi ft + \varphi) \quad (2.1)$$

I parametri fondamentali al fine di caratterizzare un'onda sinusoidale dunque sono:

1. **Frequenza (f):** La frequenza di un suono è un concetto che si basa sulla periodicità di un segnale ed è espresso come l'inverso del tempo che esso impiega a ripetersi, ovvero a compiere un ciclo ($f = 1/T$).
2. **Aampiezza (A):** L'ampiezza di un segnale è la misurazione della variazione di pressione dello stesso.
3. **Fase (φ):** La fase di un segnale rappresenta la quantità di frazione di un ciclo, quindi una completa oscillazione, compiuta dall'onda.

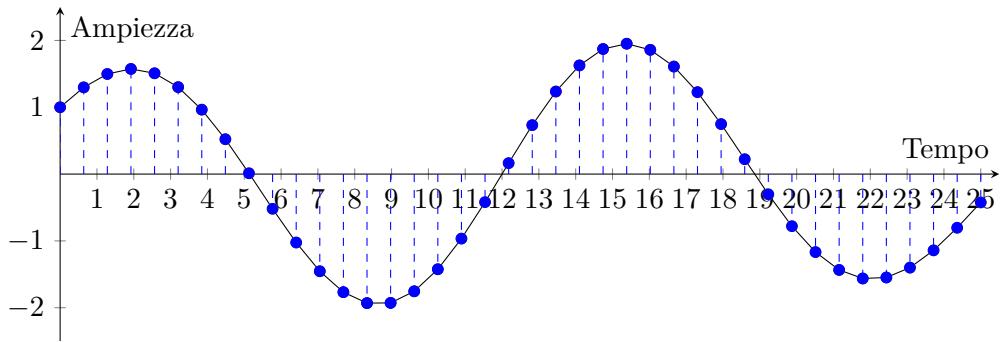


Figura 2.1. Campionamento di un segnale audio complesso. L’onda viene scomposta in campioni equidistanti che rappresentino l’ampiezza nel tempo, descrivendo l’onda originale.

2.3 Digitalizzazione segnale audio

Al fine di poter rappresentare digitalmente questi segnali analogici, abbiamo bisogno di un campionamento (Fig. 2.1). Esso consiste nel misurare e registrare la pressione sonora del segnale acustico ogni T secondi, tale valore viene definito come *intervallo di campionamento*. Il numero di campioni che registriamo in un secondo viene definita *frequenza di campionamento*.

Il teorema di Nyquist-Shannon definisce la frequenza massima rappresentabile f_N (frequenza di Nyquist) dato un campionamento con una frequenza f_s senza perdita di informazione.

$$f_N = \frac{1}{2} f_s \quad (2.2)$$

Nonostante questa tipologia di codifica dell’audio risulti ottimale per la conversione analogico-digitale dei segnali, non ci permette agilmente di ottenere quelle che sono le feature più interessanti per la nostra percezione.

2.4 Spettrogramma

Uno spettrogramma è una rappresentazione dello spettro di potenza di un segnale audio sul dominio del tempo (Fig. 2.3). Al fine di definire uno spettrogramma, è dunque necessario prima descrivere cosa sia uno spettro di potenza. Esso consiste nella rappresentazione di un segnale audio in un certo istante temporale t in funzione della frequenza.

Il metodo più veloce per calcolare lo spettro di potenza di un segnale audio è attraverso la *fast Fourier transform* (FFT), un algoritmo che calcola la *trasformata discreta di Fourier* (DFT) con costo computazionale $O(n \cdot \log(n))$ [28]. Calcolando la DFT sull’intero audio otteniamo dunque lo spettro dell’intero segnale audio analizzato (Fig. 2.2) ma non possiamo sapere in quale istante sono avvenuti i principali picchi.

Per ovviare a questo problema di mancanza di riferimento temporale, dobbiamo calcolare la DFT su porzioni di audio in sequenza temporale. Questa operazione viene definita *short-time fourier transform* (STFT) e possiamo così visualizzarne il risultato sotto forma di spettrogrammi (Fig. 2.3)[4].

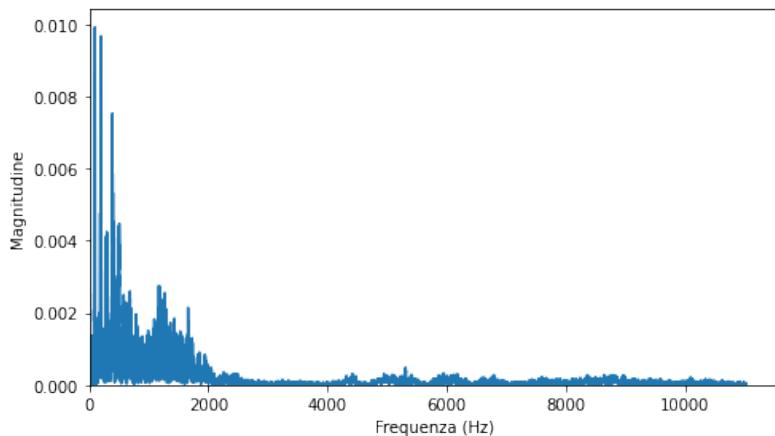


Figura 2.2. Spettro di potenza di una voce maschile pronunciando "It's not forever".

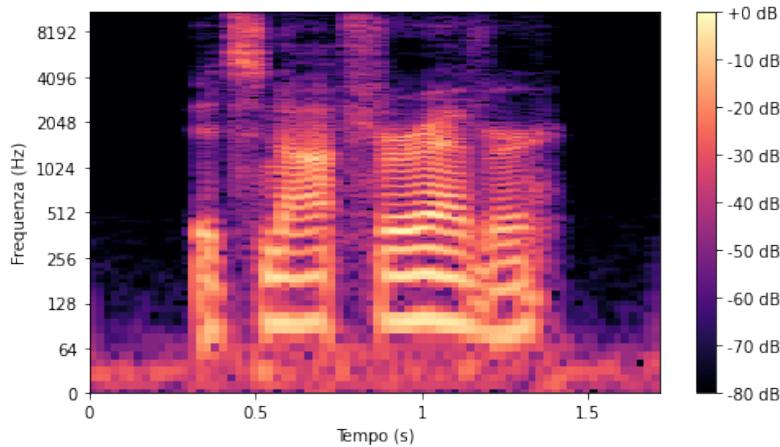


Figura 2.3. Spettrogramma di una voce maschile pronunciando "It's not forever".

Uno spettrogramma può essere rappresentato con una heat map in cui gli assi descrivono la frequenza e il tempo, mentre l'intensità sonora è espressa attraverso il colore. Questo tipo di codifica ci permette quindi di accedere ad informazioni più rilevanti riguardo il suono in uno spazio ristretto.

2.5 Spettrogramma mel

Tuttavia è importante considerare come la percezione umana del suono non sia lineare ma sia dipendente dall'altezza sonora (pitch) e quindi dalla frequenza dei suoni che udiamo. La scala mel viene per la prima volta sviluppata nel 1937 in una ricerca effettuata da Stevens e consiste in una scala di frequenze con altezze diverse (pitch) giudicate dagli ascoltatori come equidistanti (Fig. 2.4)[25]. Non esiste propriamente una formula per la scala di mel ma tra le più comuni troviamo:

$$mel(x) = 2595 \log_{10}\left(1 + \frac{x}{700}\right) \quad (2.3)$$

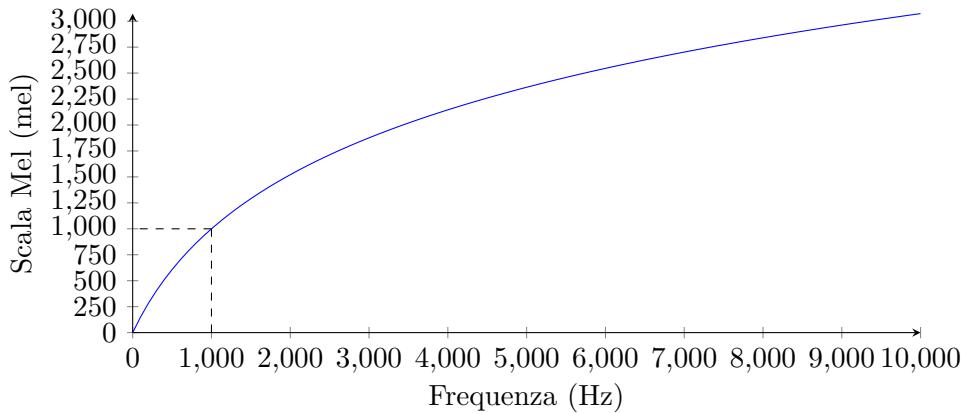


Figura 2.4. Grafico della scala mel.

