Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Σημάτων, Ελέγχου και Ρομποτικής
Εργαστήριο Όρασης Υπολογιστών, Επικοινωνίας Λόγου και Επεξεργασίας Σημάτων

Music Source Separation on Classical Guitar Duets

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Μάριου Γλυτσού

Music Source Separation on Classical Guitar Duets

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Μάριου Γλυτσού

Επιβλέπων: Πέτρος Μαραγκός
Καθηγητής Ε.Μ.Π.

Συν-επιβλέπουσα: Δρ. Αθανασία Ζλατίντση
Μεταδιδακτορική Ερευνήτρια Ε.Μ.Π.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Σημάτων, Ελέγχου και Ρομποτικής
Εργαστήριο Όρασης Υπολογιστών, Επικοινωνίας Λόγου και Επεξεργασίας Σημάτων

Music Source Separation on Classical Guitar Duets

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Μάριου Γλυτσού

Επιβλέπων: Πέτρος Μαραγκός
Καθηγητής Ε.Μ.Π.

Συν-επιβλέπουσα: Δρ. Αθανασία Ζλατίντση
Μεταδιδακτορική Ερευνήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 28η Μαρτίου, 2024.

.....	.....	.....
Πέτρος Μαραγκός	Αλέξανδρος Ποταμιάνος	Αθανάσιος Ροντογιάννης
Καθηγητής Ε.Μ.Π.	Αναπληρωτής Καθηγητής Ε.Μ.Π.	Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2024

Μαριος Γλυτσος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός
και Μηχανικός Υπολογιστών Ε.Μ.Π.

Στον Λουκά

Περίληψη

Η παρούσα διπλωματική εργασία μελετά το πρόβλημα του διαχωρισμού μουσικών πηγών σε ντουέτα κλασικής κιθάρας, το οποίο εντάσσεται σε μια υποκατηγορία του γενικότερου προβλήματος διαχωρισμού μουσικών πηγών. Στα πλαίσια της έρευνας αυτής δημιουργήσαμε δύο νέα σύνολα δεδομένων που αποτελούνται από πραγματικές και συνθετικές ηχογραφήσεις ντουέτων κλασικής κιθάρας, με στόχο να διερευνήσουμε και να αξιολογήσουμε τις τεχνικές διαχωρισμού μουσικών πηγών που εφαρμόζονται μέχρι σήμερα στο δικό μας πρόβλημα. Αφού κάναμε τα αναλυτικά πειράματα, διαπιστώσαμε πως οι μετρικές που χρησιμοποιούνται για το ευρύτερο πρόβλημα του διαχωρισμού μουσικών πηγών δεν είναι αντιπροσωπευτικές σε περιπτώσεις διαχωρισμού ίδιων οργάνων. Στη συνέχεια, προτείνουμε μια καινούργια αρχιτεκτονική, η οποία συνδυάζει δυο διαφορετικά μοντέλα μεταγραφής μουσικής και διαχωρισμού μουσικών πηγών. Στην πράξη, ένας ανθρώπινος ακροατής ακούγοντας ένα ντουέτο κλασικής κιθάρας θα αντιλαμβανόταν φυσικά το μέρος της κάθε κιθάρας, και εκμεταλλευόμενος αυτή τη πληροφορία θα διακρίνει τον ήχο της κάθε κιθάρας στο μυαλό του. Η αρχιτεκτονική που προτείνουμε στοχεύει να μιμηθεί αυτή την ανθρώπινη προσέγγιση ενσωματώνοντας μεταγραφή και διαχωρισμό του μουσικού σήματος. Επιπλέον, αυτή η έρευνα υποστηρίζει ότι οι μεθοδολογίες και οι διορατικότητες που αποκτήθηκαν από τη μελέτη του διαχωρισμού ντουέτων κλασικής κιθάρας θα μπορούσαν πιθανώς να εφαρμοστούν σε άλλους συναφείς τομείς όπως ο διαχωρισμός ομιλητών και ο διαχωρισμός φωνής τραγουδιού.

Λέξεις Κλειδιά — Ανάκτηση Μουσικής Πληροφορίας, Μουσική και Τεχνητή Νοημοσύνη, Διαχωρισμός Πηγών Μουσικής, Διαχωρισμός Πηγών Μουσικής Κοινής Χροιάς, Μεταγραφή Μουσικής.

Abstract

The dissertation presents an in-depth study on the separation of sources in classical guitar duets, addressing the unique challenge posed by the similar timbral characteristics of the instruments involved. This research introduces two new datasets comprised of real and synthetic recordings of guitar duets, designed to facilitate the exploration and evaluation of source separation techniques. Furthermore we propose a novel augmentation technique, OppositePanning, to enhance the separation process by exploiting the spatial distribution of sound, thereby offering a new avenue for improving source separation in settings where instruments share similar timbral characteristics.

We propose a model pipeline which is is motivated by the understanding that guitar duet separation is inherently a hybrid task for humans. In practice, a human listener would naturally perceive the symbolic score from the audio, leveraging this score to aid in the separation process. This insight forms the foundation for the proposed dual-model pipeline, which aims to mimic this human approach by incorporating symbolic musical information directly into the separation algorithm. This approach is a significant departure from traditional source separation techniques, which primarily focus on the acoustic signal without considering the underlying musical structure.

By employing a comparative analysis of Signal-to-Distortion Ratio (SDR) metrics, we evaluate the performance of the proposed dual-model pipeline against traditional methods. The findings demonstrate that incorporating symbolic musical information significantly improves separation accuracy, highlighting the importance of considering the musical context in source separation tasks.

Moreover, this research posits that the methodologies and insights gained from the study of classical guitar duet separation could potentially be applied in other related fields as of speaker separation and voice singing separation, offering new perspectives and techniques for achieving more robust separation in complex auditory environments. The exploration of OppositePanning and the dual-model pipeline not only advances the understanding and methodology of monotimbral music source separation but also opens up avenues for further research in polyphonic music analysis and beyond, potentially leading to significant improvements in various applications of audio separation technology.

Keywords — Music Information Retrieval, Music and AI, Music Source Separation, Monotimbral Music Source Separation, Guitar Duets, Music Transcription,

Ευχαριστίες

Αρχικά με την ευκαιρία της παρούσης διπλωματικής εργασίας μου, θα ήθελα να ευχαριστήσω θερμά τον καθηγητή κ. Πέτρο Μαραγκό, για την εμπιστοσύνη που μου έδειξε και για την ευκαιρία που μου έδωσε προκειμένου να εκπονήσω την εν λόγω εργασία στο εργαστήριό του. Αξίζει να αναφέρω ότι μέσα από τη διδασκαλία των προπτυχιακών του μαθημάτων στο Εθνικό Μετσόβιο Πολυτεχνείο με ενέπνευσε βαθύτατα να ασχοληθώ και να εμβαθύνω στην τομή της Μηχανικής Μάθησης και της μουσικής.

Στην συνέχεια, θα ήθελα να ευχαριστήσω τη Νάνσυ Ζλατίντση και τον Χρήστο Γαρούφη οι οποίοι συνεπέβλεψαν την εργασία μου. Η συνεισφορά τους δεν περιορίστηκε μόνο στα πλαίσια της εργασίας αλλά και στην καθοδηγήση και τον προσανατολισμό σε πολλά άλλα επίπεδα.

Θέλω ακόμα να ευχαριστήσω τον μουσικό και φίλο Ορφέα Ζίνδρο ο οποίος συμμετείχε στην ηχογράφηση με τις κλασικές κιθάρες. Τέλος θέλω να ευχαριστήσω όλους τους φίλους μου, Αναστασία, Γιώργο, Ειρήνη, Ευτυχία, Ηλία, Νίκο, Περικλή και Φαίδρα, που στάθηκαν δίπλα μου κατά τη διάρκεια των σπουδών μου ο καθένας με τον δικό του τρόπο.

Γλυτσός Μάριος
Μάρτιος 2024

Contents

List of Figures

List of Tables

Εκτεταμένη Περίληψη στα Ελληνικά

Περιγραφή Προβλήματος: Music Source Separation

Ο διαχωρισμός πηγών μουσικής (Music Source Separation) είναι μια διαδικασία η οποία στοχεύει στο να αποσυνδέσει τις ηχητικές πηγές οι οποίες συνιστούν μια ηχητική εγγραφή χωρίς να έχει κάποια περαιτέρω πληροφορία των ιδιοτήτων των συμμετεχόντων σημάτων. Η εν λόγω διαδικασία παρουσιάζει αρκετές ομοιότητες με την αποθορυβοποίηση σημάτων, καθώς και οι δύο διαδικασίες περιλαμβάνουν τον διαχωρισμό σημάτων ενδιαφέροντος από ανεπιθύμητα σήματα, ο διαχωρισμός πηγής ωστόσο επικεντρώνεται κυρίως στην απομόνωση των "κατάλληλων" σημάτων και όχι στην εξάλειψη του θορύβου. Ένα παράδειγμα διαχωρισμού φαίνεται στο παρακάτω Σχήμα 0.0.1

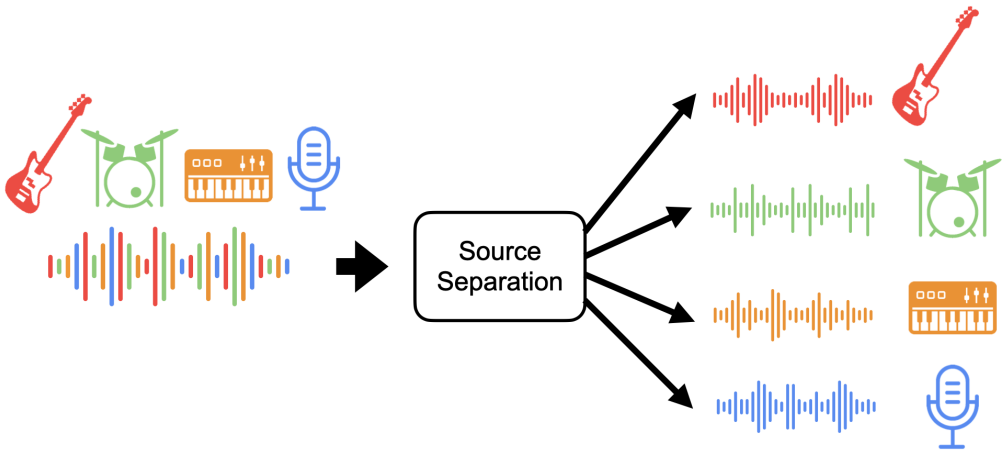


Figure 0.0.1: Διαχωρισμός μουσικών σημάτων (Ντραμς, Φωνή, Μπάσσο, Υπόλοιπα Όργανα) [1]

Διαχωρισμός Μουσικών Πηγών σε Ντουέτα Κλασικής Κιθάρας

Στα πλαίσια της συγκεκριμένης διπλωματικής εργασίας θα ασχοληθούμε με το πρόβλημα του διαχωρισμού μουσικών πηγών στο όργανο της κλασικής κιθάρας και συγκεκριμένα σε ντουέτα κλασικής κιθάρας. Το εν λόγω πρόβλημα είναι μια υποκατηγορία του Διαχωρισμού Πηγών Μουσικής (Music Source Separation) η οποία ονομάζεται Διαχωρισμός Μουσικών Πηγών ίδιας χροιάς (Monotimbral Source Separation). Στόχος του προβλήματος αυτού είναι να καταφέρει να αποσυνθέσει το σήμα της κάθε κιθάρας απο το αρχικό συνολικό σήμα. Στο παρακάτω Σχήμα 0.0.2 φαίνεται η διαδικασία του Διαχωρισμού Μουσικών Πηγών σε ένα ντουέτο κλασικής κιθάρας 0.0.2.

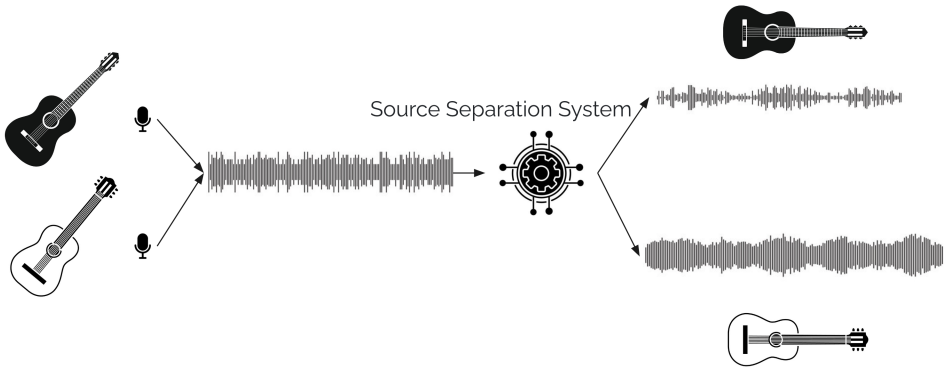


Figure 0.0.2: Διαχωρισμός Μουσικών Πηγών σε ντουέτο κλασικής κιθάρας.

Προκλήσεις

Μέχρι στιγμής, υπάρχουν αρκετές έρευνες οι οποίες έχουν πετύχει πολύ ικανοποιητικά αποτελέσματα στο πρόβλημα διαχωρισμού πηγών, στις περιπτώσεις όπου τα σήματα παρουσιάζουν διαφορετικές χροιές μεταξύ τους. Ο διαχωρισμός που γίνεται απο τους αλγορίθμους, πλέον χρησιμοποιείται για να παράξει σήματα υψηλής ποιότητας ικανά να χρησιμοποιηθούν σε νέες ηχογραφήσεις απο καταξιωμένους μουσικούς και παραγωγούς. Η έρευνα ωστόσο που έχει γίνει στο πρόβλημα του Διαχωρισμού Πηγών που παρουσιάζουν κοινές χροιές δεν είναι εκτενής, καθώς το συγκεκριμένο πρόβλημα παρουσιάζει αρκετά περισσότες προκλήσεις σε σχέση με την γενικότερη κατηγορία του διαχωρισμού. Στον κλασικό διαχωρισμό πηγής, η εστίαση είναι στον διαχωρισμό μεμονωμένων πηγών με διακριτά ηχοχρωματικά χαρακτηριστικά, όπως τα φωνητικά, τα τύμπανα, το μπάσο και άλλα. Αυτές οι πηγές παρουσιάζουν συχνά σημαντικές φασματικές και χρονικές διαφορές, γεγονός που καθιστά ευκολότερη τη διάκρισή τους. Αντίθετα, ο διαχωρισμός πηγών με κοινή χροιά στοχεύει στο διαχωρισμό πηγών που ανήκουν στην ίδια οικογένεια οργάνων ή μοιράζονται παρόμοια ηχοχρωματικά χαρακτηριστικά. Στην περίπτωση των ντουέτων κλασικής κιθάρας, και οι δύο κιθάρες παράγουν ήχους με παρόμοια ηχοχρώματα, καθιστώντας δύσκολο τον διαχωρισμό των επιμέρους μερών αποκλειστικά βάσει φασματικών διαφορών. Ακόμα, σε αντίθεση με τα μονοφωνικά όργανα, όπως μια μεμονωμένη φωνή ή μια σόλο κιθάρα, η αλληλεπίδραση μεταξύ δύο κιθάρων σε ένα ντουέτο δημιουργεί πολύπλοκα αρμονικά περιεχόμενα, τα οποία έχουν ως αποτέλεσμα την αλληλεπικάλυψη συνιστωσών συχνότητας, καθιστώντας ακόμα πιο δύσκολη την απομόνωση μεμονωμένων τμημάτων κιθάρας.

Θεωρητικό Υπόβαθρο

Κρίνεται σκόπιμο πρωτού αναλύσουμε τά διάφορα δομικά στοιχεία τα οποία συνιστούν τους αλγορίθμους οι οποίοι θα αναλυθούν παρακάτω να καλύψουμε τον τρόπο με τον οποίο αναπαριστούμε τα ηχητικά σήματα στον ηλεκτρονικό υπολογιστή και τη μορφή με την οποία οι αλγόριθμοι τα επεξεργάζονται.

Αναπαραστάσεις Ηχητικών Σημάτων

Ο τρόπος με τον οποίο αναπαρηστούμε ψηφιακά τον ήχο παίζει καθοριστικό ρόλο στην διαδικασία του διαχωρισμού πηγών μουσικής και γενικότερα στην επεξεργασία του σήματος. Διαφορετικές αναπαραστάσεις μας επιτρέπουν να απεικονίσουμε και να εξετάσουμε διαφορετικά χαρακτηριστικά των ηχητικών σημάτων. Οι σημαντικότερες αναπαραστάσεις του ήχου είναι η **χρονική δειγματοληψία του σήματος** και τα **φασματοραφήματα** που παρουσιάζουν την πληροφορία για το συχνοτικό φάσμα του σήματος στο πέρασμα του χρόνου.

Αναλύοντας περαιτέρω την κάθε αναπαράσταση, η *δειγματολειπτημένη κυματομορφή* ενός ηχητικού σήματος $s(t)$ είναι μια συνάρτηση του χρόνου η οποία δείχνει τις μεταβολές του πλάτους με την πάροδο του χρόνου t. Για την ακριβή ψηφιοποίηση ενός ηχητικού σήματος, η υψηλότερη συχνότητα στον αρχικό ήχο δεν πρέπει να είναι μεγαλύτερη από το μισό του ρυθμού δειγματοληψίας. Ο παραπάνω κανόνας είναι γνωστός ως θεώρημα Nyquist-Shannon. Μπορούμε να δούμε ένα δειγματοληπτημένο σήμα ήχου στο παρακάτω σχήμα 0.0.3.

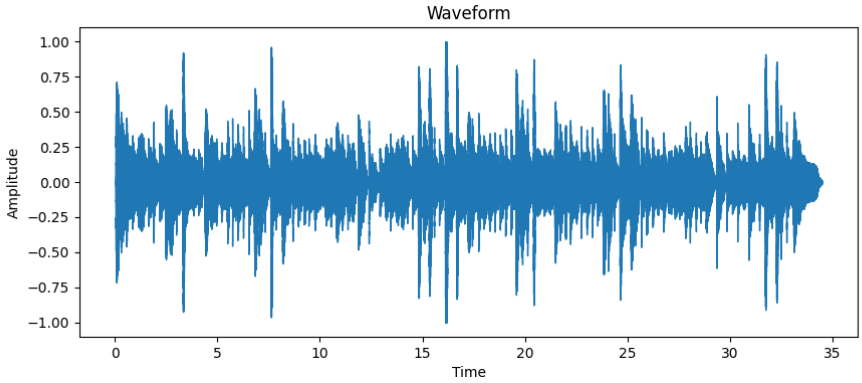


Figure 0.0.3: Γραφική αναπαράσταση σήματος ήχου.

Ένα συνεχές σήμα, όπως για παράδειγμα η φωνή, παίρνει τιμές πλάτους σε ένα συνεχές εύρος. Ανθρώπινες αισθήσεις, όπως το αυτί, μπορούν να αντιληφθούν πεπερασμένες διαφορές έντασης. Έτσι, μπορούμε να προσεγγίσουμε το αρχικό σήμα χρησιμοποιώντας ένα σήμα που αποτελείται από διακριτές τιμές πλάτους, επιλεγμένες από ένα πεπερασμένο σύνολο. Η διαδικασία μετατροπής αναλογικού δείγματος σε ψηφιακή μορφή ονομάζεται κβαντοποίηση. Γραφικά, αυτό σημαίνει ότι μια γραμμική σχέση μεταξύ εισόδου και εξόδου αντικαθίσταται από μια κλιμακωτή χαρακτηριστική σχέση. Η διαφορά μεταξύ γειτονικών διακριτών τιμών ονομάζεται κβάντο ή μέγεθος βήματος. Η διαδικασία της κβαντοποίησης φαίνεται στο παρακάτω Σχήμα 0.0.4.

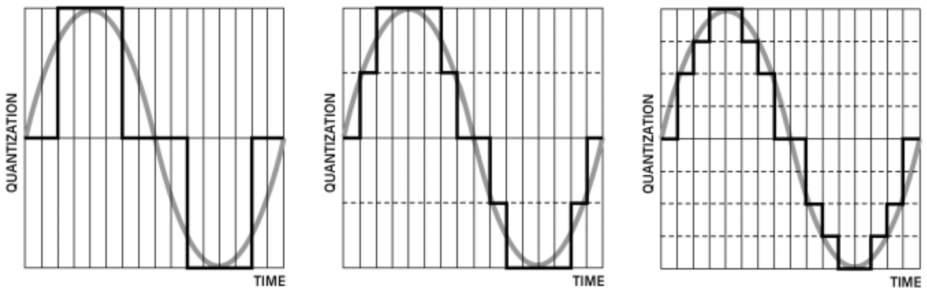


Figure 0.0.4: Κβαντοποίηση ψηφιακού σήματος. Απο [2].