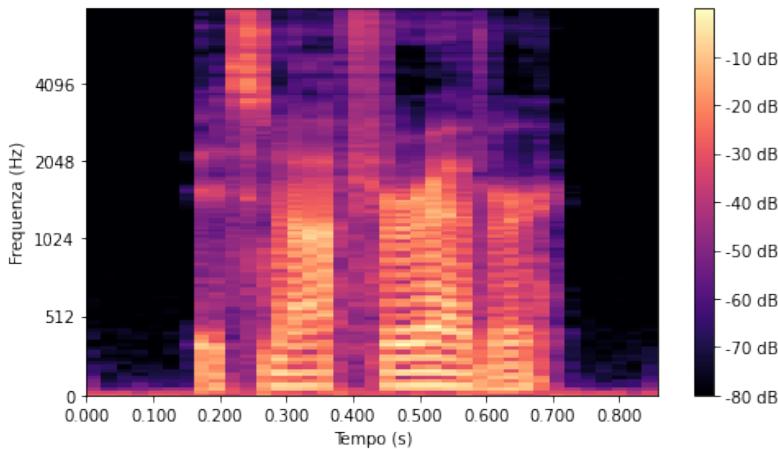


Figura 2.5. Spettrogramma mel di una voce maschile pronunciando "It's not forever".

Possiamo quindi impiegare questa scala per realizzare una rappresentazione che ci consentirà in fase di elaborazione di ottenere manipolazioni più significative per il nostro apparato uditivo (Fig. 2.5).

L'inversione degli spettrogrammi è soggetta alla produzione di artefatti in quanto nel calcolo della trasformata discreta di Fourier si ha perdita di informazione delle fasi dei segnali. Tuttavia grazie allo sviluppo recente di vocoder neurali, come MelGAN[16], in grado di rigenerare correttamente l'audio a partire da spettrogrammi mel, questi sono diventati sempre più di uso comune.

2.6 Coefficienti mel-frequency cepstrum

I coefficienti mel-frequency cepstrum (MFCC)[6] sono una rappresentazione dell'audio largamente usato nello speech processing tradizionale che ha trovato largo utilizzo anche nel machine learning, come nella speech recognition. Al fine di ottenere un MFCC è sufficiente calcolare la *trasformata discreta del coseno* (DCT) dello spettrogramma mel dell'audio desiderato (Fig. 2.6).

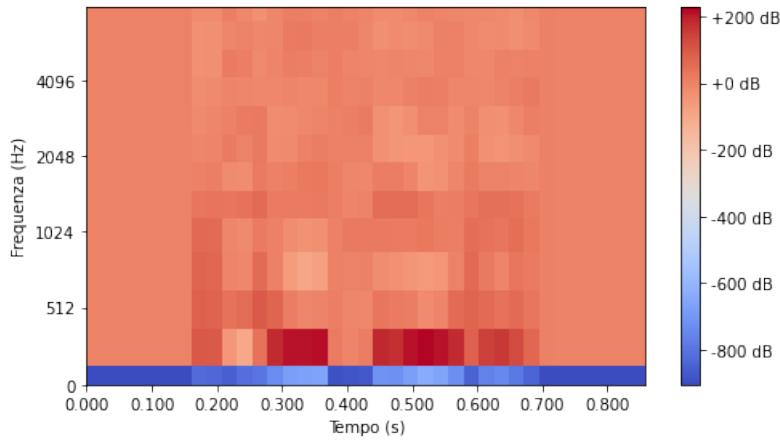


Figura 2.6. MFCC di una voce maschile pronunciando "It's not forever".

2.7 Sine-wave speech

La sine-wave speech (SWS) è una forma di audio del parlato umano a spettro ridotto, in cui vi sono presenti in genere tre o quattro componenti sinusoidali mobili (Fig. 2.7). L'audio viene generato rimpiazzando le formanti con delle sinusoidi al fine di rimuovere il più possibile la componente acustica mantenendo però intelligenza.

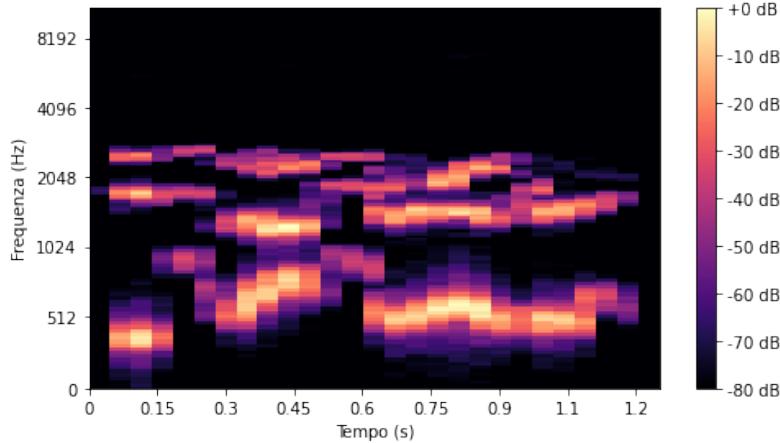


Figura 2.7. Spettrogramma mel di una voce maschile pronunciando "It's not forever" in SWS con 3 componenti.

Viene per la prima volta sviluppato negli Haskins Laboratories da Rubin[22] e successivamente impiegato in vari esperimenti riguardanti la percezione del parlato, come da Remez et al. che nel 1981 dimostrò come nonostante la sua apparente forma innaturale, la sine-wave speech conservi le proprietà sufficienti per la percezione del contenuto linguistico[20].

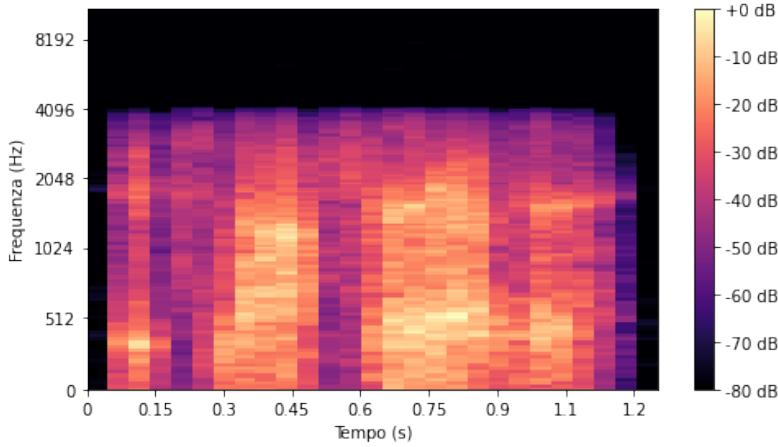


Figura 2.8. Vocoder con noise come carrier e una voce maschile pronunciando "It's not forever" come modulatore.

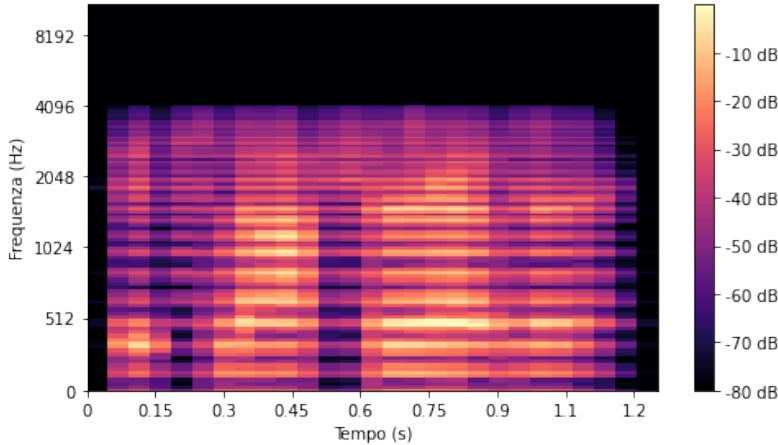


Figura 2.9. Vocoder con un buzz a 500 Hz come carrier e una voce maschile pronunciando "It's not forever" come modulatore.

2.8 Vocoded speech

Il vocoder è una tecnica di elaborazione dell'audio che richiede due sorgenti: un carrier, che accoglierà il suono, e un modulatore, che darà forma al suono del carrier. Le caratteristiche armoniche del modulatore vengono codificate all'interno del carrier, ottenendo un segnale con uno spettro alterato (Fig. 2.9 e 2.8).

Le ricerche sulla percezione del parlato in condizioni di alterazione spettrale trovano il loro interesse anche per questa forma di modulazione. Il lavoro di Davis et al. sulla comprensione del parlato noise-vocoded dimostra come anche in questo caso, l'ascoltatore impari a riconoscere il parlato nonostante la forte distorsione di esso[5].

Capitolo 3

Deep Learning

In questo capitolo sarà introdotto il deep learning, verranno descritte architetture di reti neurali semplici fino ad arrivare ad alcune più complesse utilizzate in questo lavoro.

3.1 Introduzione

Il deep learning è una branca del machine learning che si basa sull'apprendimento della rappresentazione dei dati. A differenza dei tipici approcci in cui si scrive un algoritmo per svolgere un'attività specifica, si realizzano invece degli algoritmi in grado di imparare a mappare dati tra domini differenti. Questa disciplina prende forte ispirazione dall'organizzazione del cervello che, attraverso diverse trasformazioni e rappresentazioni, riesce ad imparare a processare le informazioni.

3.2 Percettrone

Un percettrone è l'architettura neurale minima, introdotta nel 1958 da Rosenblatt[21], ed è ispirata al funzionamento del neurone, unità minima del cervello umano. Esattamente come un neurone esso infatti riceve dei segnali di ingresso e applica una somma pesata come segue:

$$y = w_0 + \sum_{i=1}^n x_i w_i \quad (3.1)$$

L'output y viene poi passato ad una funzione di attivazione che deciderà se il percettrone deve attivarsi o meno.

3.3 Funzione di attivazione

Una funzione di attivazione è una trasformazione non lineare applicata all'output di un percettrone al fine di mapparlo in un range di valori differente. Si descrivono a seguire alcune delle più note funzioni di attivazione.

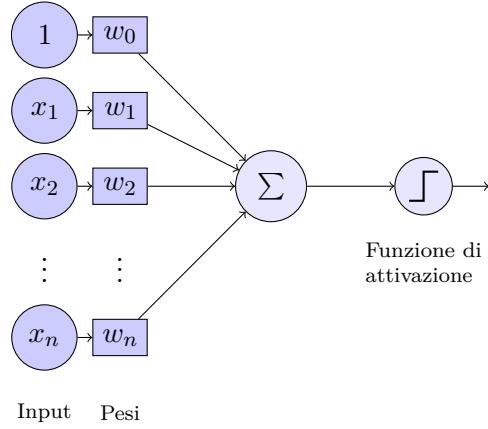


Figura 3.1. Un percettrone con una funzione di attivazione sull'output.

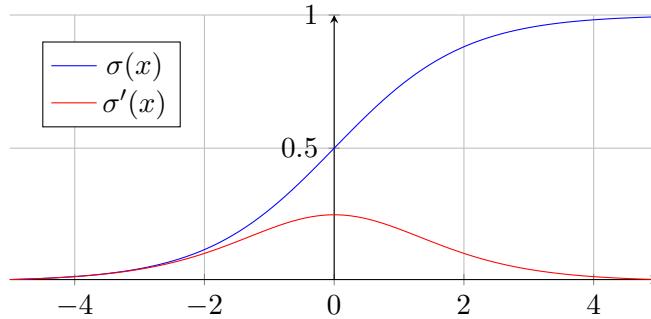


Figura 3.2. Grafico della funzione Sigmoid.

Sigmoid La funzione Sigmoid (Fig. 3.2) è stata storicamente la più usata tra le funzioni di attivazione. Essa offre il vantaggio di mappare i valori nell'intervallo (0,1).

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Tanh La funzione Tanh (Fig. 3.3) è anch'essa sigmoidea. La principale differenza che la distingue da Sigmoid è che il suo dominio di output è in (-1,1) che la rende particolarmente adatta nei problemi di classificazione con due classi.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

ReLU La funzione ReLU (Fig. 3.4) è il nuovo standard per molti tipi di reti neurali in quanto permette di abbassare notevolmente il costo computazionale dell'addestramento del modello. Essa è una funzione lineare definita a tratti, che mappa in zero tutti i numeri negativi e gli altri nell'input stesso[1].

$$\text{ReLU}(z) = \max(0, z)$$

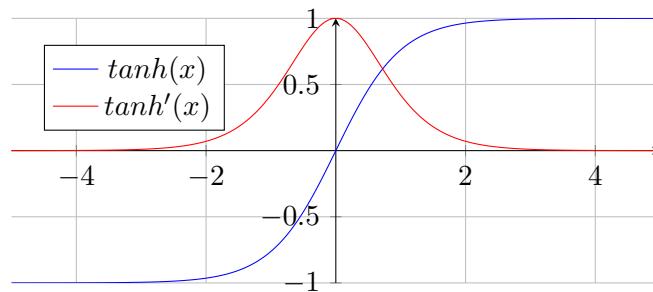


Figura 3.3. Grafico della funzione Tanh.

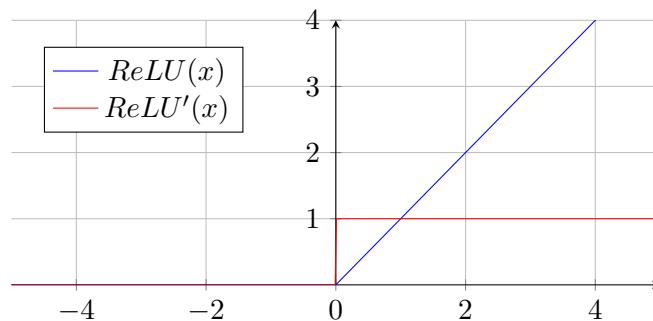


Figura 3.4. Grafico della funzione ReLU.

3.4 Rete neurale artificiale

I percetroni sono solitamente organizzati in livelli, che possono avere strutture diverse per effettuare trasformazioni differenti. Una rete neurale artificiale (ANN) è formata da più livelli interconnessi e organizzati come segue:

- **Livello di input:** ottiene i dati iniziali per la rete neurale.
- **Livelli nascosti:** livelli intermedi che svolgono la computazione.
- **Livello di output:** produce il risultato finale.

Il tipo di ANN più semplice è una Multilayer Perceptron (MLP), un'architettura fully connected e feedforward, ovvero in cui ogni nodo in uscita è connesso ad ogni altro nodo del livello successivo (Fig. 3.5).