Απο την άλλη πλευρά οι *φασματικές αναπαραστάσεις* λαμβάνονται μέσω διαφόρων τεχνικών μετασχηματισμού Fourier, και παρέχουν πληροφορίες σχετικά με πώς τα χαρακτηριστικά του ήχου μεταβάλλονται ως συνάρτηση του χρόνου. Ως υπενθύμιση, ο μετασχηματισμός Fourier ενός σήματος ορίζεται ως:

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft}\, dt \tag{0.0.1}$$

Όπου:

- $S(f)$ είναι ο μετασχηματισμός Fourier του σήματος $s(t)$
- $s(t)$ είναι το αρχικό σήμα

Μια απο τις σημαντικότερες φασματικές αναπαραστάσεις η οποία χρησιμοποιείται στη παρούσα διπλωματική εργασία είναι το φασματογράφημα (Spectrogram), το οποίο απεικονίζει τις αλλαγές στο φασματικό περιεχόμενο ενός σήματος ως συνάρτηση του χρόνου. Προκύπτει μέσω του Μετασχηματισμού Fourier Βραχείας Διάρκειας (STFT), ο οποίος θεωρείται ως η εφαρμογή του Διακριτού Μετασχηματισμού Fourier (DFT) σε συνεχόμενα, επικαλυπτόμενα τμήματα του σήματος. Αυτή η διαδικασία μπορεί να αναπαρασταθεί μαθηματικά ως εξής:

$$STFT\{x[n]\}(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j2\pi\frac{\omega}{N}n} \tag{0.0.2}$$

Όπου:

- $x[n]$ είναι η διακριτή χρονική αναπαράσταση του αρχικά συνεχούς χρονικού σήματος $x(t)$, όπου n αντιπροσωπεύει τους διακριτούς χρονικούς δείκτες. Η μετάβαση από $x(t)$ σε $x[n]$ σηματοδοτεί τη δειγματοληψία του συνεχούς σήματος σε μορφή κατάλληλη για ψηφιακή επεξεργασία.
- $w[n-m]$ είναι η συνάρτηση παραθύρου που εφαρμόζεται γύρω από τον δείκτη m.
- m σημαίνει τον διακριτό χρονικό δείκτη γύρω από τον οποίο εστιάζεται η συνάρτηση παραθύρου.
- ω αντιπροσωπεύει τη συχνότητα για την οποία υπολογίζεται ο STFT.

και τελικά

$$\text{Φασματογράφημα}(m,\omega) = |STFT\{x[n]\}(m,\omega)|^2 \tag{0.0.3}$$

όπου $S(t,f)$ είναι ο Short-Time Fourier Transform (STFT) του σήματος $s(t)$ και αντιπροσωπεύει το φάσμα ισχύος του αναλυόμενου τμήματος, απεικονίζοντας την ένταση διαφόρων συχνοτήτων κατά τη διάρκεια κάθε χρονικού σημείου m. Στο παρακάτω σχήμα 0.0.5 φαίνεται μια απεικόνιση του εν λόγω μετασχηματισμού.

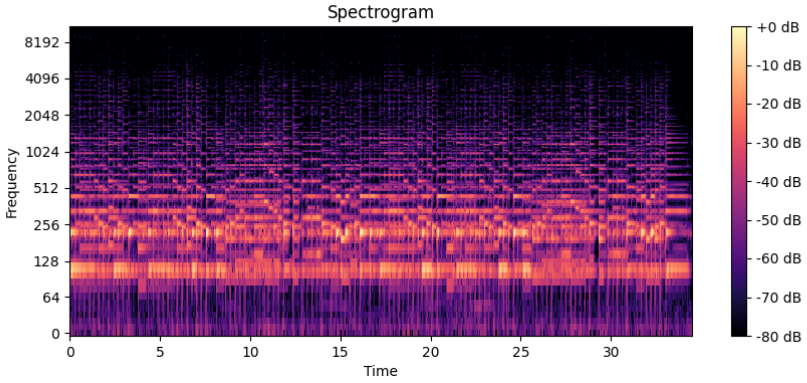


Figure 0.0.5: Φασματική αναπαράσταση ενός μουσικού ηχητικού σήματος.

Τέλος υπάρχει η κατηγοορία **συμβολικής αναπαράστασης** των μουσικών σημάτων. Μια συμβολική αναπαράσταση αποτελεί το MIDI (Musical Instrument Digital Interface), η εν λόγω αναπαράσταση δεν περιγράφει κάποιο ηχητικό κύμα αλλά περιέχει πληροφορίες, σε ψηφιακή μορφή, για την περιγραφή και τον χειρισμό μουσικών γεγονότων (events) σε πραγματικό χρόνο. Τα μουσικά γεγονότα μπορεί να αφορούν το πάτημα ή την αποδέσμευση ενός συγκεκριμένου πλήκτρου, την ταχύτητα (ένταση) με την οποία πιέζεται ένα πλήκτρο, καθώς και δεδομένα που αφορούν την κίνηση κάποιου πεντάλ ή άλλης μονάδας ελέγχου ενός ηλεκτρονικού μουσικού οργάνου. Μια αναπαράσταση MIDI φαίνεται στο παρακάτω Σχήμα 0.0.6.

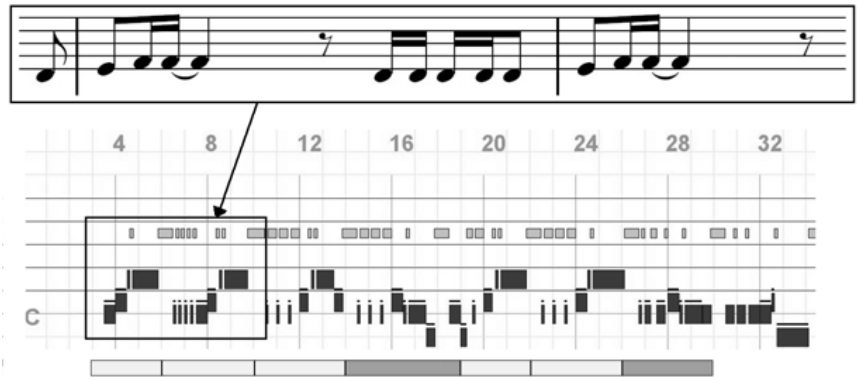


Figure 0.0.6: Αναπαράσταση MIDI [3].

Εισαγωγή στη Βαθιά Μάθηση

Συνελικτικά Νευρωνικά Δίκτυα (CNNs)

Τα Συνελικτικά Νευρωνικά Δίκτυα (CNNs) έχουν επαναστατήσει τον τομέα της επεξεργασίας εικόνας και χρησιμοποιούνται ευρέως στην Όραση Υπολογιστών, ταυτόχρονα εχουν σημαντικές εφαρμογές και στην επεξεργασία ήχου. Τα CNNs είναι ικανά να αναγνωρίζουν διάφορα ιεραρχικά μοτίβα στα δεδομένα, κάτι που τα καθιστά κατάλληλα για αναλύσεις που αφορούν τη μουσική [4, 5, 6]. Η αρχιτεκτονική ενός συνελικτικού δικτύου περιλαμβάνει συνήθως διάφορα επίπεδα και συναρτήσεις όπως επίπεδα συνέλιξης, συναρτήσεις ενεργοποίησης και επίπεδα κανονικοποίησης. Στο παρακάτω Σχήμα 0.0.7 απεικονίζεται ένα συνελικτικό δίκτυο.

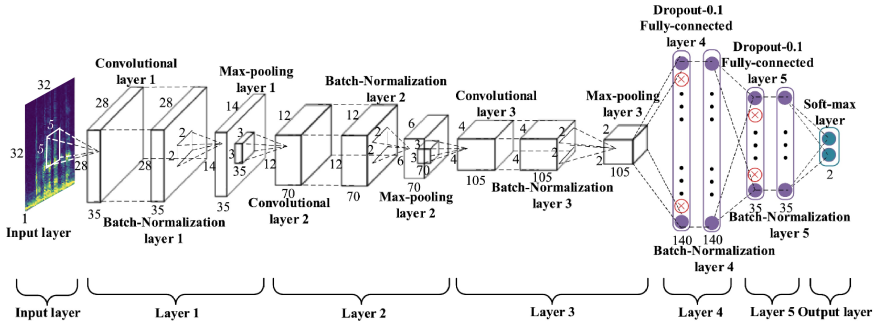


Figure 0.0.7: Ένα παράδειγμα αρχιτεκτονικής CNN. Από [7].

Πιο συγκεκριμένα, ένα συνελικτικό δίκτυο αποτελείται απο **συνελικτικά επίπεδα**, τα οποία είναι σημαντικά για την εξαγωγή χαρακτηριστικών και την αναγνώριση των μοτίβων στα δεδομένα, απο **Επίπεδα Pooling** τα οποία συναντόνται μετά τα συνελικτικά επίπεδα. Αυτά τα επίπεδα είναι υπεύθυνα για τη μείωση των διαστάσεις των δεδομένων χρησιμοποιώντας τεχνικές δειγματοληψίας. **Πλήρως Συνδεδεμένα Επίπεδα** τα οποία βρίσκονται στο τέλος της αρχιτεκτονικής, και χρησιμοποιούνται για την ταξινόμηση των δεδομένων. **Συναρτήσεις Ενεργοποίησης και Κανονικοποίησης** όπως η ReLU (Rectified Linear Unit), συχνά χρησιμοποιούνται για την εισαγωγή μη γραμμικοτήτων στο μοντέλο, ενώ επίσης εφαρμόζονται επίπεδα κανονικοποίησης για την αποφυγή του overfitting.

Εισαγωγή στα Επαναληπτικά Νευρωνικά Δίκτυα (RNNs)

Τα Επαναληπτικά Νευρωνικά Δίκτυα (RNNs) είναι μια κατηγορία νευρωνικών δικτύων που αποσκοπούν στην επεξεργασία ακολουθιακών δεδομένων, καθιστώντας τα ιδανικά για εφαρμογές στη μουσική. Τα RNNs διαθέτουν εσωτερικά στοιχεία μνήμης για να αποθηκεύουν πληροφορίες σχετικά με προηγούμενες τιμές της εισόδου, επιτρέποντας τους να καταγράφουν χρονικές εξαρτήσεις. Στο παρακάτω σχήμα 0.0.8 φαίνεται η αρχιτεκτονική των επαναληπτικών δικτύων.

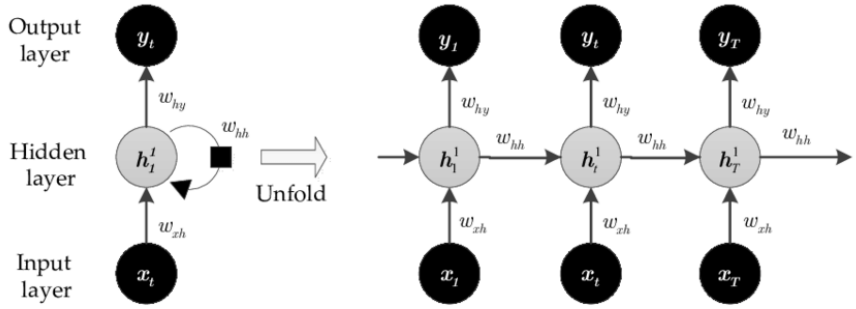


Figure 0.0.8: Ένα παράδειγμα αρχιτεκτονικής RNN. Απο [8].

Συχνά ωστόσο αντιμετωπίζουν σημαντικές προκλήσεις κατά την επεξεργασία μεγάλων ακολουθιών, όπως τα vanishing και exploding gradient problems. Για την αντιμετώπιση αυτών των προβλημάτων, έχουν αναπτυχθεί αρχιτεκτονικές όπως τα Δίκτυα Μακράς Σύντομης Προθεσμίας (LSTM) και τα Δίκτυα Μονάδας Μεταλλαγμένης Πύλης (GRU). Τα LSTM, όπως προτάθηκαν από τον Schmidhuber και άλλους [9], περιλαμβάνουν μονάδες μνήμης που μπορούν να διατηρούν πληροφορίες σε μεγάλες ακολουθίες, αντιμετωπίζοντας αποτελεσματικά το πρόβλημα της εξαφάνισης του κλίσματος. Τα Δίκτυα Μονάδας Μεταλλαγμένης Πύλης, που προτάθηκαν από την δουλειά [10], προσφέρουν μια απλοποιημένη έκδοση των LSTM με λιγότερες παραμέτρους.

Μετασχηματιστές (Transformers)

Οι αρχιτεκτονικές Transformers αποτελούν μια εξελιγμένη κατηγορία νευρωνικών δικτύων που έχουν σημειώσει σημαντική επιτυχία σε ποικίλες εφαρμογές, ακόμα και σε προβλήματα επεξεργασίας φυσικής γλώσσας. Η κύρια καινοτομία των αρχιτεκτονικών, είναι η απουσία επαναλαμβανώμενων δικτύων RNN ή LSTM και η χρήση μηχανισμών προσοχής (attention mechanisms). Η βασική ιδέα είναι η ικανότητα του μοντέλου να εστιάσει σε διάφορα τμήματα της εισόδου κατά την επεξεργασία ενός μόνο τμήματος. Η αρχιτεκτονική τους αποτελείται απο τα επίπεδα προσοχής, γνωστά ως self-attention layers, όπου κάθε στοιχείο εισόδου μπορεί να "στρέψει την προσοχή" του σε άλλα στοιχεία, βάσει της αναπαράστασης του. Κάθε επίπεδο προσοχής εξετάζει τις συνδέσεις μεταξύ όλων των στοιχείων της εισόδου, ενισχύοντας την ικανότητα του μοντέλου να αντιλαμβάνεται το ευρύτερο νόημα των δεδομένων. Ακόμα, οι Transformers χρησιμοποιούν πολυεπίπεδες feedforward νευρωνικές επιφάνειες και στρώματα κανονικοποίησης.

Νευρωνικά Δίκτυα Shuffle-Exchange

Επισκόπηση των Δικτύων Benes

Τα Νευρωνικά Δίκτυα Benes όπως φαίνεται στο παρακάτω σχήμα 0.0.9, χρησιμοποιούν την δομή ενός δικτύου Benes το οποίο βρίσκει εφαρμογή σε εργασίες δρομολόγησης πακέτων σε υπολογιστικά δίκτυα. Το εν λόγω δίκτυο είναι ικανό να δρομολογεί αποτελεσματικά σήματα απο την είσοδο στην έξοδο με βάση ένα μετασχηματισμό περιστροφής της ακολουθίας.

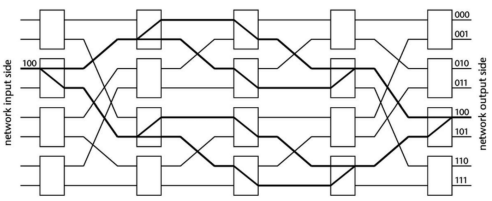


Figure 0.0.9: Σχηματισμός Δικτύου Benes. Από [11].

Νευρωνικά Δίκτυα Shuffle-Exchange

Τα νευρωνικά δίκτυα Benes αντικαθιστούν κάθε διακόπτη ενός δικτύου Benes με ένα Switch Unit, μια συνάρτηση 2-προς-2 όπως φαίνεται στο σχήμα 0.0.10. Το πρώτο στρώμα του δικτύου αποτελείται από μια σειρά Switch Units ενώ ακολουθεί ένα στρώμα αναδιάταξης, το οποίο αναδιατάσει το σήμα της εισόδου με βάση ένα καθορισμένο πρότυπο αναστροφής.

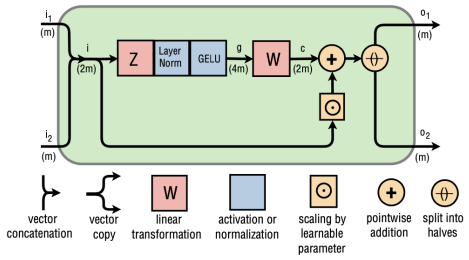


Figure 0.0.10: Μονάδα Switch Unit [12].

Σχετική Βιβλιογραφία

Βαθιά Νευρωνικά Δίκτυα

Τα βαθιά νευρωνικά δίκτυα πλέον είναι η αποδοτικότερη λύση στο πρόβλημα του διαχωρισμού πηγών, καθώς επιτυγχάνουν διαχωρισμό με υψηλή ποιότητα χωρίς να εισάγουν θορύβους στα σήματα. Οι κατηγορίες των βαθιών δικτύων που επικεντρώνονται στο πρόβλημα του διαχωρισμού πηγών είναι δύο, εκείνες που δεν στηρίζονται στην αξιοποίηση κάποιας περαιτέρω πληροφορίας εκτός απο αυτής του σήματος εισόδου και αυτές που αξιοποιούν πληροφορίες που αφορούν τις νότες που παίζει το όργανο το οποίο σκοπεύουμε να διαχωρίσουμε.

Βαθιά Νευρωνικά Δίκτυα χωρίς χρήση πληροφορίας νοτών

Τα δίκτυα τα οποία δεν κάνουν χρήση κάποιας περαιτέρω πληροφορίας των σημάτων που πρέπει να διαχωριστούν χωρίζονται και αυτά σε δύο κατηγορίες ανάλογα με τον "χώρο" στον οποίο επεξεργάζονται τα δεδομένα. Έχουμε τεχνικές που επικεντρώνονται στα κύματα, χρησιμοποιώντας τη μονοδιάστατη "φυσική" αναπαράσταση ήχου, και μεθόδους που επικεντρώνονται στα φασματογραφήματα, χρησιμοποιώντας μια δισδιάστατη χρονικοσυχνοτική αναπαράσταση των δεδομένων.

Οι αρχιτεκτονικές εκείνες που λειτουργούν πάνω σε φασματογραφήματα, έχουν ως στόχο την εκτίμηση του φασματογραφήματος του οργάνου ενδιαφέροντος. Παράγουν μια μάσκα την οποία εφαρμόζουν στο συνολικό σήμα και εξάγουν το επιθυμητό όργανο. Παρακάτω στην εικόνα 0.0.11 βλέπουμε την εφαρμογή μιας *δυαδικής μάσκας* πάνω σε ένα αρχικό σήμα και το αποτέλεσμα της εν λόγω πράξης. Αξίζει να σημειωθεί πως οι μάσκες οι οποίες προβλέπουν τα μοντέλα αυτά δεν είναι δυαδικές αλλά κάθε τους στοιχείο αποτελείται απο αριθμούς δεκαδικής ακρίβειας.

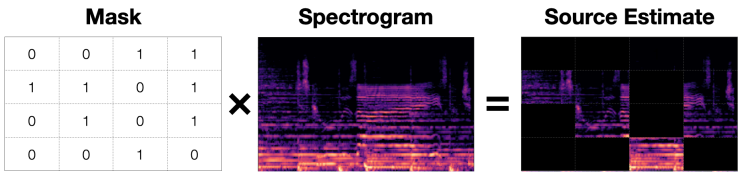


Figure 0.0.11: Δυαδική μάσκα [13]

Τα δεδομένα στα φασματογραφήματα, είναι πιο συμπαγή από τις χρονικές αναπαραστάσεις πλάτους, πράγμα το οποίο σημαίνει ότι τα μοντέλα απαιτούν λιγότερους υπολογισμούς, μειώνοντας τον χρόνο εκπαίδευσης. Είναι

σημαντικό να σημειωθεί πως οι περισσότερες προσεγγίσεις παραβλέπουν τη φάση του σήματος κατά την εκτίμηση των πηγών.

Στο βάθος του χρόνου, τα βαθιά νευρωνικά δίκτυα εξελίσσονταν σε βάθος και πολυπλοκότητα, ενσωματώνοντας διάφορα επίπεδα για την βελτίωση της απόδοσης του διαχωρισμού. Αναγνωρίζοντας ότι τα ηχητικά σήματα μπορεί να έχουν χρονικές εξαρτήσεις, οι ερευνητές ενέταξαν τα επαναληπτικά δίκτυα όπως το [14]. Ο Jannson στο [15] παρουσίασε μια προσαρμογή του U-Net [16] για τη λύση του προβλήματος διαχωρισμού πηγών μουσικής, η οποία έκανε χρήση συνελίξεων όπως φαίνεται στο παρακάτω σχήμα 0.0.12.

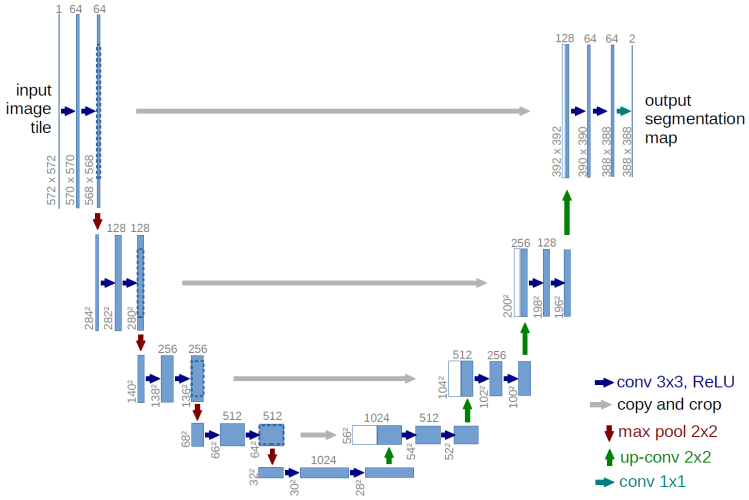


Figure 0.0.12: Αρχιτεκτονική U-Net για τον Χωρισμό Πηγών από [15]

Ταυτόχρονα αναπτύχθηκαν τεχνικές οι οποίες αξιοποίησαν αυτούσια την κυματομορφή χωρίς να εφαρμόσουν κάποιο μετασχηματισμό. Ένα κύριο πλεονέκτημα της λειτουργίας στη κυματομορφή είναι η διατήρηση της πληροφορίας της φάσης η οποία χάνεται σε φασματικές αναπαραστάσεις. Δυο απο τις δουλειές οι οποίες αποτέλεσαν πηγή έμπνευσης για την αρχιτεκτονική που ασχολούμαστε στη συγκεκριμένη διπλωματική εργασία είναι οι TasNet [17] και Wave-U-Net [4].

Το **TasNet** χρησιμοποιεί διαδοχικά συνελικτικά επίπεδα για να καταφέρει να δημιουργήσει την μάσκα διαχωριδσμού του επιθυμητού σήματος. Όπως φαίνεται στο παρακάτω σχήμα 0.0.13 ο διαχωριστής (separator) δημιουργεί την μάσκα η οποία εφαρμόζεται στην αρχική κωδικοποίηση του σήματος και έπειτα αποκωδικοποιείται στο τελευταίο στάδιο. Ο κωδικοποιητής και ο αποκωδικοποιητής αποτελούνται απο μια σειρά συνελίξεων, επιπέδων κανονικοποίησης και συναρτήσεων ενεργορποίησης.

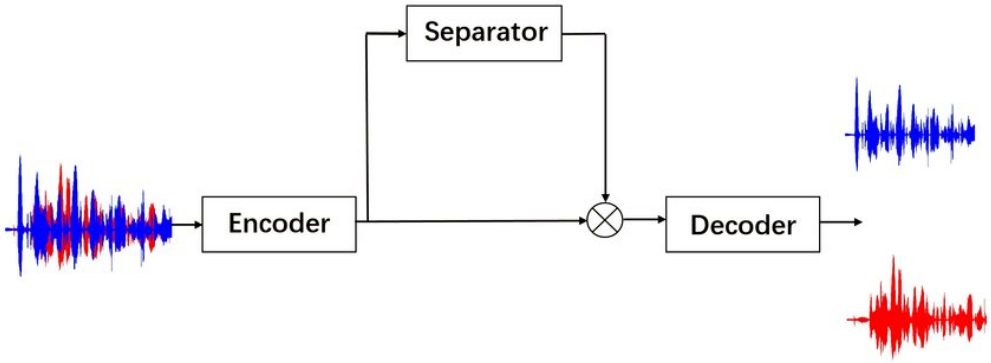


Figure 0.0.13: Αρχιτεκτονική TasNet [17]

Απο την άλλη το **Wave-U-Net**, όπως φαίνεται στο σχήμα 0.0.14 ,παρέχει ακόμα μια αποτελεσματική αρχιτεκ-

τονική προσαρμοσμένη από το μοντέλο U-Net για μονοδιάστατα σήματα ήχου. Συγκεκριμένα η αρχιτεκτονική αυτή αποτελείται από:

- Ένα μονοπάτι για τη συμπύκνωση πολυ-κλίμακων πληροφοριών βασισμένο σε συνελίξεις και δειγματοληψία του σήματος,
- Ένα επίπεδο συνέλιξης το οποίο επεξεργάζεται τα πιο "πυκνά" χαρακτηριστικά του αρχικού σήματος,
- Ένα μονοπάτι επανασύνθεσης που αποκαθιστά το σήμα στην αρχική του διάσταση, χρησιμοποιώντας συνδέσεις παράκαμψης για την ολοκλήρωση των χαρακτηριστικών απο προηγούμενα στάδια.

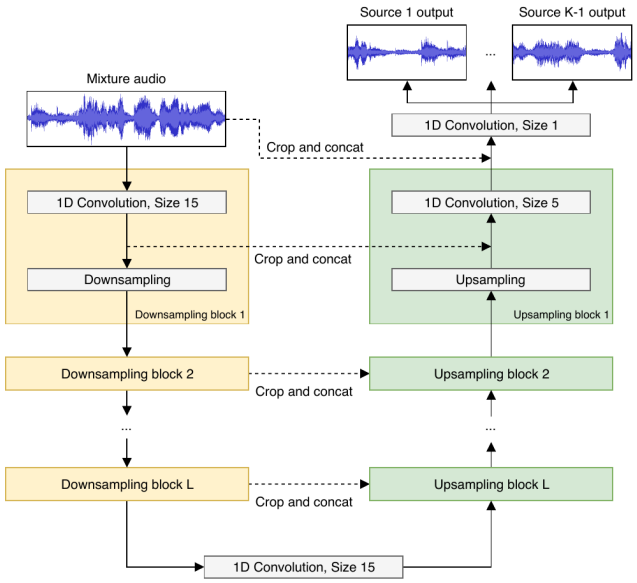


Figure 0.0.14: Αρχιτεκτονική Wave-U-Net για Διαχωρισμό Πηγών από [4]

Βασισμένη στα TasNet και Wave-U-Net είναι η αρχιτεκτονική Demucs η οποία στα πρώτα στάδια της χρησιμοποίησε τη δύναμη του μονοδιάστατου χρονικού πεδίου για την επεξεργασία σήματος. Ωστόσο, όσο εξελίσσονταν, προσαρμόστηκε έτσι ώστε να κάνει χρήση και των χρονικών και των φασματικών πεδίων των σημάτων. Αυτή η διπλή προσέγγιση επιτρέπει μια πιο σφαιρική ανάλυση και εξαγωγή δεδομένων απο τα ηχητικά σήματα.

Βαθιά Νευρωνικά Δίκτυα με χρήση πληροφορίας νοτών

Οι αλγόριθμου διαχωρισμού πηγών μουσικής που βασίζονται σε πληροφορία νοτών χρησιμοποιούν τις μουσικές παρτιτούρες για να ενισχύσουν την ακρίβεια στον διαχωρισμό των μεμονωμένων οργάνων.

Μια σημαντική εργασία πάνω στην εν λόγω προσέγγιση είναι η [18]. Το συγκεκριμένο σύστημα χρησιμοποιεί μια αρχιτεκτονική U-Net. Αυτή η αρχιτεκτονική περιλαμβάνει έναν μονοπάτι κωδικοποιητή, απο αναδρομικό δίκτυο GRU και απο ένα μονοπάτι αποκωδικοποίησης. Πληροφορίες για τις νότες που παίζουν τα όργανα ενσωματώνονται στο δίκτυο σε ένα συγκεκριμένο στάδιο του κωδικοποιητή. Η επισκόπηση της αρχιτεκτονικής φαίνεται στο Σχήμα 0.0.15.

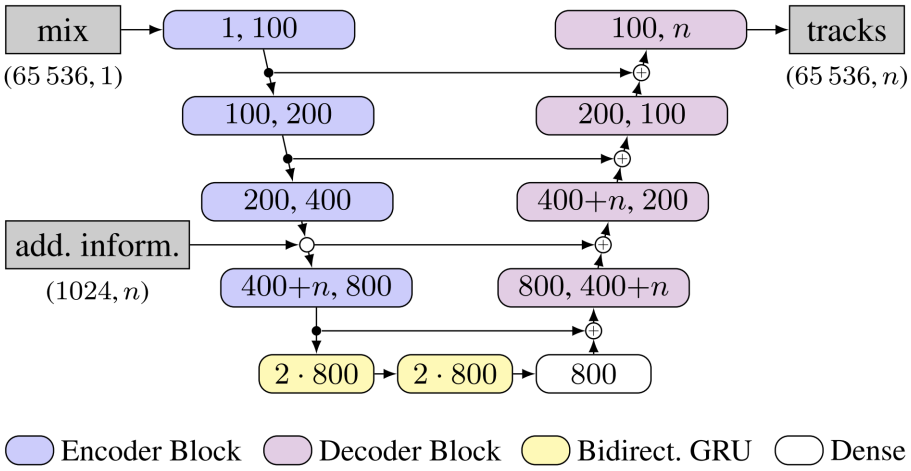


Figure 0.0.15: Αρχιτεκτονική βασισμένη στο U-Net και στην εισαγωγή δεδομένων νοτών του [18].

Τεχνικές Επαύξησης Δεδομένων

Η επαύξηση δεδομένων αποτελεί μια κεντρική στρατηγική στην εκπαίδευση βαθιών νευρωνικών δικτύων, ειδικά όταν το διαθέσιμο σύνολο δεδομένων είναι περιορισμένο [19] καθώς συμβάλλει στη δημιουργία δικτύων διαχωρισμού που γενικεύουν καλύτερα. Παρακάτω παρουσιάζονται οι διάφορες τεχνικές που χρησιμοποιούνται:

1. **Αντιστροφή Καναλιών (FlipChannels):** Αυτή η τεχνική περιλαμβάνει την τυχαία ανταλλαγή του αριστερού και δεξιού καναλιού απο ένα στερεοφωνικό αρχείο ήχου για κάθε όργανο.

2. **Μετατόπιση (Shift):** Η τυχαία μετατόπιση στο χρόνο των καναλιών βοηθά το μοντέλο να μάθει αναλλοίωτες αναπαραστάσεις που δεν εξαρτώνται από τη θέση του κάθε οργάνου στο χρόνο.

3. **Αναμείξη (Remix):** Μια τεχνική όπου, εντός ενός μόνο batch, όργανα από ένα τραγούδι ανταλλάσσονται με το αντίστοιχα όργανα από ένα διαφορετικό τραγούδι.

4. **Κλίμακα (Scale):** Η τυχαία κλιμάκωση ενός σήματος με έναν πολλαπλασιαστή της τάξης [0.5, 1.25].

5. **Αντίθετη τοποθέτηση στον "χώρο" (OppositePanning):** Αυτή η τεχνική, υλοποιήθηκε στη παρούσα διπλωματική εργασία. Δημιουργεί ποικίλες στερεοφωνικές εικόνες προσαρμόζοντας τη θέση της κάθε κιθάρας σε ένα στερεοφωνικό αρχείο.

Η Ανάγκη για Εκπαίδευση Αμετάβλητη σε Μεταθέσεις

Ο διαχωρισμός πηγής σε ντουέτα κλασικής κιθάρας μοιάζει με τον διαχωρισμό ομιλητών. Όπως δύο ομιλητές μπορεί να έχουν κοινά χαρακτηριστικά ή αλλιώς ίδια χροιά στη φωνή τους, έτσι συμβαίνει και στις κιθάρες. Λαμβάνοντας υπόψη αυτήν την παρομοίωση, γίνεται εμφανές ότι οι μεθοδολογίες και οι τεχνικές που έχουν αναπτυχθεί για τον διαχωρισμό ομιλητών μπορούν να προσφέρουν και στην δική μας έρευνα.

Η εκπαίδευση ενός μοντέλου έτσι ώστε να γίνει αμετάβλητο στις μεταθέσεις των εισόδων του είναι μια τεχνική που συναντάται στην διάκριση πολλαπλών ομιλητών. Υιοθετώντας την τεχνική με την ονομασία "Permutation Invariant Training" (PIT), προσεγγίζουμε το πρόβλημα μας ως ένα πρόβλημα καθαρά διαχωρισμού και όχι κατηγοριοποίησης της κάθε κιθάρας. Η βασική στρατηγική του PIT είναι να εντοπίζει την βέλτιστη αντιστοίχιση εξόδου-στόχου αυτή δηλαδή που δίνει το ελάχιστο σφάλμα, και με βάση αυτό το σφάλμα να εκπαιδεύει το δίκτυο. Αυτή η μέθοδος προσφέρει μια άμεση λύση στις περιπτώσεις όπου τα μοντέλα διαχωρίζουν σωστά τα σήματα αλλά τα κατευθύνουν σε λάθος σειρά στις εξόδους τους. Η διαδικασία της εν λόγω τεχνικής φαίνεται στο παρακάτω σχήμα 0.0.16.

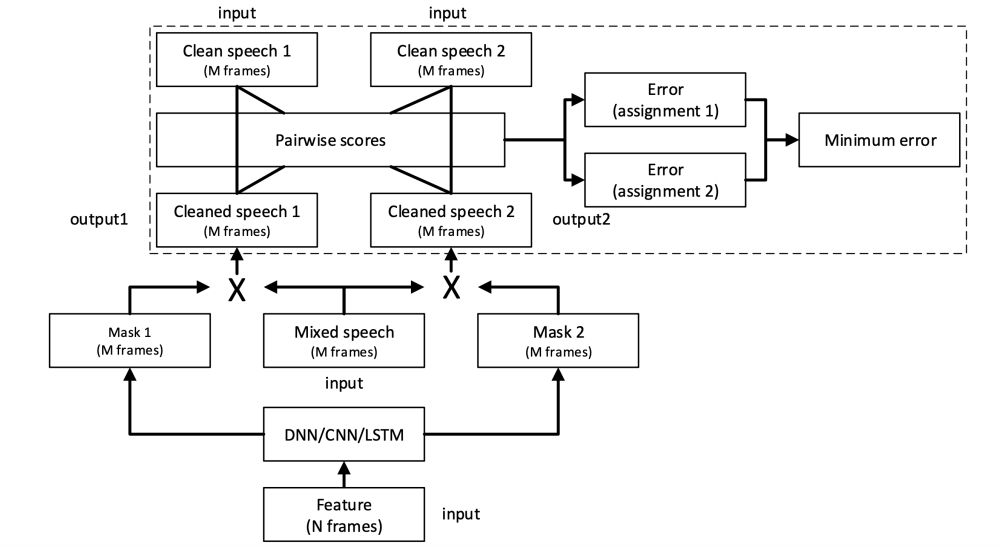


Figure 0.0.16: Το μοντέλο εκπαίδευσης (PIT) [20].

Σύνολα δεδομένων

Μέχρι στιγμής έχουν δημιουργηθεί και χρησιμοποιηθεί πολλά σύνολα δεδομένων για το γενικότερο πρόβλημα του διαχωρισμού πηγών. Κύριο χαρακτηριστικό των συνόλων αυτών είναι πως για κάθε μουσικό κομμάτι περιέχουν αρχεία ήχου για το κάθε όργανο ξεχωριστά. Για την εργασία μας, τα σύνολα δεδομένων με βάση τα όργανα τα οποία περιέχουν μπορούν να διακριθούν σε δυο κατηγορίες. Τη κατηγορία που περιέχει διαφορετικά όργανα να παίζουν ταυτόχρονα και τη κατηγορία που περιέχει ίδια όργανα με κοντινές χροιές να παίζουν ταυτόχρονα. Στην πρώτη κατηγορία ανοίκουν σύνολα όπως τα MUSDB18-HQ [21], MedleyDB [22], Slakh [23], URMP [24] και MIR-1K [25]. Ενώ στη δεύτερη κατηγορία ανήκουν σύνολα όπως το GuitarSet [26] και το EnsembleSet [27], και τα δυο σύνολα πέρα απο τα αρχεία ήχου περιέχουν συμβολική πληροφορία για τις νότες που παίζει το κάθε όργανο.

Μετρικές για αξιολόγηση του Διαχωρισμού Πηγών

Το πρόβλημα του διαχωρισμού μουσικών πηγών απο τη φύση του είναι ένα πρόβλημα το οποίο έχει ανάγκη την ανθρώπινη αξιολόγηση και δεν αρκούν μόνο οι μετρικές για την αξιολόγηση της επίδοσης κάποιου μοντέλου. Υπάρχουν πολλές μετρικές οι οποίες έχουν χρησιμοποιηθεί μέχρι σήμερα με τις πιο εδραιωμένες πλέον να είναι το SDR, SI-SDR, SAR, ISR και SIR. Παρακάτω ορίζουμε 2 απο τις σημαντικότες το SDR και το SI-SDR.

$$\text{SDR} = 10\log_{10}\frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}\|^2}.$$

Το SDR είναι μια μετρική που μετρά την αναλογία του σήματος προς την παραμόρφωση που εισάγει ο αλγόριθμος διαχωρισμού. Αντιπροσωπεύει πόσο καλά ένα σύστημα διαχωρισμού πηγών έχει απομονώσει την επιθυμητή πηγή από τυχόν παρεμβολές ή παραμορφώσεις. Υψηλότερη τιμή SDR υποδεικνύει καλύτερη απόδοση.

$$\text{SI-SDR} = 10\log_{10}\frac{\|\alpha s_{\text{target}}\|^2}{\|\alpha s_{\text{target}} - s_{\text{est}}\|^2},$$

Το SI-SDR είναι μια βελτιωμένη έκδοση της SDR που είναι ανεξάρτητη της έντασης των σημάτων. Πιο συγκεκριμένα δεν λαμβάνει υπόψιν μια διαφορά σε ένταση απο τα αρχικά σήματα την οποία ενδέχεται να εισάγει ο αλγόριθμος.

Δημιουργία Συνόλου Δεδομένων

Ηχογράφηση Συνόλου Δεδομένων GuitarDuets

Λαμβάνοντας υπόψη την έλλειψη δεδομένων για εκπαίδευση μοντέλων τεχνητής νοημοσύνης στον διαχωρισμό πηγών ίδιας χροιάς, κυριώς για τα σενάρια που περιλαμβάνουν το μουσικό όργανο της κλασικής κιθάρας, δημιουργήσαμε ένα σύνολο δεδομένων που καλύπτει πολυφωνικές ηχογραφήσεις ντουέτων κλασικής κιθάρας, έτσι ώστε να διευκολυνθεί η ανάπτυξη και αξιολόγηση τεχνικών διαχωρισμού πηγών για τις περιπτώσεις αυτές.

Ρύθμιση Ηχογράφησης

Ηχογραφήθηκαν απο ένα ντουέτο κλασικής κιθάρας τα κομμάτια που συνιστούν το μουσικό ρεπερτόριο μιας συναυλίας. Η ηχογράφηση έγινε σε κατάλληλο δωμάτιο ηχογράφησης με ακουστική βελτίωση. Κάθε κλασική κιθάρα ηχογραφήθηκε χρησιμοποιώντας το μικρόφωνο υψηλής ποιότητας (Presonus PM-2). Κατά τη διάρκεια της ηχογράφησης, χρησιμοποιήθηκαν τέσσερις διαφορετικές κλασικές κιθάρες για να εξασφαλιστεί μια ποικιλία χροιών στο σύνολο δεδομένων μας. Παρά την προσεκτική τοποθέτηση των μικροφώνων και τις ρυθμίσεις, οι ηχογραφήσεις παρουσίασαν το φαινόμενο (microphone bleeding) διαρροή μικροφώνου, όπου ο ήχος από τη μια κιθάρα ηχογραφήθηκε στο μικρόφωνο που καταγράφει την άλλη κιθάρα.

Η δημιουργία ενός ολοκληρωμένου και εκφραστικού συνόλου δεδομένων απαιτούσε προσεκτικό σχεδιασμό των εκτελέσεων των κομματιών. Διαμορφώσαμε μια επιλογή διαφόρων έργων κλασικής μουσικής, που αντιπροσωπεύουν διάφορα στυλ, ρυθμούς και πολυπλοκότητες. Κατά τη διάρκεια των ηχογραφήσεων, καταγράφηκαν πολλές ηχογραφήσεις για ορισμένα κομμάτια, προκειμένου να υπάρχουν διαφορετικές ερμηνείες.

Προεπεξεργασία

Πριν ενσωματώσουμε τα δεδομένα των ηχογραφήσεων στους αλγόριθμους διαχωρισμού πηγών, πραγματοποιήσαμε βασικά βήματα προεπεξεργασίας για να εξασφαλίσουμε τη συνοχή και συμβατότητα των δεδομένων. Αυτό περιλαμβάνει την κανονικοποίηση των εντάσεων, την αφαίρεση ανεπιθύμητων θορύβων και τον διαχωρισμό των ηχογραφήσεων σε κατάλληλα κομμάτια. Στον παρακάτω πίνακα υπάρχει μια συνολική επισκόπηση των κομματιών που ηχογραφήθηκαν μαζί με τη διάρκειά τους.

Όνομα	**Διάρκεια (Δευτερόλεπτα)**
Valses Poeticos 1 (Enrique Granados)	85.5
Valses Poeticos 1 2nd	97.5
Valses Poeticos 2	92.0
Valses Poeticos 2 2nd	25.0
Valses Poeticos 2 3rd	49.5
Valses Poeticos 3	88.5
Valses Poeticos 3 2nd	86.5
Valses Poeticos 4	116.0
Valses Poeticos 5	46.5
Valses Poeticos 5 2nd	46.0
Valses Poeticos 6	60.0
Valses Poeticos 7	87.0
Valses Poeticos 8	49.0
Valses Poeticos 9	91.0
Valses Poeticos 10	91.5
Summer Garden Suite 1 Opening (Sergio Assad)	82.0
Summer Garden Suite 1 Opening 2nd	73.0
Summer Garden Suite 2 Summer Garden	156.0
Summer Garden Suite 2 Summer Garden 2nd	142.0
Summer Garden Suite 3 Farewell	175.0
Summer Garden Suite 3 Farewell 2nd	168.0
Summer Garden Suite 4 Butterflies	175.0
Tango 1 (Astor Piazzolla)	338.0
Tango 1 2nd	331.0
Tango 2	297.5
Tango (N. Mavroudis)	154.0
demo1	34.5
demo2	66.0
demo3	49.5
demo4	27.0
demo5	41.5
demo6	22.5
demo7	27.0
demo8	47.5
Συνολική Διάρκεια	**58.6 λεπτά**

Table 1: Ονόματα Ηχογραφήσεων και διάρκεια του συνόλου δεδομένων GuitarDuets.