3.5 Algoritmi di apprendimento

Gli algoritmi di machine learning si possono suddividere in tre categorie in base alle modalità di svolgimento della fase di addestramento[7]:

- **Supervised Learning:** paradigma applicabile laddove sia presente un dataset con elementi che abbiano un'etichetta applicata o che abbiano un corrispondente elemento obiettivo, utile ai fini di classificazione o regressione.

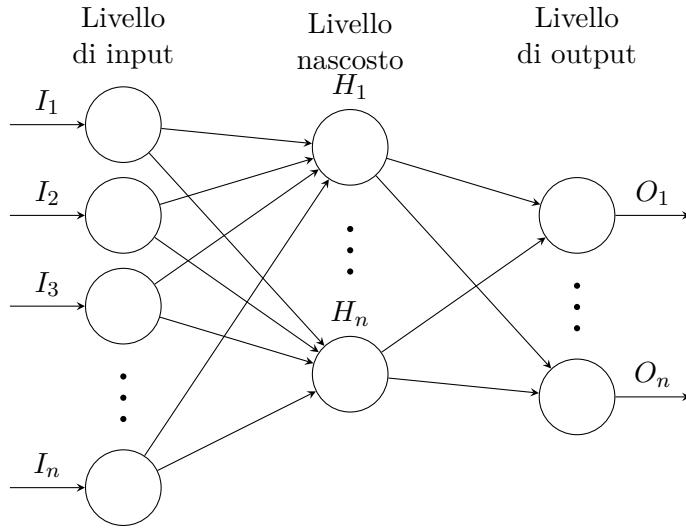


Figura 3.5. Grafo rappresentante una MLP.

- **Unsupervised Learning:** paradigma applicabile laddove sia utile imparare proprietà sulla struttura dei dati, come per la generazione o la riduzione del rumore da essi.
- **Semi-supervised Learning:** sfrutta sia tecniche del supervised learning, sia dell’unsupervised learning. Nello specifico si sfrutta la conoscenza fornita dai dati etichettati per classificare anche i restanti. Questa tecnica agevola la costruzione del dataset in quanto si possono usare più dati ma richiede comunque una prevalenza di dati etichettati.

3.6 Reti neurali avanzate

In questa sezione vengono descritte le principali reti neurali avanzate utilizzate in questo lavoro.

3.6.1 CNN

Una CNN (Convolutional Neural Network) è una tipologia di rete neurale artificiale comunemente utilizzata nella classificazione di immagini (Fig. 3.6)[18]. La rete è composta in genere da tre tipi di livelli:

- **Livelli convoluzionali:** applicano dei filtri alle immagini al fine di estrarre feature da esse.
- **Livelli di pooling:** riducono lo spazio di rappresentazione raccogliendo output vicini basandosi su calcoli statistici come media o massimo.
- **Livelli fully connected:** connettono tutti i nodi tra loro in modo da poter permettere la classificazione basandosi sulle feature estratte.

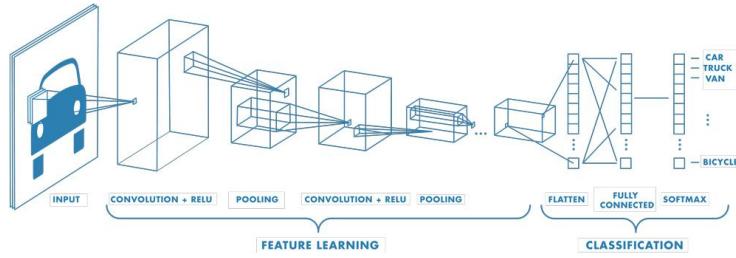


Figura 3.6. Convolutional Neural Network per la classificazione di immagini. Fonte <https://it.mathworks.com/discovery/convolutional-neural-network-matlab.html>.

3.6.2 GAN

Una GAN (Generative Adversarial Network) è una classe di algoritmi di deep learning proposta nel 2014 da Goodfellow[8]. Consiste in due reti neurali che competono in un gioco a somma zero (3.2), addestrando due modelli allo stesso tempo: un modello generativo (G) e un modello discriminatore (D). La rete generativa produrrà immagini che verranno passate in input alla rete discriminatrice mentre quest'ultima si occuperà di classificarle come reali o generate (Fig. 3.7).

$$\min_G \max_D V(D, G) = \min_G \max_D [\mathbb{E}_{x \sim p_{data}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log (1 - D(G(z)))]] \quad (3.2)$$

Il discriminatore è un classificatore che impara a distinguere dati reali da quelli creati dal generatore, questo sarà addestrato usando dati reali e l'output del generatore, e verrà penalizzato in caso classifichi scorrettamente i dati e aggiornando i suoi pesi attraverso backpropagation.

Il generatore è una rete che impara a generare immagini false che sembrino realistiche, questo sarà addestrato usando dati casuali e il suo output verrà collegato al discriminatore, e verrà penalizzato in caso esso rilevi l'immagine generata come falsa.

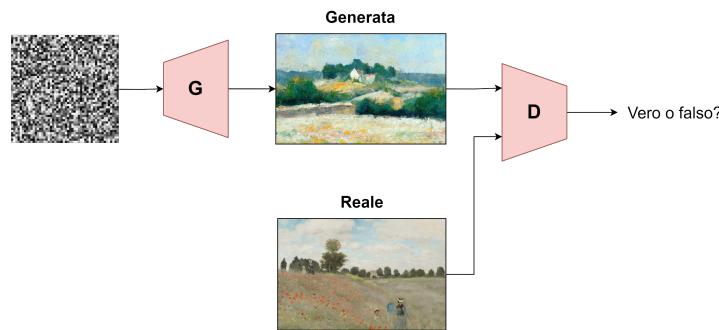


Figura 3.7. GAN che genera dipinti in stile Monet.

3.6.3 CycleGAN

La *image-to-image translation* è una classe di problemi nelle quali l'obiettivo è riuscire a imparare a mappare input e output di coppie di immagini. Tuttavia avere

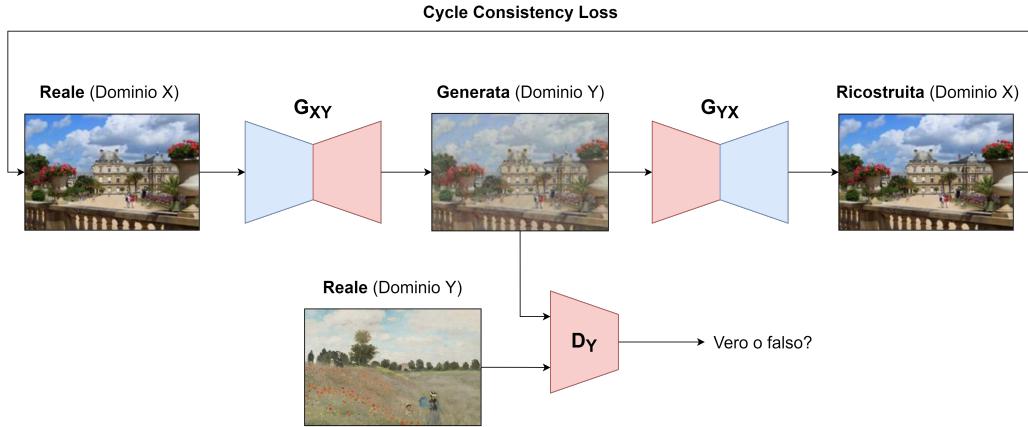


Figura 3.8. Ciclo forward della CycleGAN che trasforma foto in dipinti di Monet.

a disposizione questi dati paralleli non è sempre possibile. Basti pensare ad un convertitore di foto a dipinti, bisognerebbe dunque avere una serie di foto e dipinti corrispondenti e abbinati.

Nella image-to-image translation la GAN come input, anziché rumore casuale, prende un’immagine di un certo dominio X e cercherà di mapparla in un’immagine generata che abbia proprietà simili al dominio Y (immagini date di riferimento al discriminatore). Al fine di costruire un framework che permetta di lavorare con dati non paralleli, semplificando quindi la collezione del dataset, è stata progettata la CycleGAN[31].