Δημιουργία Δεδομένων με Εικονικά Όργανα

Εισαγωγή στα Εικονικά Όργανα

Ένα εικονικό όργανο, στο πλαίσιο της παραγωγής μουσικής και της ψηφιακής τεχνολογίας ήχου, αναφέρεται σε μια προσομοίωση ή αναπαραγωγή ενός αληθινού μουσικού οργάνου μέσα απο τον ηλεκτρονικό υπολογιστή. Σήμερα, τα εικονικά όργανα παίζουν ένα ζωτικό ρόλο στη δημιουργία μουσικής, δημοκρατικοποιώντας την πρόσβαση σε ένα εκτεταμένο φάσμα ήχων. Η ενσωμάτωση των εικονικών οργάνων στα προγράμματα δημιουργίας μουσικής (Digitan Audio Workstations DAWs) έχει επαναπροσδιορίσει τον τρόπο με τον οποίο οι καλλιτέχνες δημιουργούν τις συνθέσεις τους, δίνοντάς τους τη δυνατότητα να "παίζουν" οποιοδήποτε όργανο θελήσουν, χωρίς στη πραγματικότητα να το ηχογραφούν οι ίδιοι. Το όργανο το οποίο χρησιμοποιούν υπάρχει μέσα στο ψηφιακό περιβάλλον και παράγει πανομοιότυπους ήχους με το αντίστοιχο πραγματικό όργανο, είτε μέσω τεχνολογίας βασισμένης σε δείγματα είτε σε τεχνολογία βασισμένη σε σύνθεση. Υπάρχουν δύο βασικοί τύποι εικονικών οργάνων. Τα εικονικά όργανα βασισμένα σε δείγματα, τα οποία εξαρτώνται από εκτεταμένες συλλογές ηχογραφημένων δειγμάτων από πραγματικά όργανα. Και τα εικονικά όργανα βασισμένα σε αλγόριθμους παραγωγής ήχου, οι οποίοι προσομοιώνονουν τη συμπεριφορά των οργάνων [28].

Για να δημιουργήσουμε τα δεδομένα απο τα εικονικά όργανα χρησιμοποιήσαμε το **Εικονικό Όργανο Native Instruments: "Session Guitarist - Picked Nylon"** [29], το οποίο είναι ένα λογισμικό βασισμένο σε δείγματα, σχεδιασμένο για να αποτυπώσει τον ήχο της κλασικής κιθάρας.

Για να καταφέρουμε να δώσουμε της εντολές στο εικονικό όργανο να παράξει τις μουσικές νότες οι οποίες θα συνιστούν τα κομμάτια, χρησιμοποιήσαμε MIDI δεδομένα απο την κοινότητα του MuseScore [30], και δουλέψαμε στο περιβάλλον επεξεργασίας ήχου Logic Pro X [31]. Για κάθε αρχείο MIDI, επιλέξαμε προσεκτικά διαφορετικές ρυθμίσεις του εικονικού οργάνου έτσι ώστε να έχουμε διαφορετικές χροιές σε κάθε κιθάρα.

Ένας βασικός στόχος αυτής της διαδικασίας δημιουργίας συνόλου δεδομένων ήταν η επέκταση του συνόλου δεδομένων που έχει προκύψει μέσω ηχογράφησης καθώς υπάρχει δυσκολία στον εντοπισμό πραγματικών ηχογραφήσεων μεμονωμένων κιθάρων που παίζουν μαζί. Για την ολοκλήρωση της παραπάνω διαδικασίας, εξήγαμε κάθε μεμονωμένη εκτέλεση κιθάρας ως ένα αρχείο WAV 16-bit με συχνότητα δειγματοληψίας 44.100 Hz. Στον παρακάτω πίνακα παρουσιάζεται μια συνολική επισκόπηση των κομματιών που δημιουργήθηκαν και της διάρκειας τους.

Αριθμός Κομματιού	**Διάρκεια (Δευτερόλεπτα)**	
Track1	Bach, Minuet in G major, BWV Anh 114	120.0
Track2	Bach Prelude n3 BWV 935	120.0
Track3	Blind Guardian The Bard's Song	45.818
Track4	Unkoown (No named Provided by MuseScore)	32.0
Track5	Unkoown (No named Provided by MuseScore)	36.0
Track6	Unkoown (No named Provided by MuseScore)	38.571
Track7	Marcello/Bach - Concerto in D minor	109.5
Track8	Duo en Sol op.27 n°8 - Ferdinando Carulli	57.6
Track9	BWV 304 Bach. J.S. Choral; Eins ist noth, ach Herr, dies Eine	93.103
Track11	Sir Edward Elgar - Pomp and Circumstance March No.1	374.4
Track12	Sibelius Etude Op.76 No.2	192.0
Track13	The Police - Every Breath You Take	214.839
Track14	Jordon Drumgoole - Four Short Seasons for Guitar Duet	300.632
Track15	Gerald Schwertberger - Blue and Rythmic Duets	577.92
Track16	Unkoown (No named Provided by MuseScore)	500.909
Track17	J.O.Marques: Six Easy Duets for Guitars - No.1 in C major	56.048
Track18	J.O.Marques: Six Easy Duets for Guitars - No.2 in G major	92.857
Track19	J.O.Marques: Six Easy Duets for Guitars - No.4 in F major	106.667
Track20	J.O.Marques: Six Easy Duets for Guitars - No.6 in C major	89.302
Track21	Suite c minor (BWV997) - Preludio for tenor	154.884
Track22	Mazurka - Francesco Tarrega (1852 - 1909) - Duo	63.717
Track23	Milonga Guitar Duo	105.366
Track24	NIGHTWISH - Ever Dream	83.137
Track25	Poco Allegretto - Ferdinando Carulli (1770 - 1841) - Duo	94.815
Track26	Recuerdos de la Alhambra - Francisco Tárrega	240.0
Track27	Scherzino Mexicano	141.639
Track28	Unkoown (No named Provided by MuseScore)	129.836
Track29	Unknown	63.066
Track30	Unknown	57.81
Track31	Unknown	151.579
Track32	Unknown	122.553
Track33	Terpsichore - Duo op.45 - José Ferrer y Esteve	208.0
Track34	Louis Moreau Gottschalk - The Dying Poet	261.818
Track35	Ferdinando Carulli Trois Noctures op.90	727.183
Track36	Unkoown (No named Provided by MuseScore)	604.337
Συνολική Διάρκεα		**106 λεπτά**

Table 2: NI Dataset

RSE

Αρχιτεκτονική

Το Residual Shuffle-Exchange Network (RSE) προκύπτει από τη θεμελιώδη δομή που είναι γνωστή ως δίκτυο Benes. Αυτό το δίκτυο λειτουργεί ως κρίσιμος μηχανισμός για τη διαδικασίας "ανακάτεματος" και ανταλλαγής χαρακτηριστικών μέσα σε ένα βαθύ νευρωνικό δίκτυο. Το δίκτυο Benes είναι γνωστό για τη δυνατότητά του να χειρίζεται αποτελεσματικά τις περιστροφές των εισόδων, η οποία είναι μια θεμελιώδης λειτουργία σε εργασίες όπως η ταξινόμηση και τα δίκτυα δρομολόγησης. Στο πλαίσιο του RSE, το δίκτυο Benes υποστηρίζει τη δυνατότητα του μοντέλου να διαχειρίζεται και να μαθαίνει από τα υψηλής διάστασης δεδομένα ενός σήματος ήχου. Αποτελείται από δύο τμήματα Benes όπως φαίνεται στο παρακάτω Σχήμα 0.0.17.

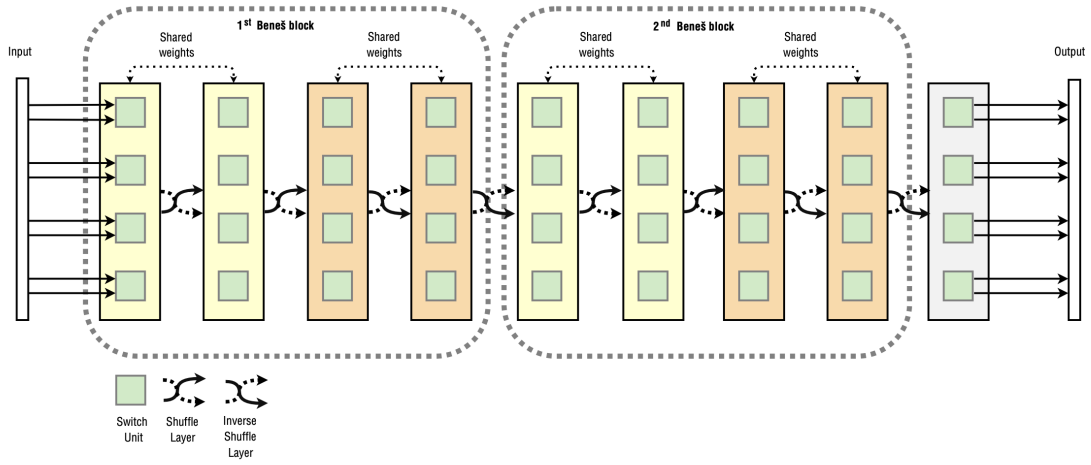


Figure 0.0.17: Δίκτυο Υπολοίπου Shuffle-Exchange με δύο τμήματα Benes και οκτώ εισόδους [12].

Αντικαθιστώντας τα Switch Units του Benes Network με την Residual Shuffle Exchange Unit όπως φαίνεται στο σχήμα 0.0.18, το δίκτυο RSE αξιοποιεί τα ενσωματωμένα πλεονεκτήματα του δικτύου Benes, ενισχύοντας την ικανότητά του να μαθαίνει και να γενικεύει σε δεδομένα ήχου.

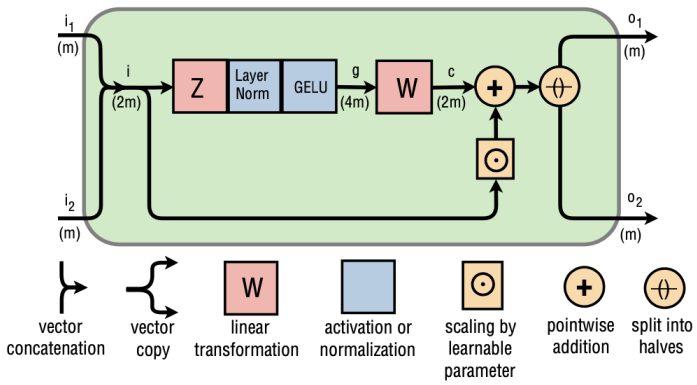


Figure 0.0.18: Αρχιτεκτονική του Switch Unit, το οποίο χρησιμοποιείται στη θέση των blocks του Benes Δικτύου [12].

Το δίκτυο RSE χρησιμοποιεί συνελίξεις οι οποίες εφαρμόζονται πριν από το κύριο δίκτυο, με σκοπό να αυξήσουν τον αριθμό των χαρακτηριστικών καινα μειώσουν το μήκος το μήκος της εισόδου [12]. Η συνολική αρχιτεκτονική φαίνεται στο παρακάτω σχήμα 0.0.19.

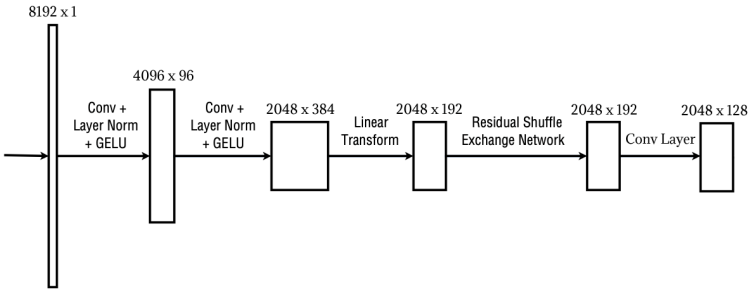


Figure 0.0.19: Η συνολική αρχιτεκτονική του RSE Δικτύου [12].

Η προσέγγισή μας μετατρέπει την αρχιτεκτονική του δικτύου Residual Shuffle-Exchange (RSE) απο ένα σύστημα μονής εξόδου σε ένα σύστημα διπλής εξόδου. Αυτό το σύστημα είναι ικανό να δημιουργεί ξεχωριστές "παρτιτούρες" για κάθε κιθάρα απο το ηχητικό αρχείο εισόδου.

Ένα κύριο επιχείρημα υπέρ της επιλογής μιας αρχιτεκτονικής επικεντρωμένης στη μεταγραφή αντί για ένα παραδοσιακό μοντέλο διαχωρισμού έγκειται στα εγγενή πλεονεκτήματα των συστημάτων μεταγραφής στη χειρισμό ξεχωριστών νοτών. Δεδομένου ότι οι αρχιτεκτονικές μεταγραφής έχουν ήδη επιδείξει σημαντική απόδοση στην αναγνώριση νοτών που παίζονται απο ένα όργανο, η στρατηγική μας εκμεταλλεύεται αυτό το πλεονέκτημα για να βελτιώσει την απόδοση διαχωρισμού. Με το να μεταγράφει το μοντέλο ακριβώς όλες τις νότες που υπάρχουν σε ένα μουσικό αρχείο, το επόμενο βήμα είναι να αναθέσει κάθε νότα στη σωστή κιθάρα. Στόχος είναι το μοντέλο να μάθει και να κατανοήσει τις φυσικές συσχετίσεις και αλληλοεξαρτήσεις μεταξύ των νοτών της κάθε κιθάρας. Για παράδειγμα, το μοντέλο μπορεί να μάθει πώς η παρουσία μιας συγκεκριμένης νότας σε μια κιθάρα μπορεί συχνά να αποκλείσει την ταυτόχρονη εκτέλεση ορισμένων άλλων νοτών στο ίδιο όργανο. Πιστεύουμε ότι η επίτευξη αυτού του στόχου και η ανακάλυψη αυτών των συσχετίσεων είναι πιο προκλητικές για ένα μοντέλο διαχωρισμού ήχου που επικεντρώνεται μόνο στη διάκριση των ήχων. Αντίθετα, ένα μοντέλο μεταγραφής, που εξάγει τις νότες που παίζονται σε μορφή δυαδικού διανύσματος $\mathbf{y} \in \{0,1\}^{128}$, παρέχει έναν πιο ξεκάθαρο δρόμο για την κατανόηση αυτών των συσχετίσεων.

Έχουμε δοκιμάσει 2 διαφορετικές προσεγγίσεις στην τροποποίηση της αρχιτεκτονικής:

Τροποποίηση 1η

Η αρχιτεκτονική διατηρεί τη θεμελιώδη δομή του Residual Shuffle Exchange Network που αφορά την υλοποίηση του MusicNet Dataset, αλλά εισάγει έναν διακλαδωμένο δρόμο μετά το επίπεδο γραμμικού μετασχηματισμού. Κάθε δρομολόγιο αφορά ένα από τα δύο όργανα, ενσωματώνοντας έτσι ένα ξεχωριστό Residual Shuffle Exchange Network για κάθε μια απο τις δύο κιθάρες. Το τελικό στάδιο σε κάθε δρομολόγιο αποτελείται από ένα επίπεδο συνέλιξης που είναι ειδικά ρυθμισμένο για να συμπιέσει τα χαρακτηριστικά σε ένα 128-διάστατο αποτέλεσμα. Αυτή η διάσταση αντιστοιχεί στις 128 πιθανές νότες MIDI, επιτρέποντας στο μοντέλο να αποτυπώσει το πλήρες φάσμα των νοτών που κάθε όργανο μπορεί να παράγει.

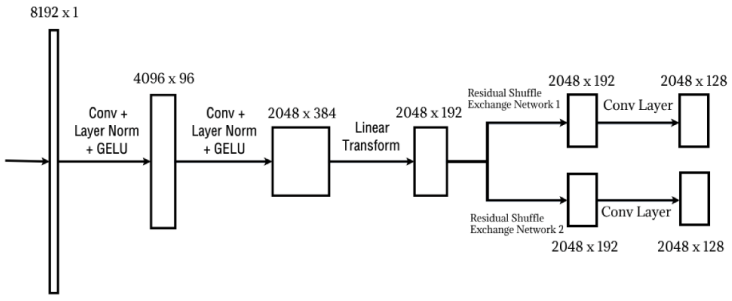


Figure 0.0.20: Τροποποίηση 1η.

Τροποποίηση 2

Στη δεύτερη τροποποίηση, υλοποιήσαμε μια πιο απλή προσαρμογή του αρχικού Residual Shuffle Exchange Network. Ο πυρήνας του δικτύου, συμπεριλαμβανομένου του στοιχείου Residual Shuffle Exchange, παραμένει αναλλοίωτος για να διατηρηθεί η ακεραιότητα της εξαγωγής χαρακτηριστικών που πραγματοποιείται από τον αρχικό σχεδιασμό. Ωστόσο, η αλλαγή έχει γίνει στο τελευταίο στάδιο του μοντέλου, όπου το τελευταίο επίπεδο συνέλιξης έχει αναδιαμορφωθεί έτσι ώστε να εξάγει διάνυσμα διπλάσια σε διάσταση απο το αρχικό. Αυτό το επίπεδο συνέλιξης είναι ικανό να αναθέσει τα εξαγμένα χαρακτηριστικά στο κατάλληλο όργανο, εξάγοντας μια αναπαράσταση 2x128 που ενσωματώνει την κατανομή πιθανοτήτων για όλες τις 128 νότες MIDI σε και τα δύο όργανα.

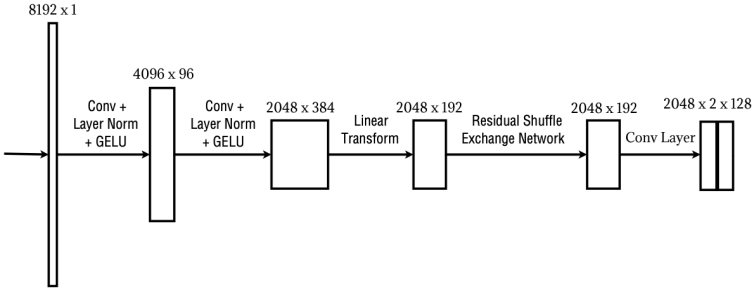


Figure 0.0.21: Τροποποίηση 2

Πειράματα και Αποτελέσματα

Τα μοντέλα εκπαιδεύτηκαν με μέγεθος πακέτου 4, σε παράθυρο 8192 δειγμάτων με βήμα 128. Η εκπαίδευση χρησιμοποίησε δύο σύνολα δεδομένων: το GuitarSet και το σύνολο δεδομένων με τα εικονικά όργανα που δημιουργήσαμε.

Εκπαίδευση στο NI Dataset

Table 3: Μελέτη προσθήκης Permutation Invariant Training

APS SCORE	GuitarSet	NI Dataset Test
Modification 1 no PIT	10.06%	59.03%
Modification 1 PIT	13.14%	62.99%
Modification 2 no PIT	12.37%	60.58%
Modification 2 PIT	13.29%	63.97%

Το αρχικό πείραμα είχε ως στόχο τη σύγκριση της απόδοσης και της αποτελεσματικότητας των δύο τροποποιήσεων. Ενώ και οι δύο τροποποιήσεις είχαν παρόμοια ακρίβεια, η δεύτερη τροποποίηση είχε ελαφρώς καλύτερη απόδοση όπως φαίνεται και απο τον πίνακα 0.0.10. Είναι σημαντικό να σημειωθεί ότι υπερέβη την πρώτη αρχιτεκτονική στην ακρίβεια, με υποδιπλάσιο αριθμό παραμέτρων. Δεδομένου αυτού του πειράματος απο εδώ και στο εξής θα χρησιμοποιούμε την 2η τροποποίηση για όλα τα υπόλοιπα πειράματα.

Εκπαίδευση με διαφορετικά σύνολα δεδομένων

Table 4: Αποτελέσματα στα διάφορα σύνολα δεδομένων

MOD2 - APS SCORE	GuitarSet	NI Dataset Test
Train NI - Finetune GuitarSet	71.81%	7.99%
Train GuitarSet	72.77%	7.65%
Train GuitarSet + NI Dataset	68.62%	62.62%

Το πρώτο πείραμα αποκάλυψε πως το μοντέλο ξεχνάει γρήγορα τα δεδόμενα στα οποία έχει προεκπαιδευτεί καθώς έχοντας εκπαιδευτεί σε δέυτερο χρόνο στο σύνολο NI Dataset ώντας προεκπαιδευμένο στο GuitarSet, η απόδοση του μοντέλου στο σύνολο δοκιμής του GuitarSet ήταν μη βέλτιστη, υποδηλώνοντας απώλεια πληροφοριών που αποκτήθηκαν κατά την αρχική εκπαίδευση.