Una CycleGAN è composta da 2 GAN, una per convertire dal dominio X al dominio Y e una per l’inverso, e dunque da 2 generatori e 2 discriminatori. Nella fase di training abbiamo due cicli che si alternano: un ciclo forward e uno backward. Nel ciclo forward (Fig. 3.8) viene selezionata un’immagine dal dominio X e viene trasformata dal primo generatore (G_{XY}) in un’immagine del dominio Y . Questa viene testata dal discriminatore che dovrà capire se sia reale o generata. La stessa immagine viene anche trasformata nuovamente dal secondo generatore (G_{YX}) in un’immagine del dominio X per poi essere confrontata impiegando una funzione di *Cycle Consistency Loss* con l’immagine originale.

Nel ciclo backward avviene la stessa operazione del ciclo forward ma partendo dal dominio Y e usando gli opportuni generatori e discriminatori come illustrato in Fig. 3.9.

3.6.4 MelGAN

Elaborare segnali audio è un processo complesso che richiede spesso il passaggio a rappresentazioni intermedie che possono portare artefatti e comprometterne i risultati. Nel 2019 viene proposto da Kumar un vocoder neurale per spettrogrammi mel basato su reti generative avversarie: MelGAN[16]. L’architettura è una fully convolutional network che prende in input spettrogrammi mel e fornisce come output l’audio corrispondente.

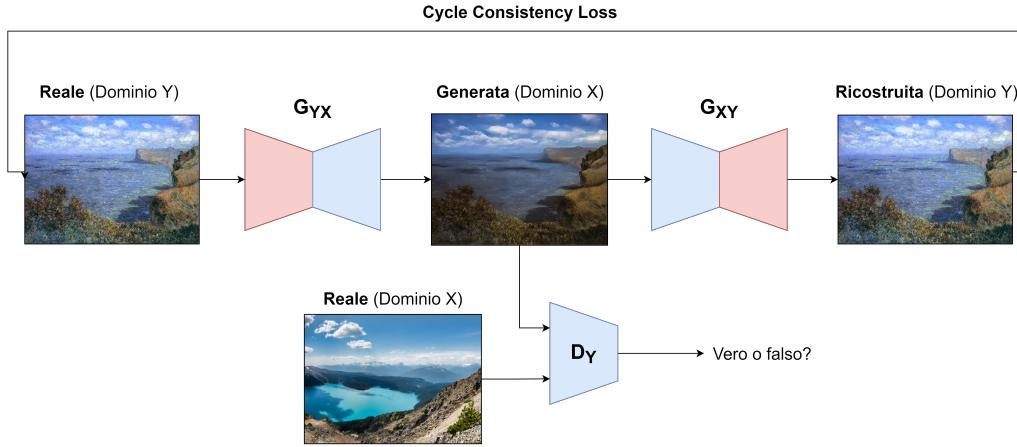


Figura 3.9. Ciclo backward della CycleGAN che trasforma foto in dipinti di Monet.

3.6.5 CycleGAN-VCs

Di seguito vengono descritte le architetture delle CycleGAN-VCs per poter vedere quali sono i passi che hanno portato ad ottenere la MaskCycleGAN-VC, ovvero la rete impiegata in questo lavoro.

CycleGAN-VC Nel 2017 Kaneko e Kameoka propongono un metodo per la voice conversion che permette di imparare a mappare una voce source in una target senza bisogno di dati paralleli. Il metodo, chiamato CycleGAN-VC, si basa sull'architettura di una CycleGAN a cui vengono applicate due modifiche: l'impiego di gated CNN e l'impiego di una identity-mapping loss (Fig. 3.10)[11].

L'introduzione di gated CNN all'interno della rete permette di conservare la struttura gerarchica della voce mantenendo un costo computazionale piuttosto basso mentre l'impiego di una funzione di identità (3.3) ha lo scopo di mantenere il contenuto linguistico intatto.

$$\mathcal{L}_{id}(G_{XY}, G_{YX}) = \mathbb{E}_{y \sim P_{Data}(y)}[||G_{XY}(y) - y||_1] + \mathbb{E}_{x \sim P_{Data}(x)}[||G_{YX}(x) - x||_1] \quad (3.3)$$

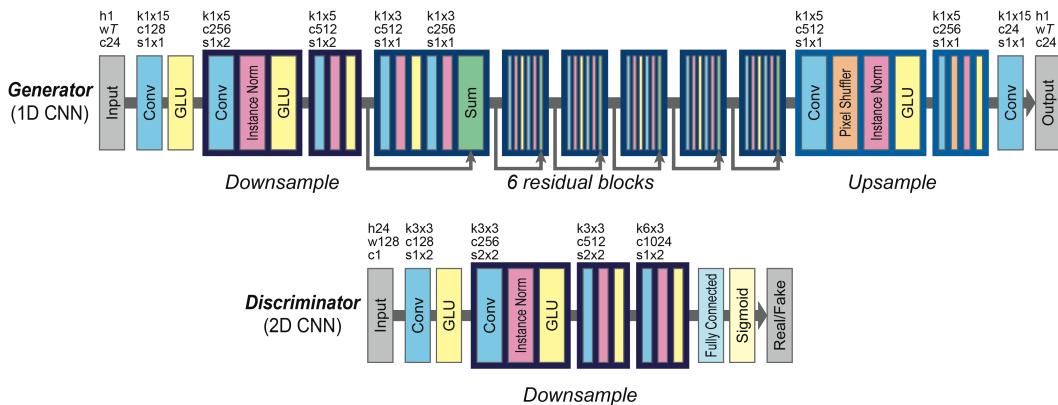


Figura 3.10. Architettura di CycleGAN-VC. Fonte [11].

Il modello utilizza coefficienti mel cepstrum, frequenza fondamentale logaritmica e aperiodicità come feature per la conversione. I risultati ottenuti sono di particolare interesse in quanto simili a procedure con impiego di dati paralleli senza necessitare di ulteriori dati o dell'allineamento temporale di questi.

CycleGAN-VC2 Successivamente, nel 2019 viene proposta una versione migliorata, chiamata CycleGAN-VC2 (Fig. 3.11), che incorpora tre principali cambiamenti: introduzione di una seconda adversarial loss (3.4), generatore migliorato (2-1-2D CNN) e discriminatore migliorato (PatchGAN)[12].

$$\mathcal{L}_{adv2}^{X \rightarrow Y \rightarrow X} = \mathbb{E}_{x \sim P_X} [\log D'_X(x)] + \mathbb{E}_{x \sim P_X} [\log(1 - D'_X(G_{YX}(G_{XY}(x))))] \quad (3.4)$$

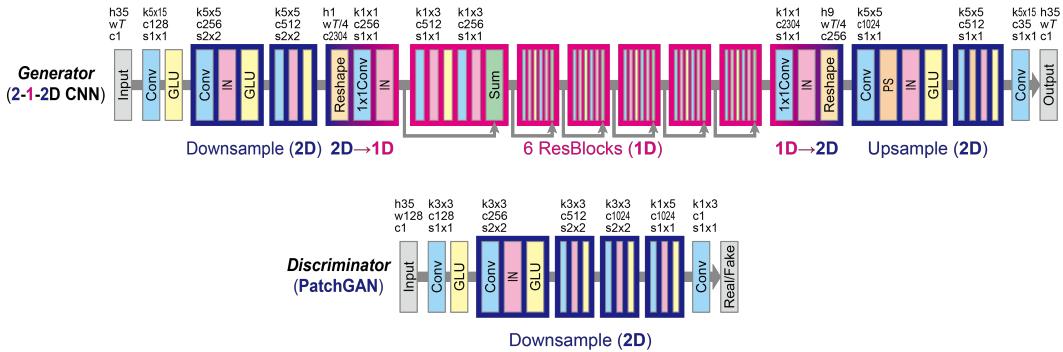


Figura 3.11. Architettura di CycleGAN-VC2. Fonte [12].