Παρόμοια, το δεύτερο πείραμα υπογράμμισε τις προκλήσεις της μεταφερσιμότητας του συνόλου δεδομένων. Η εκπαίδευση σε πραγματικές ηχογραφήσεις (GuitarSet) δεν μεταφράστηκε καλά στην απόδοση στο συνθετικό σύνολο δεδομένων (NI Dataset), υποδεικνύοντας ένα σημαντικό χάσμα μεταξύ της μάθησης του μοντέλου σε πραγματικά έναντι συνθετικών δεδομένων.

Στο τελευταίο πείραμα, ο συνδυασμός του συνόλου δεδομένων NI Dataset και του GuitarSet για εκπαίδευση οδήγησε σε ένα μοντέλο με μετρικές απόδοσης κοντά στις υψηλότερες βαθμολογίες APS που επιτεύχθηκαν όταν εκπαιδευόταν αποκλειστικά σε κάθε σύνολο δεδομένων. Αυτός ο συνδυασμός δεδομένων ενίσχυσε την ανθεκτικότητα του μοντέλου και την ικανότητά του να γενικεύεται σε ποικίλα δεδομένα. Προχωρώντας, η αρχιτεκτονική από τη δεύτερη τροποποίηση, εκπαιδευμένη στο συνδυασμένο σύνολο δεδομένων, θα χρησιμοποιηθεί για την εκπαίδευση του μοντέλου διαχωρισμού που πληροφορείται από το σκορ.

DEMUCS

Υβριδικός Transformer Demucs για τον Διαχωρισμό Πηγών Μουσικής

Ο Défossez και άλλοι [32] παρουσίασαν τον Υβριδικό Transformer Demucs (HT Demucs), ο οποίος βασίστηκε σε προηγούμενες εκδοχές της ίδιας αρχιτεκτονικής. Η νέα αρχιτεκτονική ενσωματώνει έναν διακλαδούμενο Transformer Encoder, ο οποίος καλείται να συνενώσει και να αλληλοεξαρτήσει πληροφορίες απο 2 τομείς αυτόν της κυματομορφής και αυτόν του φασματογραφήματος.

Το Hybrid Demucs αποτελείται από δύο U-Nets που λειτουργούν τόσο στον χρόνο όσο και στον φάσμα συχνοτήτων, με κάθε ένα να διαθέτει πέντε επίπεδα κωδικοποιητή και αποκωδικοποιητή. Τα επίπεδα συγκλίνουν μετά τον πέμπτο κωδικοποιητή, ακολουθούμενα από ένα κοινό έκτο επίπεδο. Το κύριο επίπεδο αποκωδικοποιητή είναι επίσης κοινό. Η φασματική έξοδος, μετά από αντίστροφο μετασχηματισμό Fourier μικρού χρόνου (iSTFT), συγχωνεύεται με τη χρονική έξοδο, παράγοντας την πρόβλεψη του μοντέλου. Όπως φαίνεται στο σχήμα 0.0.22, η αρχιτεκτονική παρουσιάζει ένα μόνο επίπεδο self-attention κωδικοποιητή από τον Transformer. Ο μηχανισμός προσοχής αποτελείται από 8 "κεφάλια".

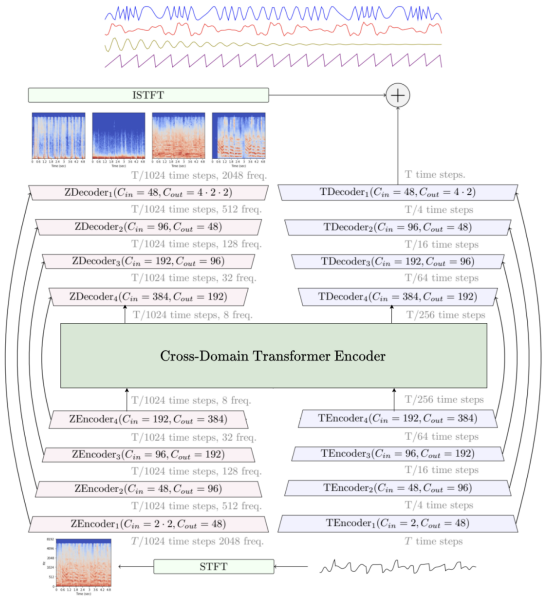


Figure 0.0.22: Αρχιτεκτονική DEMUCS [33].

Αναλυτική Σύγκριση των Μετρικών SDR και SI-SDR

Η αξιολόγηση της ποιότητας της απομόνωσης ενός αλγορίθμου πολλές φορές μετριέται με χρήση μετρικών όπως το SDR και το SI-SDR. Ενώ αυτές οι μετρικές έχουν χρησιμοποιηθεί εκτενώς σε μελέτες που επικεντρώνονται στην απομόνωση διαφορετικών οργάνων, η συμπεριφορά τους στο πλαίσιο της απομόνωσης πηγών με παρόμοια χαρακτηριστικά χροιάς παραμένει λιγότερο διερευνημένη. Δεδομένου ότι η πλειονότητα των προηγούμενων ερευνών περιλαμβάνει όργανα με διαφορετικές χροιές, η άμεση σύγκριση των τιμών SDR που επιτυγχάνουμε εμείς ενδέχεται να μην είναι ενδεικτική. Για να έχουμε μια αρχική εκτίμηση της καταλληλότητας των μετρικών αυτών στα πειράματα μας, δημιουργήσαμε 2 διαφορετικά σενάρια μίξεων για να συγκρίνουμε. Το πρώτο σενάριο αφορούσε 2 ηχητικά σήματα απο 2 διαφορετικές κλασικές κιθάρες ενώ το δεύτερο σενάριο αφορούσε 1 ηχητικό σήμα απο κλασική κιθάρα και 1 ηχητικό σήμα απο πιάνο. Σε κάθε σενάριο παραλλάξαμε συστηματικά την αναλογία ανάμειξης για να προσομοιώσουμε διάφορα επίπεδα απομόνωσης πηγών. Με το πείραμα αυτό στοχεύουμε στο να εξετάσουμε πιθανές ανισότητες στις απαντήσεις των μετρικών. Η μεθοδολογία που ακολουθούμε φαίνεται στο παρκάτω σχήμα 0.0.23.

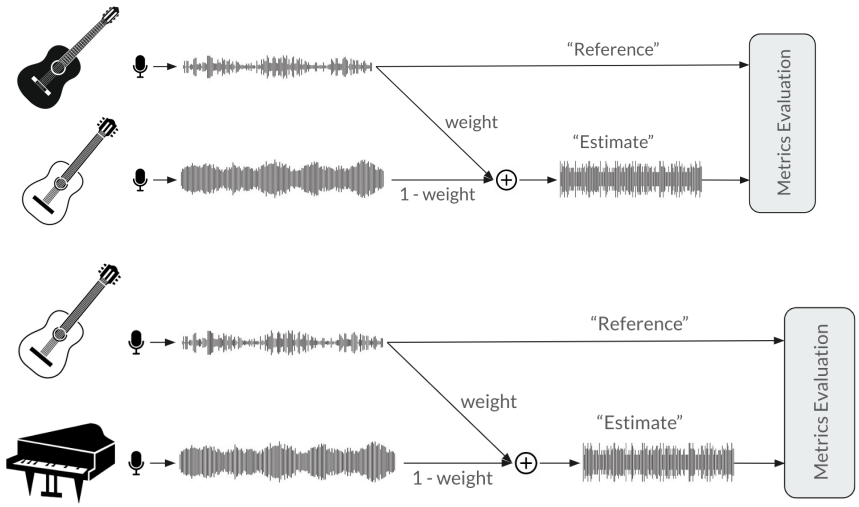


Figure 0.0.23: Μεθοδολογία για αξιολόγηση μετρικών.

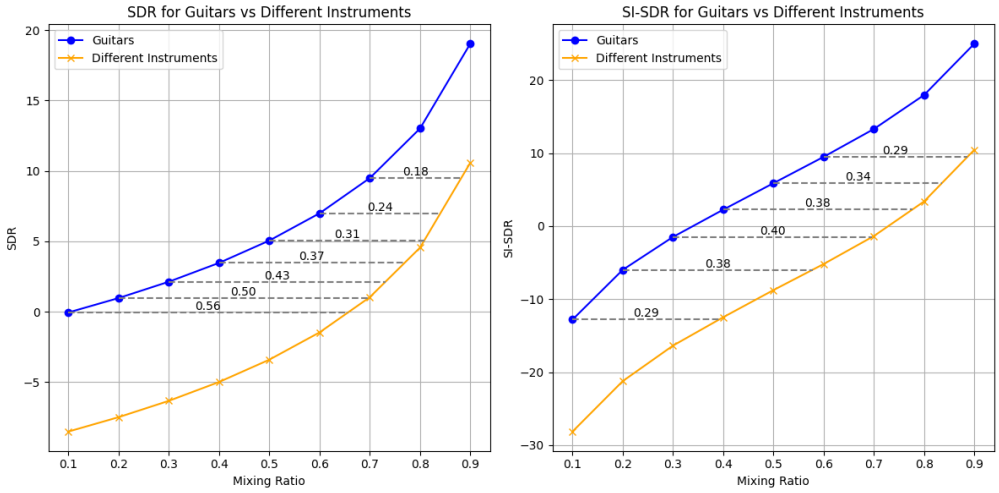


Figure 0.0.24: Ανάλυση των μετρικών SDR και SI-SDR για μίξεις μεταξύ δύο κλασικών κιθάρων και διαφορετικών οργάνων.

Τα αποτελέσματα, όπως φαίνεται στο σχήμα 0.0.24 δείχνουν ότι οι τιμές SDR και SI-SDR για τα μίξεις κι-

θάρας είναι συνεχώς υψηλότερες από αυτές που προέρχονται από μίξεις διαφορετικών οργάνων. Συγκεκριμένα, παρατηρήθηκαν ίδιες τιμές SDR στα μείγματα κιθάρας ακόμη και όταν μια κιθάρα είχε αρκετά χαμηλότερη ένταση σε σύγκριση με τα διαφορετικά όργανα. Αυτό υποδηλώνει ότι η ομοιότητα στη χροιά μεταξύ των δύο κιθάρων αποτελεί πρόκληση για τις μετρικές για να αξιολογήσουν ακριβώς την ποιότητα της απομόνωσης.

Υλοποίηση Συνάρτησης Ενίσχυσης Δεδομένων Opposite Panning

Η υλοποίηση της τεχνικής επαύξησης OppositePanning έγκειται στην ευρεία χρήση του panning στις σύγχρονες ηχογραφήσεις. Με τον όρο panning στην ηχοληψία αναφερόμαστε στη σκόπιμη τοποθέτηση ενός ήχου σε ένα απο τα δύο κανάλια ενός στερεοφωνικό αρχείου. Σε παραγωγές όπου υπάρχουν πολλά μουσικά όργανα, το panning είναι μια κοινή τεχνική για τη δημιουργία χώρου και μιας πληρέστερης εικόνας ήχου. Αυτό είναι ιδιαίτερα σημαντικό όταν πρόκειται για τον διαχωρισμό οργάνων όπως οι κλασικές κιθάρες, οι οποίες συχνά τοποθετούνται αντιδιαμετρικά σε μια μίξη. Στο παρακάτω σχήμα 0.0.25 φαίνεται μια απεικόνιση της μεθοδολογίας μας.

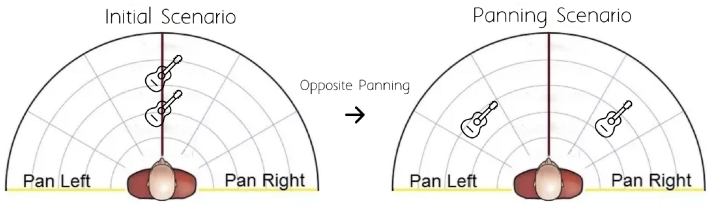


Figure 0.0.25: Τεχνική Επαύξησης Opposite Panning.

Στο σύνολο δεδομένων μας, οι αρχικές ηχογραφήσεις δεν περιλαμβάνουν αυτήν την μεθοδολογία panning, παρουσιάζοντας ενα σενάριο που δεν αντιπροσωπεύει πλήρως τις συνθήκες του πραγματικού κόσμου. Για να αντιμετωπίσουμε αυτό, η τεχνική OppositePanning εισάγεται ως στρατηγική επαύξησης. Αυτή η τεχνική δημιουργεί τεχνητά σενάρια όπου δύο κιθάρες έχουν διαφορετικά panning, προσομοιώντας εγγενώς τις συνθήκες πραγματικής ηχογράφησης.

Πειράματα και αποτελέσματα

Στο πλαίσιο των πειραμάτων μας, τροποποιήσαμε το μοντέλο έτσι ώστε να εξάγει μόνο δύο στερεοκανάλια, με κάθε κανάλι να αντιπροσωπεύει μια από τις δύο κλασικές κιθάρες. Ακόμα ρυθμίσαμε το μοντέλο για να λειτουργεί σε τμήματα 4 δευτερολέπτων. Σε συμφωνία με την ρυθμίσεις των αρχικών ηχογραφήσεων, η συχνότητα δειγματοληψίας για τις στερεοφωνικές κυματομορφές διατηρήθηκε στα 44.100 Hz. Εφαρμόσαμε την εκπαίδευση PIT για να λάβουμε υπόψη τις πιθανές μεταθέσεις πηγών κατά τη διάρκεια της εκπαίδευσης.

Κατά τη διάρκεια της εκπαίδευσης, η συνάρτηση λάθους ήταν ένας συνδυασμός του L1 σφάλματος (βάρος 0,8) και του αθροιστικού σφάλματος (βάρος 0,2). Τα βάρη καθορίστηκαν με βάση τα βέλτιστα αποτελέσματα από διάφορες δοκιμές. Η ενσωμάτωση του αθροιστικού σφάλματος εξασφαλίζει ότι το μοντέλο αναγνωρίζει ότι και τα δύο αποτελέσματα θα πρέπει συλλογικά να προσεγγίζουν την είσοδο. Χρησιμοποιήθηκε ο βελτιστοποιητής Adam, με ρυθμό μάθησης ορισμένο στο 0,0003, συμβατό με το αρχικό Demucs. Η εκπαίδευση του μοντέλου κράτησε περίπου 100 εποχές. Είναι σημαντικό να σημειωθεί ότι τα πιο αποτελεσματικά μοντέλα προέκυψαν συνεπώς μεταξύ της 70ης και 100ης εποχής. Τα δεδομένα μας χωρίστηκαν με αναλογία 80-20 εκπαίδευσης-επικύρωσης αντίστοιχα. Για την αξιολόγηση της απόδοσης, χρησιμοποιήσαμε μετρικές όπως οι SDR, SI-SDR, SAR, ISR και SIR.

Τα πειράματα που τρέξαμε χωρίζονται σε 2 βασικές κατηγορίες. Η πρώτη κατηγορία αφορά τα πειράματα τα οποία έγιναν στην αρχιτεκτονική του DEMUCS χωρίς την αξιοποίηση κάποιας περαιτέρω πληροφορίας για τις νότες που παίζει το κάθε όργανο ενώ η δεύτερη κατηγορία αφορά τα πειράματα στα οποία τροποποιήσαμε την αρχιτεκτονική του DEMUCS με στόχο να μπορεί να δέχεται δεδομένα για τις νότες που αφορούν την κάθε κιθάρα. Για την πληροφορία των νοτών χρησιμοποιήσαμε είτε δεδομένα groundtruth είτε δεδομένα τα οποία είχαν προβλεφθεί απο την αρχιτεκτονική RSE που παρουσιάστηκε προηγουμένως. Παρακάτω φαίνεται μια γενική επισκόπηση της διαδικασίας που χρησιμοποιήθηκε για να εκπαιδευτεί και να αξιολογηθεί το μοντέλο. Το παρακάτω σχήμα 0.0.26 περιγράφει τη διαδικασία εκπαίδευσης του μοντέλου.

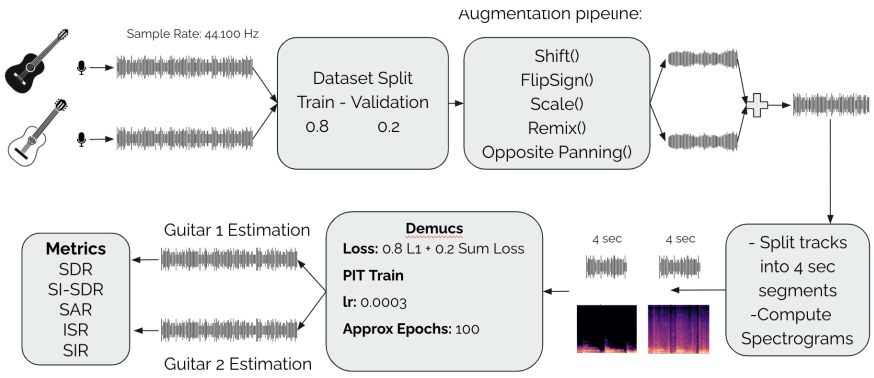


Figure 0.0.26: Διαδικασία εκπαίδευσης και αξιολόγησης του μοντέλου.

Τα τελικά αποτελέσματα απο όλα τα πειράματα φαίνονται στον παρακάτω πίνακα όπου χρησιμοποιούμε τις ακρωνυμίες.

GD GuitarDuets Dataset

GS GuitarSet Dataset

NI NI Dataset

GS SI GuitarDuets Score Informed

Table 5: Συνολικά αποτελέσματα σε όλα τα σύνολα δεδομένων

Metric	URMP	GD+GS+NI	GD + GS	GS	NI	GD	GD SI	GD NI
SDR	G1: 1.633 G2: 2.609	G1: 4.306 G2: 0.765	G1: 4.200 G2: 1.206	G1: 4.580 G2: 1.323	G1: 2.672 G2: 0.012	G1: 5.140 G2: 0.991	G1: 5.399 G2: 1.166	G1: 5.988 G2: 0.934
SI-SDR	G1: -2.440 G2: -2.444	G1: -6.161 G2: -6.174	G1: -0.334 G2: -0.340	G1: -4.223 G2: -4.178	G1: -0.841 G2: -0.815	G1: 1.803 G2: 1.806	G1: 1.657 G2: 1.664	G1: 2.370 G2: 2.362
SAR	G1: 2.906 G2: 6.088	G1: 7.988 G2: 1.964	G1: 8.839 G2: 10.670	G1: 8.314 G2: 6.627	G1: 4.706 G2: 8.491	G1: 7.719 G2: 1.417	G1: 7.815 G2: 2.101	G1: 8.835 G2: 0.893
SIR	G1: 5.113 G2: 8.250	G1: 10.596 G2: 4.518	G1: 6.235 G2: 7.732	G1: 7.552 G2: 8.939	G1: 3.989 G2: 8.612	G1: 10.186 G2: 4.935	G1: 11.280 G2: 5.158	G1: 11.777 G2: 4.271
ISR	G1: 1.621 G2: 2.817	G1: 5.873 G2: 1.146	G1: 5.908 G2: -1.427	G1: 8.115 G2: -0.310	G1: 6.368 G2: -2.957	G1: 6.400 G2: 0.571	G1: 6.808 G2: 0.589	G1: 7.229 G2: 2.215

Μετά την ανάλυση των αποτελεσμάτων από τα διάφορα σύνολα δεδομένων, προκύπτουν οι εξείς παρατηρήσεις. Πρώτον, αποσκοπούμε στην ανάλυση της βέλτιστης απόδοσης με βάση τη μετρική SDR, ο συνδυασμός του 'My-Dataset+NI Dataset' για την πρώτη κιθάρα (G1) παρήγαγε την υψηλότερη SDR στα 5.988. Αυτό υποδηλώνει ότι το μοντέλο που εκπαιδεύτηκε σε αυτό το συνδυασμένο σύνολο δεδομένων ήταν πιο αποτελεσματικό στη συνολική ποιότητα διαχωρισμού σήματος για την κιθάρα που παίζει σόλο. Η προσθήκη του NI Dataset, που δημιουργείται από ένα πρόσθετο εικονικό όργανο κιθάρας, πιθανόν παρείχε επιπλέον πληροφορίες που βελτίωσαν τη δυνατότητα του μοντέλου να διακρίνει ανάμεσα στις δύο κιθάρες. Για τη μετρική SI-SDR, ορισμένα από τα σύνολα δεδομένων εμφάνισαν αρνητικές τιμές SI-SDR, υποδεικνύοντας προκλήσεις στην επίτευξη διαχωρισμού λαμβάνοντας υπόψιν την ένταση του κάθε σήματος, ενώ άλλα είχαν χαμηλές θετικές τιμές. Ωστόσο, και πάλι το σύνολο δεδομένων "GuitarDuets + NI Dataset" εμφάνισε τα λιγότερο ικανοποιητικά αποτελέσματα. Για αυτό το συνδυασμό, η κοντινότητα των τιμών SDR στις τιμές SI-SDR υποδεικνύει ότι το μοντέλο διατηρεί επαρκώς την ένταση των αρχικών πηγών, εξασφαλίζοντας συνεπή ποιότητα διαχωρισμού.Παρατηρείται μια έντονη ασυμμετρία μεταξύ των μετρικών SDR και SI-SDR, με την τελευταία να καταγράφει συχνά χαμηλότερες τιμές. Αυτό είναι ένα αναμενόμενο αποτέλεσμα, δεδομένης της ανάλυσης που κάναμε παραπάνω. Αυτή η απόκλιση υπογραμμίζει την παρουσία σφαλμάτων διαχωρισμού που υπερβαίνουν απλώς την κλιμάκωση της έντασης, πιθανόν περιλαμβάνοντας παραμορφώσεις, παρεμβολές ή άλλα artifacts.