CycleGAN-VC3 Gli ottimi risultati ottenuti dalle CycleGAN-VCs hanno portato all'impiego di queste come metodi di benchmark per altri studi. Questo però ha evidenziato la necessità di confrontare i risultati sotto forma di spettrogrammi mel. Tuttavia impiegando spettrogrammi mel, al posto di coefficienti mel cepstrum, direttamente come input di queste reti, si producono risultati scarsi dovuti alla compromissione della struttura temporale del segnale.

Al fine di ovviare a questo problema viene proposta CycleGAN-VC3 (Fig. 3.12) che incorpora una normalizzazione adattiva sul dominio tempo-frequenza (TFAN)[13].

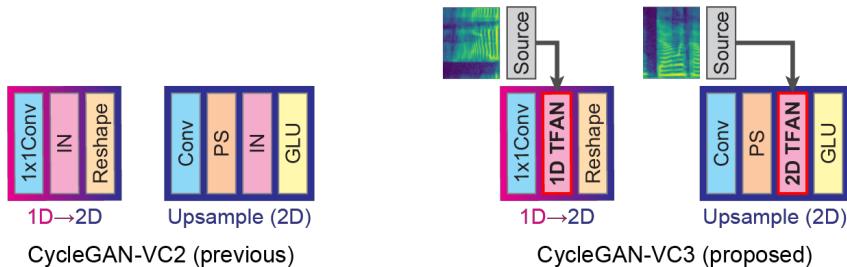


Figura 3.12. Comparazione di CycleGAN-VC3 rispetto a CycleGAN-VC2. In CycleGAN-VC3 viene incorporato TFAN nel generatore di CycleGAN-VC2. In particolare viene rimpiazzato nel blocco 1D→2D e nel blocco Upsample. Fonte [13].

MaskCycleGAN-VC La MaskCycleGAN-VC (Fig. 3.13) è una variante della CycleGAN-VC2 che, come la CycleGAN-VC3, impiega spettrogrammi mel al posto di coefficienti mel cepstrum [14].

Tra le modifiche principali rispetto alla precedente CycleGAN-VC3, questa architettura non impiega il modulo aggiuntivo TFAN, che comportava un incremento del numero di parametri da imparare, ma sfrutta invece una tecnica di *filling in frames* (FIF). Con tale tecnica vengono applicate maschere temporali durante la fase di addestramento, che hanno lo scopo di far apprendere al modello come riempire il frame mancante in base al contesto.

Si procede con la descrizione del funzionamento dell'architettura. Dato uno spettrogramma mel x , viene prima creata una maschera temporale m e viene moltiplicata ad esso. Si ottiene quindi un nuovo spettrogramma mel \hat{x} a cui è stato rimosso dei frame:

$$\hat{x} = x \cdot m \quad (3.5)$$

Si procede a passare al convertitore \hat{x} e la sua maschera m , ottenendo la sua conversione y' :

$$y' = G_{XY}^{mask}(concat(\hat{x}, m)) \quad (3.6)$$

Passando la maschera esplicitamente, si permette al convertitore di sapere dove andare a generare informazioni mancanti. Viene calcolata la adversarial loss per assicurarsi che y' sia nell'insieme target Y e viene inoltre convertita nuovamente nel dominio X . Si ricostruisce dunque x'' applicando una maschera fittizia m' che non rimuoverà alcun frame (una matrice di soli uno):

$$x'' = G_{YX}^{mask}(concat(y', m')) \quad (3.7)$$

Si applica quindi la cycle-consistency loss per lo spettrogramma mel ricostruito e viene calcolata la seconda adversarial loss (3.4).

$$\mathcal{L}_{mcyc}^{X \rightarrow Y \rightarrow X} = \mathbb{E}_{x \sim P_X, m \sim P_M} [| | | x'' - x | |_1] \quad (3.8)$$

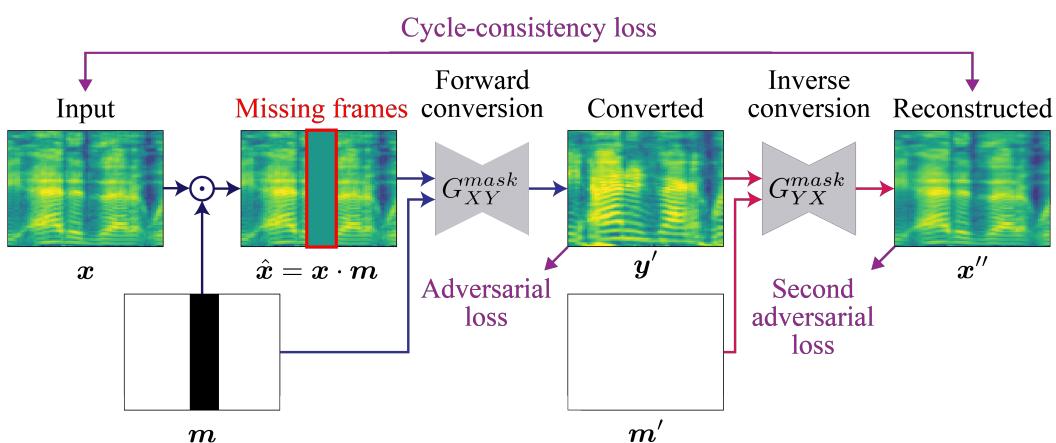


Figura 3.13. Ciclo forward dell'architettura MaskCycleGAN-VC. Fonte [14].

Capitolo 4

Architettura proposta

In questo capitolo sarà descritta l'architettura del framework sviluppato per la conversione di voci a spettro ridotto mediante l'utilizzo di vocoded speech e di sine-wave speech.

4.1 Metodo

L'idea alla base di questo framework è la conversione di voci tra due speaker differenti che andremo a chiamare X (sorgente) e Y (target).

L'innovazione principale del framework proposto è la riduzione dello spettro sonoro, processo attraverso il quale si riducono le caratteristiche acustiche dello speaker sorgente X .

Andremo a presentare due moduli per questo scopo: un modulo di riduzione a sine-wave speech (SWS) e uno di riduzione a vocoded speech.

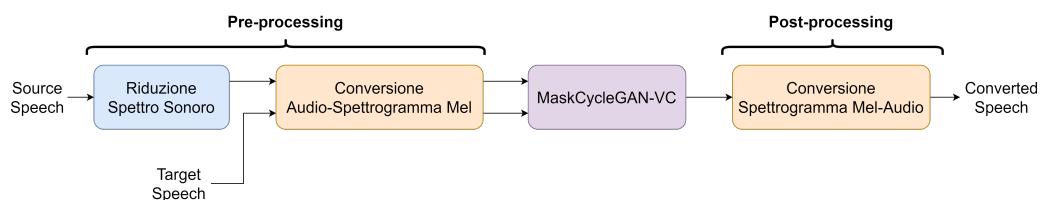


Figura 4.1. Pipeline dell'architettura proposta.

4.2 Modulo di riduzione dello spettro sonoro

I seguenti moduli sono parte della fase di pre-processamento dell'audio (Fig. 4.1 e 4.2).

4.2.1 Modulo di riduzione a vocoded speech

Il metodo implementato per generare il vocoded speech si basa su Linear Predictive Coding (LPC), impiegando l'algoritmo di Burg[9]. Si descrive di seguito la procedura applicata:

- L'audio originale viene ricampionato a 8 kHz. In questo modo la frequenza di Nyquist, che per definizione si attesta a $\frac{1}{2}f_s$, sarà di 4 kHz ovvero sarà possibile rappresentare senza distorsioni suoni fino a tale frequenza. Tale range di frequenze viene usato come banda standard per la voce in comunicazioni telefoniche.
- Vengono estratte finestre di 200 sample dell'audio.
- Per ogni finestra vengono calcolati i coefficienti del filtro lineare di ordine 8.
- Viene generato un segnale carrier della durata di 200 sample (es. buzz a 500 Hz oppure noise).
- Viene applicato un filtro lineare del resiudo LPC al carrier in modo da ricostruire il segnale vocale su di esso.
- Viene calcolata la magnitudine della finestra e viene adattato il segnale generato ad essa.
- L'audio viene ricostruito e ricampionato alla frequenza originale.

4.2.2 Modulo di riduzione a SWS

Al fine di trasformare degli audio in forma sine-wave speech abbiamo la necessità di trovare le formanti e rimpiazzarle con delle onde sinusoidali pure. Il metodo implementato si basa sulla stima delle posizioni delle formanti data dalla Linear Predictive Coding (LPC)[24], utilizzando l'algoritmo di Burg[9]. Si descrive di seguito la procedura applicata:

- L'audio originale viene ricampionato a 8 kHz.
- Vengono estratte finestre di 200 sample dell'audio.
- Per ogni finestra vengono calcolati i coefficienti del filtro lineare di ordine 12 e si ottengono le frequenze delle formanti. La scelta dell'ordine deriva dalla seguente formula $o = 2 \cdot n_f + 2$, dove n_f rappresenta il numero delle formanti da trovare.
- Viene calcolata la magnitudine della finestra.
- Vengono interpolate le formanti al fine di creare 3 segnali sinusoidali mobili.
- Si ricostruisce l'audio alla frequenza di campionamento originale con sinusoidi alle frequenze delle formanti.

4.3 Modulo di conversione audio-spettrogrammi

Al fine di poter utilizzare la rete MaskCycleGAN-VC è necessario trasformare degli audio in spettrogrammi da fornire come input ad essa e successivamente di invertire questa trasformazione per ottenere un audio in output. È stata scelta di impiegare

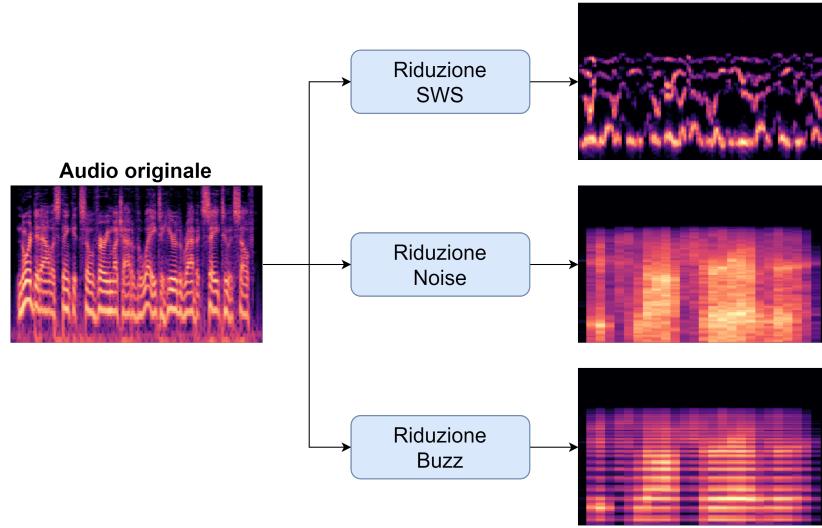


Figura 4.2. Risultati della riduzione dello spettro sonoro applicando i tre metodi proposti: sine-wave speech, noise vocoded speech e buzz vocoded speech.

lo stesso modello utilizzato nel paper di riferimento di Kaneko et al.[14], ovvero una MelGAN[16] pretrainata, al fine di poter effettuare un confronto più diretto sui risultati ottenuti dalle rappresentazioni scelte come input.

4.4 Architettura della rete

L’architettura della rete neurale artificiale utilizzata è la MaskCycleGAN-VC come descritta nel Capitolo 3 (Fig. 4.3). La scelta di non modificare parametri della rete è voluta al fine di poter ottenere dei risultati oggettivi dipendenti esclusivamente dalla riduzione spettrale proposta.

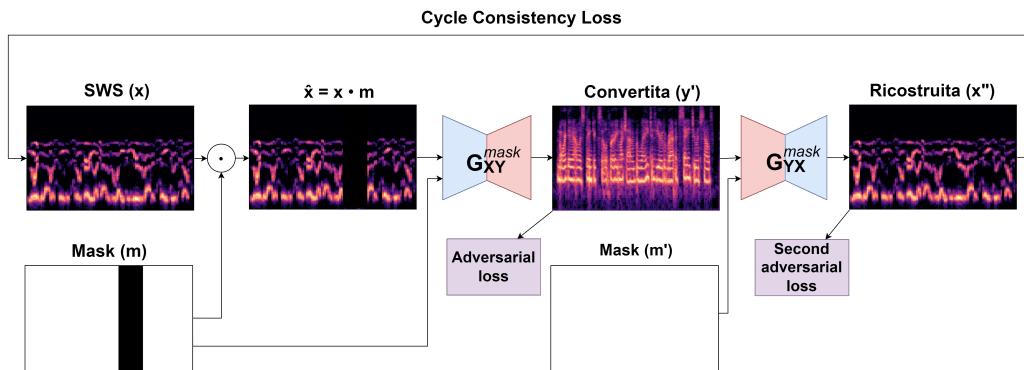


Figura 4.3. Ciclo forward dell’architettura proposta impiegando il modulo di riduzione a SWS. Il modello verrà addestrato a trasformare la forma ridotta della voce di X nella voce di Y .

Capitolo 5

Test e valutazioni

In questo capitolo verranno descritti i test effettuati, i criteri utilizzati per la valutazione e verranno comparati le tre riduzioni spettrali proposte: SWS, noise vocoded speech e buzz vocoded speech.

5.1 Dataset

È stato utilizzato il dataset fornito dalla VCC2018 (Voice Conversion Challenge 2018) in quanto riferimento principale per le performance di voice conversion.

Il dataset è formato da 8 source speaker e 4 target speaker, a ciascuno di essi corrispondono 81 audio di frasi lette (in totale circa 5 minuti di durata) con frequenza di campionamento a 22.05 kHz. Al fine di lavorare su dati non paralleli, è necessario effettuare le conversioni proposte dal task "Spoke" della VCC2018.

Sono stati usati un sottoinsieme di speaker in modo da testare conversioni tra generi differenti e tra lo stesso genere, elencate a seguire:

- VCC2SF3 → VCC2TF1 (F→F)
- VCC2SF3 → VCC2TM1 (F→M)
- VCC2SM3 → VCC2TF1 (M→F)
- VCC2SM3 → VCC2TM1 (M→M)

I nomi corrispondono a quelli assegnati internamente del dataset e descrivono il genere (M, F) e l'appartenenza all'insieme di source (S) o target (T).

5.2 Training

Il dataset è stato processato come descritto nel Capitolo 4. In particolare sono stati predisposti quattro test separati per valutare le seguenti rappresentazioni: nessuna riduzione spettrale, vocoder con noise carrier, vocoder con buzz carrier (500 Hz) e sine-wave speech con 5 formanti.

Le reti sono state trainate per 150k iterazioni seguendo le specifiche della MaskCycleGAN-VC ovvero usando un ottimizzatore Adam, con learning rate impostato a 0.0002 per il generatore e 0.0001 per il discriminatore e momentum β_1 e β_2

rispettivamente 0.5 e 0.999. La batch size è stata impostata a 1, dove ogni sample consiste in 64 frame tagliati casualmente. La maschera applicata sull'input è stata impostata a 25.