Συνολικά, το καλύτερο μοντέλο είναι αυτό που εκπαιδεύτηκε στο συνδυασμό "GuitarDuets + NI Dataset" επιτυγχάνοντας συνεπή αποτελέσματα σε όλες τις μετρικές.

Η φύση κάθε συνόλου δεδομένων παίζει καθοριστικό ρόλο στην απόδοση του μοντέλου. Για παράδειγμα, η καθαρή διάκριση μεταξύ συνοδείας και μελωδίας στο 'GuitarSet' απλοποιεί τον καθορισμό διαχωρισμού. Αντιθέτως, το 'myDataset', που απο τη φύση του παρουσιάζει σύνθετους ρόλους της κάθε κιθάρας, οδηγεί σε σχετικά χαμηλές μετρικές. Το 'NI Dataset', που προέρχεται από ένα εικονικό όργανο, προσφέρει ένα πιο καθαρό ήχο, το οποίο, όταν συνδυάζεται με πραγματικές ηχογραφήσεις, μπορεί να ενισχύσει την ικανότητα γενίκευσης του μοντέλου.

Συμπεράσματα και Μελλοντικές Επεκτάσεις

Συμπεράσματα

Στην εν λόγω έρευνα μελετήσαμε το πρόβλημα διαχωρσιμού μουσικών πηγών με κοινή χροιά, συγκεκριμένα σε ντουέτα κλασικής κιθάρας. Χρησιμοποιήσαμε για τα πειράματα μας τις state-of-the-art αρχιτεκτονικές για διαχωρισμό μουσικών πηγών για να εξιολογήσουμε την επίδοση τους στο δικό μας πρόβλημα. Για το σκοπό αυτό, δημιουργήσαμε δύο νέα σύνολα δεδομένων που αποτελούνται από πραγματικές και εικονικές ηχογραφήσεις ντουέτων κιθάρας, καθώς δεν υπήρχε κάποιο άλλο σύνολο δεδομένων διαθέσμο. Εισήγαμε μια νέα τεχνική επαύξησης δεδομένων, την OppositePanning. Ταυτόχρονα προτείναμε μια αρχιτεκτονική σειρά η οποία αξιοποιεί 2 διαφορετικά μοντέλα μουσικής μεταγραφής και διαχωρισμού πηγών για τον καλύτερο διαχωρισμό. Απο τα πειράματα μας παρατηρήσαμε πως τα μοντέλα που επιτυγχάνουν πολύ καλό διαχωρισμό στην περίπτωση που οι μουσικές πηγές παρουσιάζουν διαφορετικές χροιές, δεν καταφέρνουν να έχουν ίδιο αποτέλεσμα για τα ντουέτα κλασικής κιθάρας. Παρατηρήθηκε μάλιστα οτι παρά τις υψηλές τιμές των μετρικών για την αξιολόγηση του μοντέλου, η ποιότητα του διαχωρσιμού όπως διαπιστώθηκε απο ατομικό ακουστικό τέστ δεν ήταν ανάλογη. Έτσι λοιπόν κάνοντας συγκριτικά πειράματα, είδαμε πως οι μετρικές που χρησιμοποιούνται μέχρι στιγμής για τον διαχωρισμό πηγών με διαφορετικές χροιές δεν είναι τόσο αντιπροσωπευτικές για τον διαχωρισμό πηγών με κοινή χροιά. Ακόμα προτείναμε μια νεα αρχιτεκτονική η οποία παραθέτουμε ένα μοντέλο μουσικής μεταγραφής στη σειρά με ένα μοντέλο διαχωρσιμού. Η εν λόγω αρχιτεκτονική φάνηκε να δίνει ένα σημαντική αύξηση στις τιμές κάποιων απο των μετρικών.

Μελλοντικές Επεκτάσεις

- Κρίνεται σημαντικό για τις μελλοντικές επεκτάσεις να δωθεί περισσότερη βάση στη δημιουργία ενός πληρέστερου σε διάρκεια και σε διαφορετικές χροιές κιθάρας. Με αυτόν τον τρόπο θα μπορέσουν τα μοντέλα να γενικέυουν καλύτερα και να εκπαιδεύονται σε ένα πληρέστερο σύνολο δεδομένων. Ακόμα είναι αναγκαίο να παρατεθούν μαζί με τα αρχεία ήχου της κάθε κιθάρας, αρχεία που περιγράφουν τις νότες που παίζει η κάθε μια, έτσι ώστε να είναι ευκολότερη η εκπαίδευση αρχιτεκτονικών που ασχολούνται με την μουσική μεταγραφή.
- Εξισού σημαντικό είναι να γίνει ένα τεστ ακροάσεων απο εκπαιδευμένους μουσικούς, έτσι ώστε να αξιολογήσουμε ποιοτικά την απόδοση του διαχωρισμού αλλά και ταυτόχρονα να αντιστοιχίσουμε τις τιμές των μετρικών με την ποιοτική βαθμολόγηση των ακροατών
- Ακόμα, είναι απαραίτητο να γίνει μια εκτενέστερη μελέτη της απόδοσης των μετρικών στο κατα πόσο είναι αντιπροσωπευτικές του διαχωρισμού μουσικών πηγών οι οποίες παρουσιάζουν κοινά χαρακτηριστικά χροιάς.

Chapter 1

Introduction

Contents

Music source separation is an essential task in audio signal processing that aims to separate individual sound sources from a mixed audio recording without prior knowledge on the properties of the participating signals. The ability to isolate and extract individual instruments or voices from a musical ensemble has numerous applications in music production, audio restoration, and content analysis [34]. Classical guitar duets, characterized by the intricate interplay and harmonization between two guitarists, present a unique and challenging scenario for music source separation.

The motivation behind this thesis originates from the extensive and diverse repertoire of classical guitar duets, where the interaction between two guitars creates a rich sound of harmonies and timbres. There are many situations where the ability to separate the individual guitar parts could be valuable. For instance, in educational settings, learners may benefit from isolating and studying each guitarist's performance independently. Furthermore, in music production and performance, the ability to separate the guitar parts can provide greater control over the mix, enabling adjustments of volume levels, equalization, and spatial placement as it has been shown in [35, 36, 37].

Moreover, the practical implications of successful source separation in this context are vast. In the domain of audio forensics, separating guitar tracks can aid in the analysis and authentication of recordings. This capability could be instrumental in copyright disputes where the origin of a specific guitar part is in question. Additionally, the extracted audio can serve as a valuable resource for remixing and remastering historical recordings, where original multitracks are unavailable, thereby preserving and revitalizing cultural heritage.

The significance of this research also extends to the development of assistive technologies for musicians and composers. By facilitating the isolation of individual parts, composers can experiment with rearranging pieces for educational purposes or adaptive performances, catering to musicians with varying skill levels or disabilities. Similarly, assistive listening devices can be enhanced to focus on specific instruments within an ensemble, tailoring the listening experience to the preferences or needs of the user, such as focusing on a particular guitar part in a duet for learning or enjoyment purposes.

1.1 Definition of Problem: Music Source Separation

Source separation is a common problem in the field of digital signal processing (DSP) and has attracted significant attention in AI research. It involves decoupling different source signals within a given signal mixture, aiming to eliminate unwanted interferences or isolate specific source signals for further processing. Source separation can be applied to various types of signals, including images and audio signals.

While source separation shares some similarities with signal denoising, as both involve separating signals of interest from unwanted components, source separation primarily focuses on isolating "proper" signals rather than noise. There exist effective techniques for generic denoising problems, but source separation tackles the challenge of extracting specific source signals from mixtures.

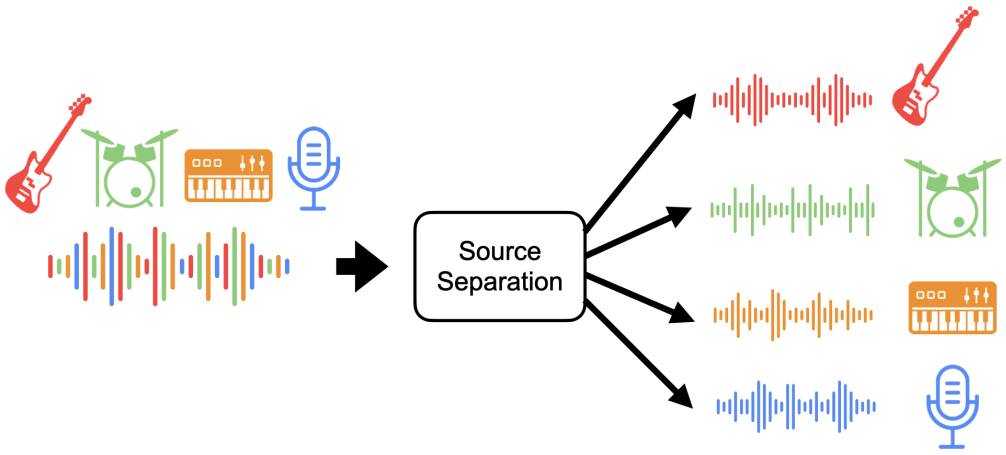


Figure 1.1.1: Source Separation image [1]

1.2 Monotimbral Source Separation in Classical Guitar Duets: Challenges and Complexities

This thesis specifically addresses the task of source separation in the context of classical guitar duets, a type of monotimbral source separation problem. The goal is to separate the individual guitar parts from a mixed recording without any prior information about the sources, such as musical arrangements. This problem falls under the broader category of music source separation, which involves separating different instruments or vocals in a musical ensemble. It is also related to speech separation, which aims to isolate speech from a mixture in multi-speaker environments.

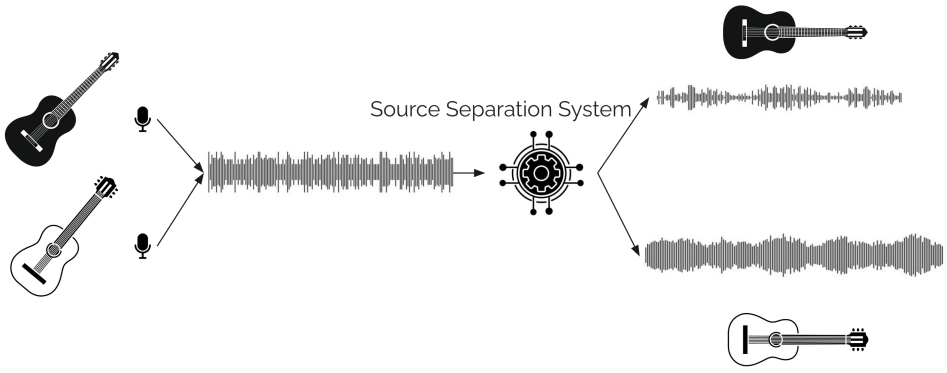


Figure 1.2.1: Source Separation in Classical Guitar Duets

Before delving into tmonotimbral source separation, it is essential to understand the foundational role of timbre in music. Timbre, often described as the "color" or "quality" of a musical sound, distinguishes different types of sounds, even when they share the same pitch and loudness. At a high level, timbre is what enables our ears to identify instruments, voices, or sounds as unique, such as differentiating a piano from a violin even if they play the same note at the same volume. This unique quality arises from the complex interplay of sound waves, including the fundamental frequency and a series of overtones or harmonics that the instrument or voice produces. The specific mixture of these harmonics, along with the way they evolve over time, contribute to the distinctive timbre of a sound. The material and shape of an instrument, the technique of a player, or the characteristics of an electronic sound source can all significantly affect these acoustic properties, crafting the sounds we can recognize and appreciate in music.

Difference between Classic Source Separation and Monotimbral Source Separation

In classic source separation, the focus is on separating individual sources with distinct timbral characteristics, such as vocals, drums, and bass. These sources often exhibit significant spectral and temporal differences, making them easier to distinguish. In contrast, monotimbral source separation aims to separate sources that belong to the same instrumental family or share similar timbral characteristics. In the case of classical guitar duets, both guitars produce sounds with similar timbres, making it challenging to separate the individual guitar parts solely based on spectral differences.

Polyphonic Nature of the Guitar

Another significant challenge in monotimbral source separation of classical guitar duets is the polyphonic nature of the guitar itself. Unlike monophonic instruments, such as a single voice or a solo guitar, the interaction between two guitars in a duet creates complex polyphonic textures, where multiple notes are played simultaneously. This polyphony results in overlapping and intertwined frequency components, making it difficult to isolate individual guitar parts based solely on spectral cues.

Intertwined Melodies and Harmonies

Classical guitar duets are known for their intricate interplay and harmonization between the two guitarists.

The melodies and harmonies produced by each guitarist are often closely intertwined, with notes from one guitar complementing or harmonizing with the other. This interdependence further complicates the separation process, as the individual guitar parts are not only spectrally similar but also musically intertwined. It requires sophisticated algorithms and models that can effectively capture the intricate relationships and dependencies between the guitar parts.

Performance Variability and Expressiveness

The performance style and expressiveness of the guitarists add another layer of complexity to the monotimbral source separation task. Each guitarist may have a unique playing technique, dynamic range, and expressiveness, which further affects the spectral characteristics of the individual guitar parts. Additionally, the performance variability, including timing fluctuations, nuances, and stylistic variations, introduces further challenges in accurately separating the guitar parts.

Addressing these challenges in monotimbral source separation of classical guitar duets requires advanced algorithms and models that can exploit both spectral and temporal cues effectively. Techniques that incorporate higher-level musical knowledge, such as score information or harmonic models, can provide additional context and improve the separation quality.

1.3 Goals and Contributions

1.3.1 Goals

The primary objectives of this thesis are:

- To explore and develop effective techniques for monotimbral music source separation of classical guitar duets.
- To contribute to the broader field of music source separation by expanding the understanding of the challenges and opportunities specific to classical guitar duets.
- To conduct subjective and objective evaluations to assess the quality and perceptual fidelity of the separated guitar parts.

1.3.2 Contributions

This thesis contributes to the field in several ways:

- Introduction of a dataset comprised of real classical guitar duet recordings, totaling approximately one hour of music, which serves as a valuable asset for training and evaluating source separation models.
- Creation of a synthetic dataset using online transcriptions of guitar duets and virtual instruments to generate approximately two hours of music. This dataset includes MIDI representations for each guitar part, providing a rich resource for in-depth analysis and algorithm training.
- Implementation of the *OppositePanning* augmentation technique to simulate real-world recording conditions and improve the model's performance in scenarios with spatial audio variations.
- Comparative analysis of SDR metrics in the context of classical guitar duets to understand their effectiveness and limitations for sources with similar timbral characteristics.
- Development of a pipeline composed of dual models for improved separation accuracy and fidelity. This includes generating a piano roll representation for each guitar, followed by separating the mixed audio into individual guitar audio files.
- Exploration and assessment of modifications to a music transcription architecture, shifting from outputting a single transcript to generating two separate transcripts for each instrument in the recordings. This adaptation aimed to enhance the precision of the transcription process in scenarios involving two instruments.
- Modification of the Demucs hybrid transformer architecture to incorporate activity labels (soft labels) for the notes. This alteration was designed to provide a more nuanced understanding of note activity,

potentially leading to improved separation performance by accurately capturing the dynamics of note production.

1.4 Thesis Outline

In the subsequent sections of this thesis, we will present a comprehensive review of the relevant literature, describe the methodology employed in our research, detail the implementation and experimental setup, discuss the obtained results, and provide a thorough analysis and interpretation of the findings. Finally, we will conclude with an assessment of the contributions made by this study and outline potential avenues for future research.

This thesis is organized into five chapters. Each chapter is outlined as follows:

- **Chapter 1: Introduction**
  This chapter provides an introduction to the problem of music source separation, particularly focusing on classical guitar duets, the specific challenges they present, the goals, and the contributions of this research.
- **Chapter 2: Theoretical Background**
  A comprehensive overview of the machine learning methodologies and audio representations pertinent to music source separation, including advanced neural network architectures.
- **Chapter 3: Literature Review**
  An analysis of the current state-of-the-art approaches in music source separation, discussing both digital signal processing and deep neural network approaches.
- **Chapter 4:Creating Datasets for Monotimbral Source Separation**
  An analysis of the methodology and an overview of the two distinct datasets created.
- **Chapter 5: Residual Shuffle-Exchange Music Transcription Network And Experiments**
  Introducing and analyzing the Residual Shuffle-Exchange Music Transcription Network. Modification of the aforementioned model and experiments on various datasets, including our own datasets.
- **Chapter 6: Demucs Architecture And Experiments**
  An exploration of the Demucs architecture's history and experiments, discussing the rationale behind the OppositePanning augmentation, and detailing the score-informed and non-score informed experiments.

Chapter 2

Theoretical Background

Contents

2.1 Machine Learning

Machine Learning (ML) is a subfield of artificial intelligence that focuses on the development of algorithms and statistical models that enable computers to perform tasks without explicit instructions, relying instead on patterns and inference. The primary goal of ML is to allow computers to learn automatically without human intervention.

2.1.1 Types of Machine Learning

There are three main types of machine learning:

- **Supervised Learning:** This involves learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. Mathematically, given a training set $(x_i, y_i)_{i=1}^n$ where x_i represents the input and y_i the corresponding output, supervised learning algorithms try to find a function f such that $f(x_i) \approx y_i$.
- **Unsupervised Learning:** Unlike supervised learning, unsupervised learning algorithms are given no labels and are left to find structure in their input on their own. The primary objective is to model the underlying structure or distribution in the data in order to learn more about the data, with prominent techniques including clustering and dimensionality reduction.
- **Reinforcement Learning:** This type of learning is concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. The problem is modeled as a Markov Decision Process (MDP) with states S, actions A, and rewards R, aiming to find a policy $\pi : S \to A$ that maximizes the expected cumulative reward.

2.1.2 Fundamental Concepts

Feature Extraction and Selection Features are individual independent variables that act as the system input. Predictive models use features to make predictive decisions. The process of transforming raw data into a set of features is known as feature extraction. The selection of appropriate features in the dataset substantially influences the efficacy of the machine learning model.

Model Evaluation In ML, model evaluation is a critical step to understand the performance of the model. It involves splitting the dataset into training and testing sets, where the training set is used to train the model and the testing set, which is unseen during training, is used to evaluate its performance. The evaluation of model performance is articulated through various metrics, each designed to the specific nature of the task at hand. For classification tasks, metrics such as accuracy, precision, recall, and the F1 score are prevalent, whereas mean squared error (MSE) is typically utilized for regression tasks.

The *accuracy* metric represents the proportion of correct predictions out of the total predictions made, formulated as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{2.1.1}$$

Precision and recall are metrics that offer a more nuanced evaluation of a model's performance, particularly in imbalanced datasets. *Precision*, defined as the ratio of true positive predictions to the total positive predictions made by the model, is given by:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.1.2}$$

Recall, also known as sensitivity, measures the proportion of actual positives correctly identified by the model, calculated as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2.1.3}$$

The *F1 score* harmonizes the balance between precision and recall, providing a single metric to assess the model's accuracy while considering both the false positives and false negatives. It is the harmonic mean of precision and recall:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.1.4}$$

In the assessment of regression models, common losses that are being utilized are the L1 loss, or Mean Absolute Error (MAE), and L2 loss (also known as Mean Squared Error, MSE) are particularly significant.

L1 loss is defined as the mean absolute difference between the actual and predicted values, offering a linear measure of errors. This characteristic makes it less sensitive to outliers in comparison to L2 loss, thus providing a robust evaluation metric in scenarios where outliers are present:

$$\text{L1 Loss} = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i| \tag{2.1.5}$$

Following the introduction of L1 loss, it is essential to discuss L2 loss, which emphasizes the square of the differences between the actual values and the predictions made by the model. *L2 loss*, synonymous with MSE, heavily penalizes larger errors, which accentuates the impact of outliers within the dataset:

$$\text{L2 Loss (MSE)} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2.1.6}$$

Here, Y_i denotes the actual value, whereas $\hat{Y}_i$ represents the predicted value. The sensitivity of L2 loss to error magnitude makes it a crucial tool for identifying and mitigating the influence of outliers in the data.

Model Training Techniques Effective training of machine learning models is crucial for achieving optimal performance. This process can be approached through either *iterative optimization algorithms* or *closed-form solutions*, depending on the nature of the model and the problem being addressed. Iterative algorithms, such as Gradient Descent or its variants (Stochastic Gradient Descent, Mini-batch Gradient Descent), are widely used for models where the solution cannot be analytically computed. These algorithms iteratively update the model parameters θ to minimize the loss function $J(\theta)$, according to the rule:

$$\theta_{next} = \theta_{current} - \alpha\nabla J(\theta_{current}) \tag{2.1.7}$$

where α is the learning rate and $\nabla J(\theta)$ is the gradient of the loss function with respect to the model parameters. This approach is suitable for large datasets and complex models.

On the other hand, closed-form solutions, like the Normal Equation in Linear Regression, provide a direct computation of the optimal model parameters without the need for iteration. For a linear regression model, the optimal parameters θ can be found using:

$$\theta = (X^T X)^{-1} X^T y \tag{2.1.8}$$

where X is a full rank matrix of input features and y is the vector of target values. However, closed-form solutions are not always feasible due to computational complexity or the inability to express the solution in a closed form for many machine learning models. When X is not a full-rank matrix; in such a case you should use a Penrose Pseudo-Inverse implemented via SVD.

In addition to selecting an appropriate optimization algorithm, the use of a validation set plays an important role in monitoring the training process and preventing overfitting, which is explained in Paragraph 2.1.2. The validation set, separate from the training and test sets, is used to evaluate the model during training, allowing for the tuning of hyperparameters and the assessment of the model's generalization ability to unseen

data. By carefully monitoring performance on the validation set, practitioners can make informed decisions about when to stop training to avoid overfitting, thereby striking a balance between bias and variance.

The choice between iterative and closed-form optimization methods, along with the strategic use of a validation set, are essential considerations in the development of machine learning models, ensuring they are both accurate and generalizable.

Overfitting and Underfitting Overfitting occurs when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. Underfitting occurs when a model cannot capture the underlying trend of the data. Both overfitting and underfitting lead to poor predictions on new data.

Bias-Variance Tradeoff The bias-variance tradeoff is an important concept in machine learning that involves balancing the error introduced by the bias with the error introduced by the variance.

- **Bias:** Error due to overly simplistic assumptions in the learning algorithm. High bias can cause an algorithm to miss relevant relations between features and target outputs (underfitting).
- **Variance:** Error due to too much complexity in the learning algorithm. High variance can cause overfitting.

A good model requires finding a balance between these two types of errors.

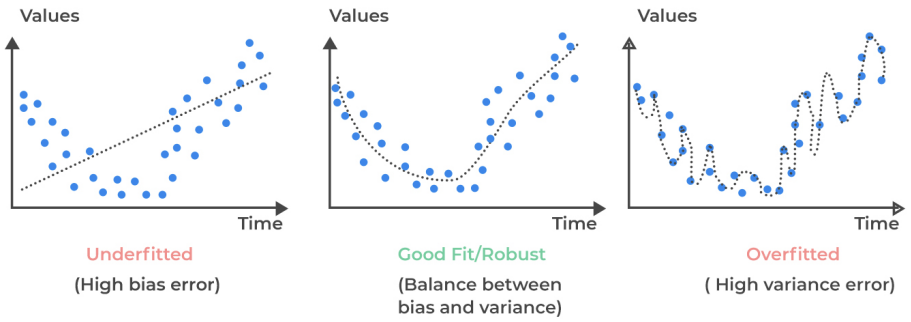


Figure 2.1.1: Bias-Variance tradeoff [38]

2.2 Audio Representations

Audio representation plays a crucial role in the field of music source separation and audio signal processing. Different representations allow us to capture various aspects of audio signals, such as time-domain features, frequency content, and temporal evolution. In this section, we explore several key forms of audio representation.

2.2.1 Waveform Representation

The most basic representation of an audio signal is its waveform, which is a time-domain representation.

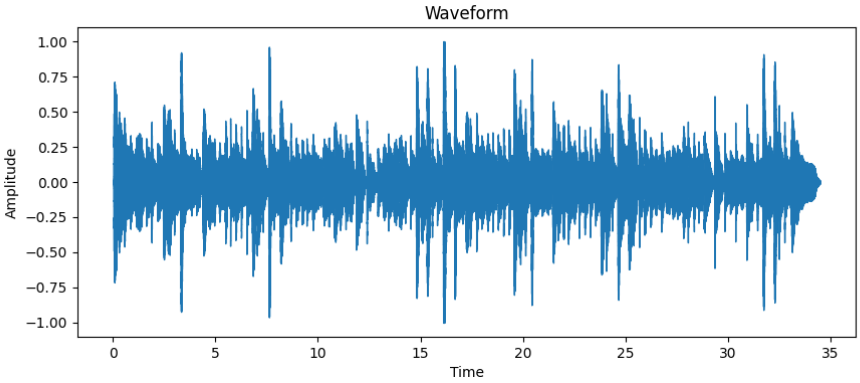


Figure 2.2.1: Waveform of an audio signal.

In waveform representation, the audio signal $s(t)$ is represented as a function of time t. This is the rawest form of the audio, showing the amplitude variations over time. To accurately capture an audio signal using digital samples, the highest frequency in the original sound should be no more than half of the rate at which it's sampled. This rule is known as the Nyquist-Shannon theorem. For example, in many audio applications, sounds are sampled at 44.100 Hz, which means they can represent frequencies up to about 22.050 Hz. During this digitization process, quantization is applied, whereby each sound level is transcribed into a specific number from a predetermined range of values see Fig 2.2.2. This quantization process is crucial for converting the continuous amplitude of the audio signal into a discrete set of levels, which can be represented digitally.

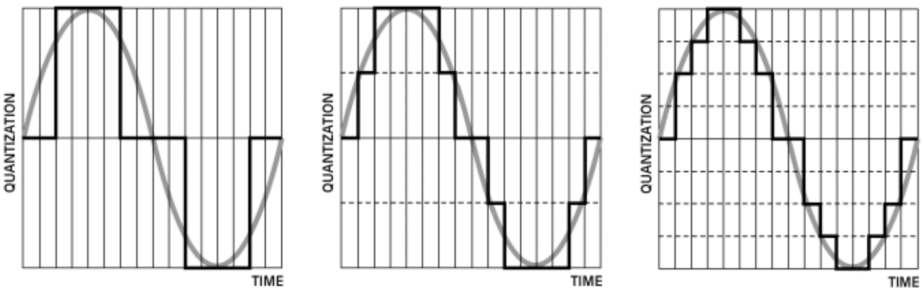


Figure 2.2.2: Quantization illustration. From [2]

2.2.2 Spectral Representations

Spectral representations are crucial in analyzing the frequency content and temporal evolution of an audio signal. These representations, obtained through various Fourier Transform techniques, provide insights into how the sound's characteristics change over time and frequency. As a reminder, the Fourier Transform of a signal is defined as

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft}\, dt \tag{2.2.1}$$

Where:

- $S(f)$ is the Fourier Transform of $s(t)$, representing the signal in the frequency domain.
- $s(t)$ is the original time-domain signal.
- f is the frequency in Hertz (Hz).

- t is the time in seconds.

Spectrogram Representation

The spectrogram is a time-frequency representation that illustrates how the spectral content of a signal changes over time. It is derived using the Short-Time Fourier Transform (STFT), which can be considered as applying the Discrete Fourier Transform (DFT) to consecutive, overlapping segments of the signal. This process can be mathematically represented as:

$$STFT\{x[n]\}(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j2\pi\frac{\omega}{N}n} \tag{2.2.2}$$

In this equation:

- $x[n]$ is the discrete-time representation of the originally continuous time signal $x(t)$, where n represents discrete time indices. The transition from $x(t)$ to $x[n]$ signifies the sampling of the continuous signal into a form suitable for digital processing.

- $w[n-m]$ is the window function applied to localize the signal in time around the index m. This windowing is crucial for analyzing specific segments of the signal while minimizing boundary effects.

- m signifies the discrete time index around which the window function is centered, allowing for the temporal localization of the Fourier analysis.

- ω represents the digital frequency components for which the STFT is calculated, correlating to specific frequencies based on the sampling rate. Unlike analog frequency expressed in Hz, ω is dimensionless, representing a fraction of the sampling frequency.

The STFT equation applies the principles of the Discrete Fourier Transform (DFT) to these windowed segments, enabling the analysis of the signal's frequency content over time. Each term in the sum evaluates the contribution of a frequency component at a specific time, providing a granular view of the signal's spectral evolution.

Most commonly, the magnitude squared of the STFT is visualized in the spectrogram:

$$\text{Spectrogram}(m,\omega) = |STFT\{x[n]\}(m,\omega)|^2 \tag{2.2.3}$$

where $|STFT\{x[n]\}(m,\omega)|^2$ represents the power spectrum of the analyzed segment, highlighting the intensity of various frequencies at each time point m.

The transition from the continuous-time signal $x(t)$ to its discrete-time counterpart $x[n]$, and the subsequent analysis using STFT, underscores the adaptability of Fourier analysis principles in the digital signal processing domain. This approach facilitates a detailed examination of how the frequency components of a signal change over time, making the spectrogram an invaluable tool in audio signal analysis and other applications.

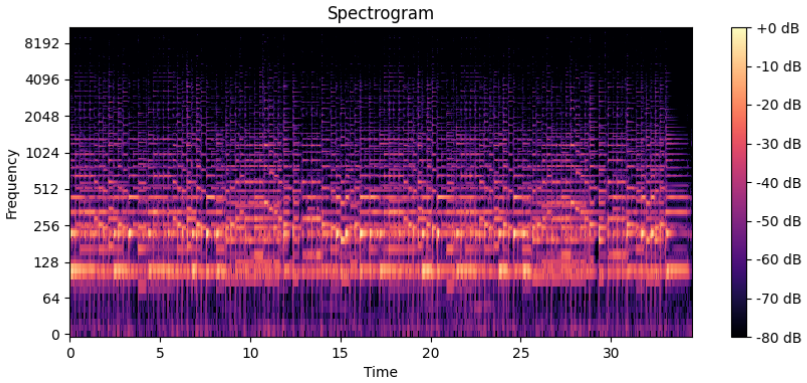


Figure 2.2.3: Spectral representation of an audio signal.

Mel-Frequency Cepstral Coefficients (MFCCs)

Mermerstein et al. [39] introduced MFCCs which are a feature widely used in audio signal processing, particularly in speech and music analysis. They are derived through a multi-step process:

1. **Fourier Transform and Power Spectrum Calculation:** For a digital signal, compute the Discrete Fourier Transform (DFT) to convert it from the time domain to the frequency domain. The DFT is given by:

$$S[k] = \sum_{n=0}^{N-1} s[n] e^{-j2\pi \frac{kn}{N}} \tag{2.2.4}$$

   where N is the total number of samples, $s[n]$ is the signal in the time domain, and k corresponds to the index of the discrete frequency bins. The power spectrum, $P[k]$, is obtained by taking the magnitude squared of the DFT:

$$P[k] = |S[k]|^2 \tag{2.2.5}$$

2. **Frequency Discretization and Mel Scale Mapping:** After computing the DFT and before mapping to the mel scale, acknowledge that the frequencies have been discretized into bins. Then, map the power spectrum obtained from the DFT to the mel scale, which is a perceptual scale of pitches. The mapping can be expressed as follows for discrete frequency bins:

$$m[k] = 2595 \log_{10}\left(1 + \frac{f_k}{700}\right) \tag{2.2.6}$$

   where f_k represents the center frequency of the k-th bin in Hz, which is derived from the discrete frequency bins obtained in the DFT.

3. **Logarithmic Scaling:** Take the logarithm of the powers at each of the mel frequencies, now clearly defined for discrete bins:

$$L[m] = \log(P[m[k]]) \tag{2.2.7}$$

4. **Discrete Cosine Transform:** Apply the Discrete Cosine Transform (DCT) to the vector of mel log powers to de-correlate the energy bands, now with explicit notation that these operations are on discrete sets of data:

$$\text{MFCC}(l) = \sum_{m=1}^{M} L[m] \cos\left[l(m - 0.5)\frac{\pi}{M}\right] \quad \text{for } l = 1, 2, \ldots, L \tag{2.2.8}$$

5. The MFCCs are the amplitudes of the resulting spectrum, with each component now clearly tied to discrete frequency analysis.

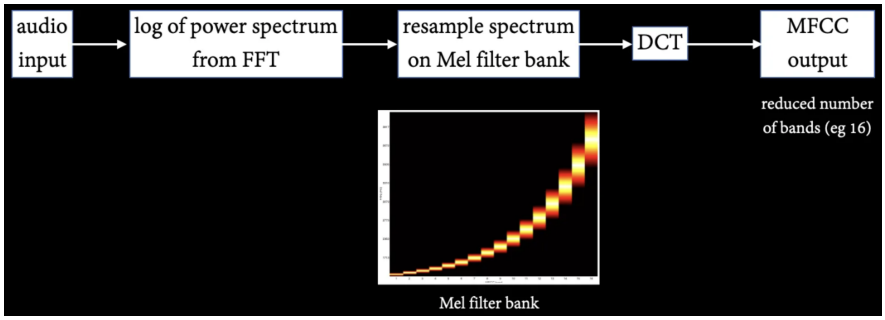


Figure 2.2.4: The process of computing Mel-Frequency Cepstral Coefficients [40].

Chroma Features

Chroma features are an important representation in music signal processing, focusing on the twelve different pitch classes. They condense the entire spectrum of a music piece into 12 distinct bins, each corresponding to one of the 12 semitones of the musical octave. This representation is particularly useful for analyzing the musical content of a signal in terms of harmonies, chords, and melody [41].

The computation of Chroma features typically involves several steps:

1. **Spectral Analysis:** Perform a spectral analysis of the signal, often using the Short-Time Fourier Transform.
2. **Pitch Class Profiling:** Map the spectral energy to each of the 12 pitch classes across all octaves.
3. **Normalization:** Normalize the energy in each pitch class bin to make the representation robust to variations in dynamics.

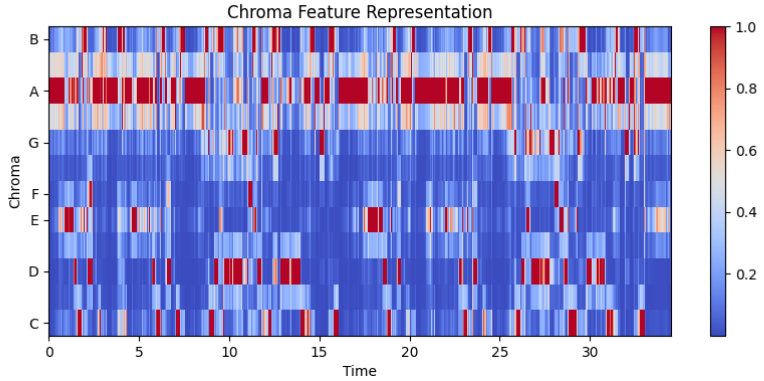


Figure 2.2.5: Chromagram of the same audio signal as Fig. 2.2.3 showing the intensity of the 12 pitch classes over time.

2.2.3 Symbolic Representations

Symbolic representations, such as MIDI (Musical Instrument Digital Interface) and piano rolls, are particularly prominent in music processing for several reasons. MIDI is not only a protocol for the operation of electronic musical instruments but also a powerful tool for capturing the specific features of musical performance, including details like the notes played, their pitch, velocity, and duration. This compact representation allows for precise manipulation and analysis of musical elements.

Piano rolls offer a visual and symbolic representation of music, where music is depicted as a binary matrix with rows corresponding to different pitches (notes) and columns representing successive time steps. This format simplifies the complexity of musical composition into a more accessible form, enabling straightforward

analysis and synthesis of music. The binary nature of piano rolls indicates the presence or absence of notes at given times, making it particularly useful for algorithms that generate or modify music.

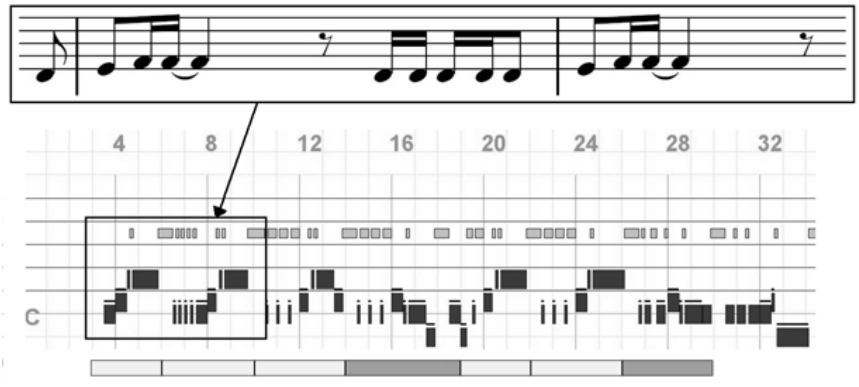


Figure 2.2.6: Graphical representation of a MIDI track [3]

Compared to direct audio representations, symbolic formats like MIDI and piano rolls offer several advantages. Firstly, they compress information about the music into a more manageable form, reducing the dimensionality of data and focusing on the essential musical aspects rather than the audio signal's intricate details. This compression facilitates more efficient processing and analysis, especially in applications like music composition, automated performance, and music information retrieval. Secondly, these representations allow for easier manipulation and editing of musical components, such as altering pitch, tempo, and rhythm, without affecting the audio quality. Lastly, symbolic representations are more interpretable for humans and machines alike, supporting tasks such as music theory analysis, score generation, and interactive music systems.

2.3 Advanced Neural Network Architectures for Music Processing

2.3.1 Convolutional Neural Networks (CNNs)

Introduction to CNNs

Convolutional Neural Networks (CNNs) have revolutionized the field of image processing and have significant applications in audio signal processing, especially in analyzing musical components. CNNs are particularly adept at capturing hierarchical patterns in data, which makes them suitable for tasks involving music, such as genre classification, instrument recognition, and music source separation [4, 5, 6].

A convolutional network's architecture typically encompasses a variety of layers and processes, each contributing to the network's ability to learn complex hierarchical patterns in data. The core components of the architecture include convolution layers, pooling layers, activation functions, fully connected layers and normalization layers. Additionally, the network often integrates processes such as upsampling, concatenation, dropout, and interpolation methods like nearest neighbor or bilinear interpolation. An example of a CNN architecture is illustrated in Fig. 2.3.1.

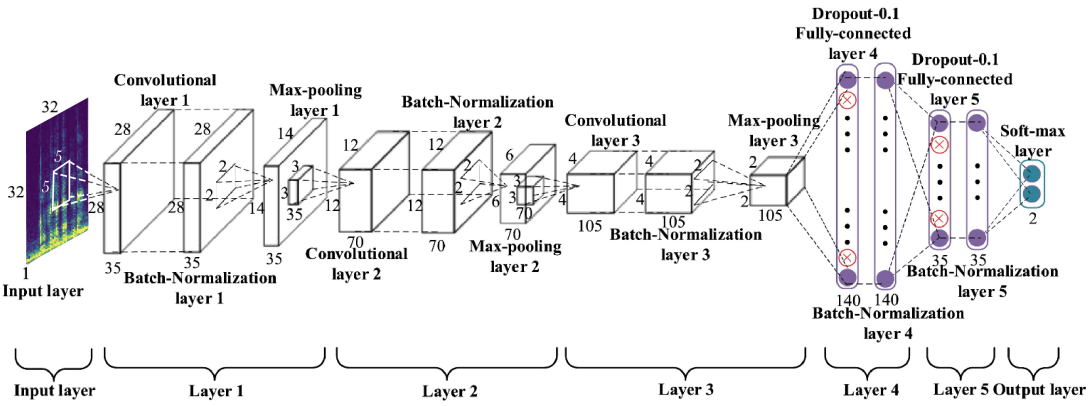


Figure 2.3.1: An example of a CNN architecture. From [7]

Architecture of CNNs

Convolutional Layers: The core of CNNs is the convolutional layers, which are important in feature extraction and pattern recognition within the input data. These layers employ multiple learnable filters (kernels) to systematically scan through the input image or signal. Each filter is designed to detect specific features, such as edges, textures, or more complex patterns at higher layers of the network. Formally, the output of a convolutional filter with a 2D kernel $K \in \mathbb{R}^{K_x \times K_y}$, applied on an input $I \in \mathbb{R}^{I_x \times I_y}$, is given by:

$$\text{Output}(i,j) = \sum_m \sum_n \text{Input}(i+m, j+n) \cdot \text{Kernel}(m,n) \tag{2.3.1}$$

illustrates how the output is computed by applying filters to the input. This operation ensures translational invariance, enabling the network to recognize patterns irrespective of their spatial location in the input. The process involves sliding each filter over the input and computing the dot product between the filter and input at each position, generating a feature map that highlights the presence of detected features.