5.3 Metodi di valutazione

Esistono varie metriche per la valutazione oggettiva della voice conversion, in questo lavoro si valuteranno come per la MaskCycleGAN-VC le seguenti: mel-cepstral distortion (MCD)[15] e Kernel DeepSpeech Distance (KDSD)[2].

La MCD misura la variazione di spettro, di conseguenza non è necessariamente correlata con la naturalezza del suono, mentre la KDSD misura la distanza tra le feature degli audio reali e quelli generati e ha dimostrato una correlazione con la valutazione effettuata da persone. Data la natura di queste misurazioni, una conversione è considerabile migliore quando queste metriche hanno risultato più basso.

Verrà inoltre impiegata una MOSNet[19] pre-trainata per ottenere una stima di riferimento per quanto riguarda la valutazione soggettiva. Esso è un modello che, basandosi sulle opinioni reali fornite da ascoltatori alle submission effettuate alla VCC2018, è stato addestrato a predire valutazioni soggettive MOS (mean opinion score) e ha dimostrato una forte correlazione con i voti effettivi delle persone. La valutazione è espressa, come per la MOS, in una valutazione nell'intervallo [1,5] dove 1 significa una bassa qualità e 5 un'ottima qualità.

5.4 Risultati

Si riportano a seguire i risultati ottenuti dalle conversioni di voci. Vengono riportate le valutazioni basate sulle scale MCD, KDSD e MOSNet delle quattro conversioni effettuate per ciascuna tipologia di dati usata come input della rete.

Per ogni tipologia di conversione (es. F→M) sono stati messi in evidenza i risultati migliori per ciascuna metrica. Risulta interessante notare come il metodo originale [14] senza riduzioni di spettro, ottenga ottime risultati per la metrica di KDSD, mentre le conversioni effettuate con le riduzioni di spettro ottengono punteggi migliori per le metriche di MCD e MOSNet.

Riduzione spettro ^a	Valutazione	F→F	F→M	M→F	M→M
Nessuna riduzione ^b	MCD (dB)	6.61	6.57	6.98	6.89
	KDSD	2074	1755	2770	1583
	MOSNet	3.84	4.46	3.92	4.58
Noise vocoded ^c	MCD [dB]	6.53	6.47	6.75	6.73
	KDSD [$\times 10^5$]	3269	2247	3446	2032
	MOSNet	3.90	4.46	3.89	4.49
Buzz vocoded ^c	MCD [dB]	6.49	6.49	6.70	6.71
	KDSD [$\times 10^5$]	3063	2155	3169	1823
	MOSNet	3.80	4.47	3.94	4.53
Sine-wave speech ^c	MCD [dB]	6.55	6.55	6.98	6.78
	KDSD [$\times 10^5$]	3513	2621	4802	2492
	MOSNet	3.91	4.48	3.86	4.61

^a Modulo di riduzione dello spettro applicato sui dati di input.

^b Nessuna riduzione spettrale applicata, modello trainato come proposta da Kaneko et al. in [14].

^c Metodi di riduzione spettrale proposti, come descritti nella sezione 4.2.

Capitolo 6

Conclusioni

In questo lavoro si sono combinate tecniche di speech processing più tradizionali con le odierne di deep learning al fine di ricercare una forma a spettro ridotto della voce che permetta di ridurre la componente acustica, preservando quella linguistica.

Dalle valutazioni si evince che questo approccio è possibile e che i suoi risultati sono equiparabili a quelli ottenuti usando audio senza riduzioni di spettro. Come sviluppi futuri si ritiene interessante approfondirne l'applicazione in altri campi, come ad esempio nella speech recognition al fine di addestrare modelli anonimizzati, e sviluppare metodi per sfruttare al meglio questa forma facilmente manipolabile per data augmentation.

Bibliografia

- [1] Abien Fred Agarap. Deep Learning using Rectified Linear Units (ReLU). *arXiv e-prints*, March 2018.
- [2] Mikołaj Bińkowski, Jeff Donahue, Sander Dieleman, Aidan Clark, Erich Elsen, Norman Casagrande, Luis C. Cobo, and Karen Simonyan. High Fidelity Speech Synthesis with Adversarial Networks. 2019.
- [3] Ju-chieh Chou, Cheng-chieh Yeh, and Hung-yi Lee. One-shot Voice Conversion by Separating Speaker and Content Representations with Instance Normalization. 2019.
- [4] L. Cohen. Time-frequency distributions-a review. *Proceedings of the IEEE*, 77(7):941–981, 1989.
- [5] Matthew H. Davis, Ingrid S. Johnsrude, Alexis Hervais-Adelman, Karen J. Taylor, and Carolyn McGettigan. Lexical information drives perceptual learning of distorted speech: evidence from the comprehension of noise-vocoded sentences. *Journal of experimental psychology. General*, 134 2:222–41, 2005.
- [6] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.
- [7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. June 2014.
- [9] A. Gray and D. Wong. The Burg algorithm for LPC speech analysis/Synthesis. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(6):609–615, 1980.
- [10] Muhammad Mohsin Kabir, M. F. Mridha, Jungpil Shin, Israt Jahan, and Abu Quwsar Ohi. A Survey of Speaker Recognition: Fundamental Theories, Recognition Methods and Opportunities. *IEEE Access*, 9:79236–79263, 2021.
- [11] Takuhiro Kaneko and H. Kameoka. Parallel-Data-Free Voice Conversion Using Cycle-Consistent Adversarial Networks. *ArXiv*, abs/1711.11293, 2017.

- [12] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. Cyclegan-VC2: Improved Cyclegan-based Non-parallel Voice Conversion. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6820–6824, 2019.
- [13] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. CycleGAN-VC3: Examining and Improving CycleGAN-VCs for Mel-spectrogram Conversion. 2020.
- [14] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. Maskcyclegan-VC: Learning Non-Parallel Voice Conversion with Filling in Frames. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5919–5923, 2021.
- [15] R. Kubichek. Mel-cepstral distance measure for objective speech quality assessment. In *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, volume 1, pages 125–128 vol.1, 1993.
- [16] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brebisson, Yoshua Bengio, and Aaron Courville. MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis. 2019.
- [17] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. 2015.
- [18] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [19] Chen-Chou Lo, Szu-Wei Fu, Wen-Chin Huang, Xin Wang, Junichi Yamagishi, Yu Tsao, and Hsin-Min Wang. MOSNet: Deep Learning-Based Objective Assessment for Voice Conversion. In *Interspeech 2019*. ISCA, sep 2019.
- [20] Robert E. Remez, Philip E. Rubin, David B. Pisoni, and Thomas D. Carrell. Speech perception without traditional speech cues. *Science*, 212(4497):947–950, 1981.
- [21] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.
- [22] P.E. Rubin. Sinewave synthesis. Internal memorandum, Haskins Laboratories, New Haven, CT, 1980.
- [23] Berrak Sisman, Junichi Yamagishi, Simon King, and Haizhou Li. An Overview of Voice Conversion and Its Challenges: From Statistical Modeling to Deep Learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:132–157, 2021.
- [24] R. C. Snell and F. Milinazzo. Formant location from LPC analysis data. *IEEE Transactions on Speech and Audio Processing*, 1(2):129–134, 1993.

- [25] S. S. Stevens, J. Volkmann, and E. B. Newman. A Scale for the Measurement of the Psychological Magnitude Pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, 1937.
- [26] Lifa Sun, Kun Li, Hao Wang, Shiyin Kang, and Helen Meng. Phonetic posterograms for many-to-one voice conversion without parallel data training. In *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2016.
- [27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. 2017.
- [28] P. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967.
- [29] Da-Yi Wu and Hung-yi Lee. One-Shot Voice Conversion by Vector Quantization. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7734–7738, 2020.
- [30] Mingyang Zhang, Yi Zhou, Li Zhao, and Haizhou Li. Transfer Learning from Speech Synthesis to Voice Conversion with Non-Parallel Training Data. 2020.
- [31] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.