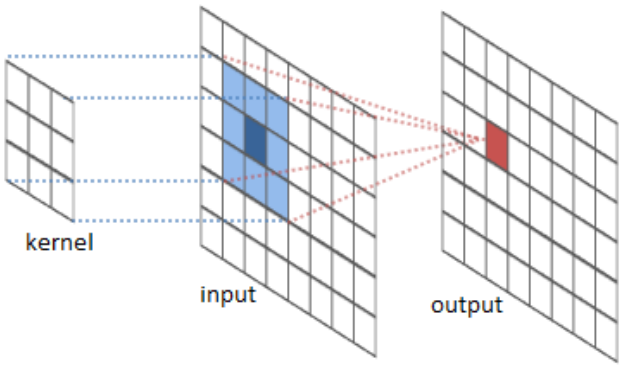


Figure 2.3.2: Visualization of the convolution operation in a CNN. From [42]

The strength of convolutional layers comes from their ability to learn a hierarchy of features. Lower layers may learn to recognize simple patterns such as lines and edges, while deeper layers can detect more complex features by combining the simpler patterns detected by earlier layers. This hierarchical feature extraction makes CNNs extraordinarily effective for tasks involving visual perception, such as image classification, object detection, and beyond. By utilizing multiple convolutional filters, each designed to capture different aspects of the input data, CNNs can adapt to a wide range of tasks and datasets, making them a versatile tool in the field of deep learning.

Dilated Convolution: Dilated convolutions, also known as atrous convolutions, introduce another dimension to standard convolutional layers. They involve skipping input values at regular intervals—a technique that expands the receptive field of the filter without increasing its size. This is particularly beneficial in audio processing, as it allows the network to capture wider temporal context in the data, essential for understanding rhythmic and harmonic structures in music.

$$\text{Dilated Output}(i,j) = \sum_m \sum_n \text{Input}(i + m \times d, j + n \times d) \times \text{Filter}(m,n) \tag{2.3.2}$$

where d is the dilation rate, determining the stride with which the input is sampled.

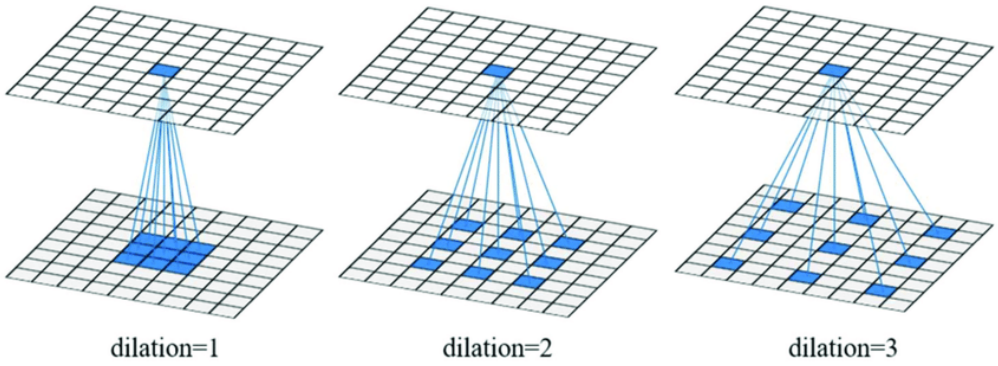


Figure 2.3.3: Visualization of dilated convolution operation in a CNN, showcasing its expanded receptive field. From [43]

Strides and Padding: Stride refers to the number of pixels by which we slide the filter across the input. A larger stride results in a smaller output dimension. Padding involves adding extra pixels around the input border, allowing the filter to be applied to bordering elements and controlling the spatial size of the output.

$$\text{Output Size} = \left\lfloor \frac{\text{Input Size} - \text{Filter Size} + 2 \times \text{Padding}}{\text{Stride}} \right\rfloor + 1 \tag{2.3.3}$$

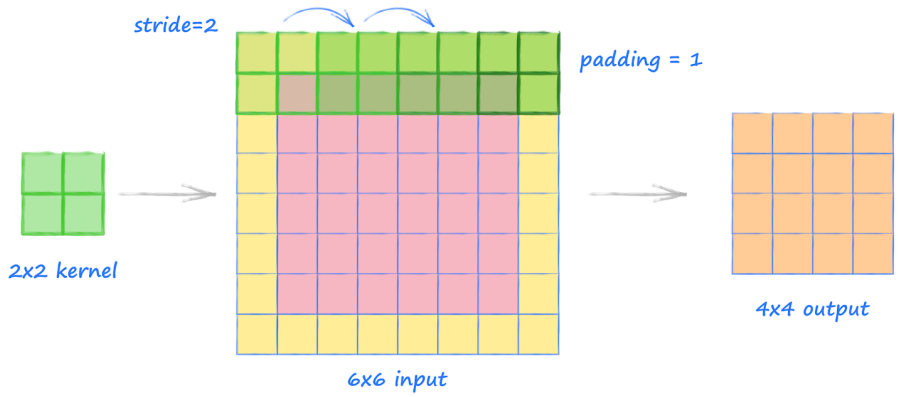


Figure 2.3.4: Illustration of stride and padding effects in a CNN. From [44]

Pooling Layers: Pooling layers reduce the spatial dimensions of the input, lowering computational complexity and parameters. Common pooling methods include max pooling and average pooling.

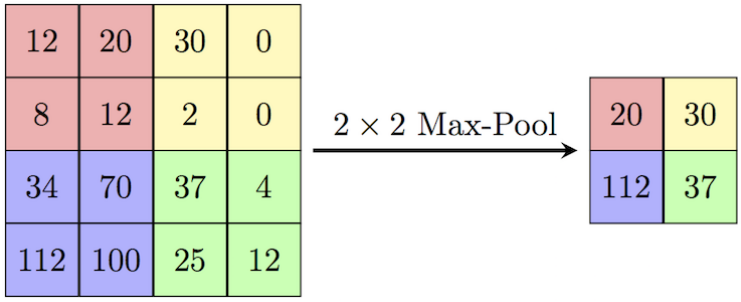


Figure 2.3.5: Example of pooling operations in a CNN. From [45]

Activation Functions: Activation functions introduce non-linearities into the CNN, enabling the network to learn complex patterns beyond linear separations. These functions are applied to the output of convolutional and fully connected layers, with common choices including the Rectified Linear Unit (ReLU) for general purposes, sigmoid for binary classification tasks, and softmax for multi-class classification. The choice of activation function plays a crucial role in the network's ability to converge and the overall model performance. Some of the activation functions depicted in Fig. 2.3.6 are:

$$\text{ReLU}(x) = \max(0, x) \tag{2.3.4}$$

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \tag{2.3.5}$$

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{2.3.6}$$

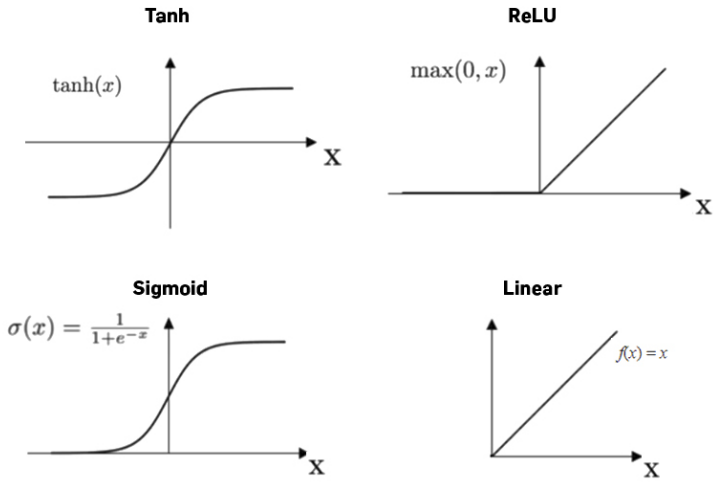


Figure 2.3.6: Visualization of common activation functions used in CNNs. From [46]

Normalization Layers: To improve training stability and efficiency, normalization layers adjust the activations throughout the network, typically after convolutional layers and before activation functions. Batch Normalization and Layer Normalization are widely used, with the former normalizing the inputs across the batch dimension and the latter across the feature dimension. These layers help in faster convergence and mitigate the problem of internal covariate shift.

$$\text{BatchNorm}(x) = \frac{x - \mu_{\text{batch}}}{\sqrt{\sigma^2_{\text{batch}} + \epsilon}} \tag{2.3.7}$$

$$\text{LayerNorm}(x) = \frac{x - \mu_{\text{layer}}}{\sqrt{\sigma^2_{\text{layer}} + \epsilon}} \tag{2.3.8}$$

Where μ and σ^2 are the mean and variance computed over the specified dimension, and ϵ is a small constant added for numerical stability.

Incorporating activation functions and normalization layers into a CNN architecture is critical for enhancing the network's learning capabilities. Activation functions allow the network to capture non-linear relationships, while normalization layers ensure that the learning process remains stable and efficient across different layers of the network.

Fully Connected Layers: The final part of a CNN, fully connected (FC) layers, compile the learned features for output tasks such as classification or regression. These layers form a dense network architecture, where each neuron is interconnected with all activations in the previous layer. The mathematical operation within an FC layer can be represented as:

$$\text{Output} = \text{Activation}(\mathbf{W} \cdot \mathbf{Input} + \mathbf{b}) \tag{2.3.9}$$

where $\mathbf{W}$ represents the weight matrix associated with the connections between the current layer's neurons and the incoming activations, **Input** is the vector of activations from the previous layer, and $\mathbf{b}$ denotes the bias vector added to the weighted inputs before the activation function is applied. The activation function, denoted as Activation, introduces non-linearity, allowing the network to learn complex patterns.

The Perceptron: The concept of the perceptron provides a fundamental understanding of how neural networks, including fully connected layers in CNNs, operate. A perceptron is the simplest form of a neural network unit, designed to perform binary classification. It takes multiple binary inputs, multiplies each with a corresponding weight, and sums them up. This weighted sum is then passed through an activation function, typically a step function, to produce a binary output. The perceptron rule updates the weights based on the error of the output compared to the expected result, gradually learning the optimal weights that minimize the error.

$$\text{a} = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2.3.10}$$

where σ is the activation function, w_i are the weights, x_i are the input features, and b is the bias. The perceptron forms the building block for more complex neural networks by illustrating how individual neurons can make decisions by weighing input signals and applying a non-linear activation.

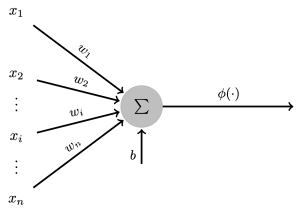


Figure 2.3.7: Perceptron

1D Convolutional Neural Networks: While the aforementioned descriptions primarily focus on CNNs designed for 2D input data, such as images, the principles and components of CNN architectures can be similarly applied to 1D data. 1D CNNs are particularly effective for analyzing sequential data, including audio signals, time series data, and text. In these applications, 1D convolutional layers operate by sliding filters along a single dimension, extracting patterns and features across temporal or sequential dimensions. This makes 1D CNNs adept at tasks such as audio genre classification, sentiment analysis from text, and forecasting in time series data.

CNNs in Music Representation

CNNs are adept at processing time-frequency representations of audio, such as spectrograms. They can efficiently capture spectral or temporal dependencies and recognize textural patterns, making them suitable for various music analysis tasks.

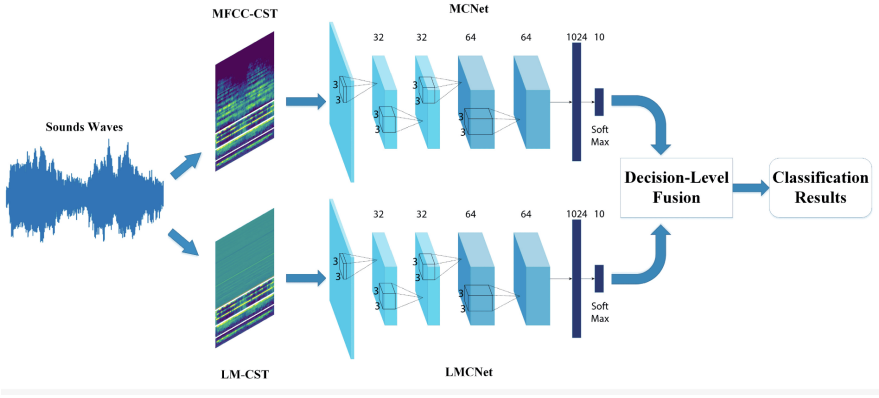


Figure 2.3.8: Visualization of a CNN on a sound inference. From [47]

2.3.2 Recurrent Neural Networks (RNNs)

Introduction to RNNs

Recurrent Neural Networks (RNNs) are a class of neural networks that are particularly powerful for sequential data processing, making them ideal for applications in music, where data is inherently sequential. RNNs have internal memory elements to store information about previous inputs, enabling them to capture temporal dependencies.

Architecture of RNNs

RNNs process sequences by iterating through the sequence elements and maintaining a state that encapsulates information learned from previous elements. The basic formula for a simple RNN is given by:

$$h_t = \text{activation}(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{2.3.11}$$

where h_t is the hidden state at time t, x_t is the input at time t, W_{xh} and W_{hh} are weights, and b_h is the bias.

Problems with Basic RNNs Basic Recurrent Neural Networks (RNNs) often encounter significant challenges when processing long sequences, notably the vanishing and exploding gradient problems. The vanishing gradient problem occurs when the gradients of the loss function decrease exponentially as they are propagated back through time, making it difficult for the RNN to learn and retain information from earlier inputs in a sequence. Conversely, the exploding gradient problem involves the gradients growing exponentially, potentially leading to numerical instability and divergent learning processes. These issues compromise the RNN's ability to effectively capture long-term dependencies in sequence data.

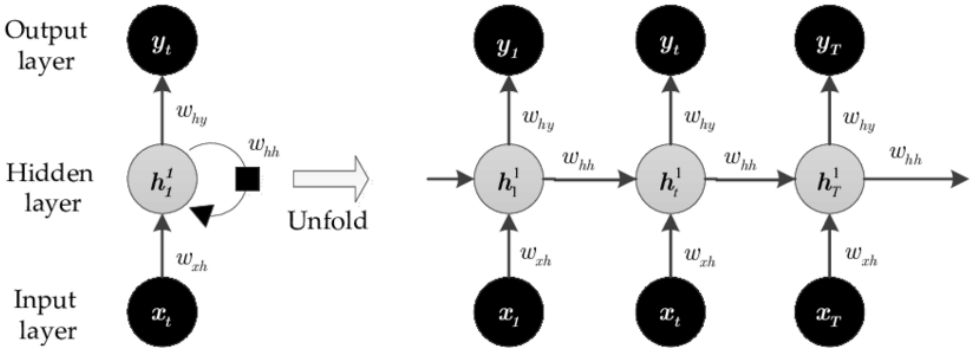


Figure 2.3.9: An example of an RNN architecture. From [8]

To mitigate these problems, advanced architectures like Long Short-Term Memory (LSTM) networks and Gated Recurrent Unit (GRU) networks have been developed. Schmidhuber et al. [9], incorporate memory cells that can maintain information across long sequences, effectively addressing the vanishing gradient problem. Cho et al. [10] , proposed GRU networks that offer a simplified version of LSTMs with fewer parameters and have been shown to perform comparably in many tasks. Both LSTM and GRU architectures are designed to retain information over long sequences, making them well-suited for tasks that require understanding temporal dependencies, such as language modeling and time series forecasting.

Structure of RNN/LSTM-like Architectures RNN, LSTM, and GRU architectures share a foundational structure that processes data sequentially, allowing these networks to maintain a form of memory over inputs through time. In a basic RNN, this structure consists of a loop that reuses the same weights at each timestep, effectively enabling the network to pass information from one step to the next. However, LSTMs and GRUs introduce more sophisticated mechanisms to this loop.

LSTM architectures are structured around a series of gates: the input gate, the forget gate, and the output gate, along with a cell state. These components work together to regulate the flow of information, allowing the network to decide which data should be retained or discarded as it processes each timestep. This structure enables LSTMs to mitigate the vanishing gradient problem by maintaining a cell state that can carry relevant information throughout the processing of the sequence, regardless of length.

GRU networks simplify the LSTM design by combining the input and forget gates into a single update gate and merging the cell state and hidden state. This streamlined architecture still allows for effective regulation of information flow but with fewer parameters than LSTMs, making GRUs computationally more efficient in some cases.

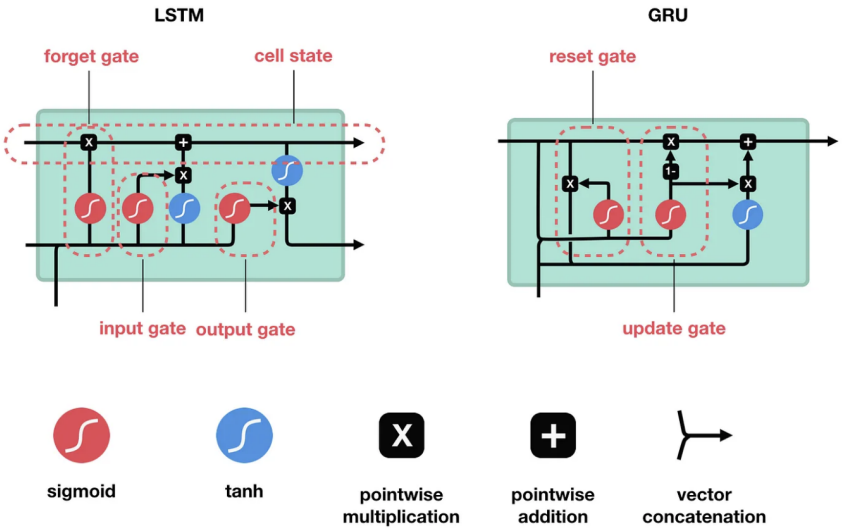


Figure 2.3.10: LSTM compared to GRU. From [48]

Both LSTM and GRU units are composed of these gates and states, forming the building blocks of more complex neural network architectures. These blocks are repeated across layers and sequences, enabling the network to perform deep learning on sequential data. By structuring the networks in this way, RNNs, LSTMs, and GRUs can learn patterns in time series data, speech, text, and other sequential forms of information, providing the basis for applications ranging from language translation to stock market prediction.

RNNs in Music Representation

RNNs, particularly LSTMs and GRUs, are well-suited for music generation, transcription, and even source separation. They can effectively model temporal dynamics and dependencies in musical elements, capturing the long-term structure in music.

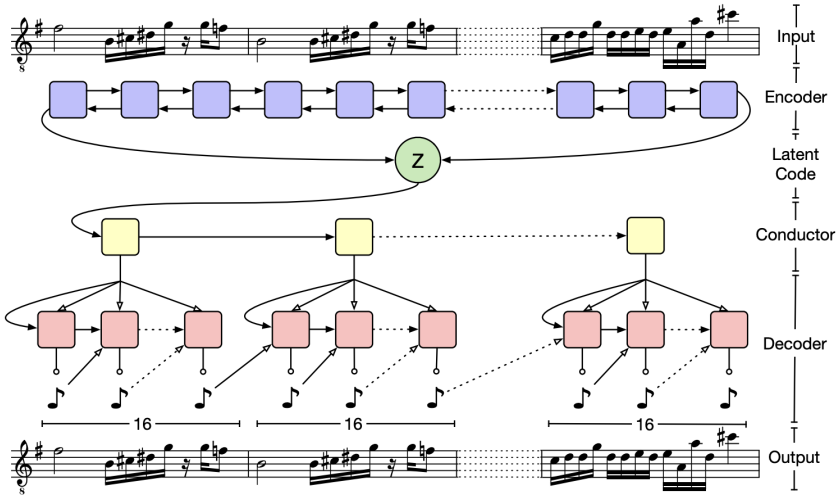


Figure 2.3.11: An example of an RNN architecture for music processing. From [49]

2.3.3 Transformers

Vaswani et al. [50] introduced transformers, an architecture that has revolutionized the field of deep learning by introducing a mechanism to process sequential data without relying on recurrent architectures. Their

core is the self-attention mechanism, which allows the model to focus on different parts of the input sequence when predicting an output.

The Self-Attention Mechanism

The self-attention mechanism enables the model to dynamically emphasize the importance of certain parts of the input data over others. Mathematically, the attention function can be described as mapping a query and a set of key-value pairs to an output, where the output is a weighted sum of the values. The weight assigned to each value is computed by a compatibility function of the query with the corresponding key. Mathematically,

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.3.12}$$

where Q, K, and V represent the queries, keys, and values matrices, respectively, and d_k is the dimensionality of the keys.

Transformer Architecture

The Transformer model architecture consists of an encoder and a decoder, each composed of multiple layers of self-attention and fully connected feed-forward networks.

Detailed Structure of Encoder Each encoder layer consists of two sub-layers: a multi-head self-attention mechanism, and a simple, position-wise fully connected feed-forward network. Normalization is applied before each sub-layer, with a residual connection around each of the two sub-layers.

$$\text{LayerNorm}(x + \text{Sublayer}(x))$$

Sublayer refers to either the multi-head self-attention mechanism or the feed-forward network. This approach is replicated across N identical layers.

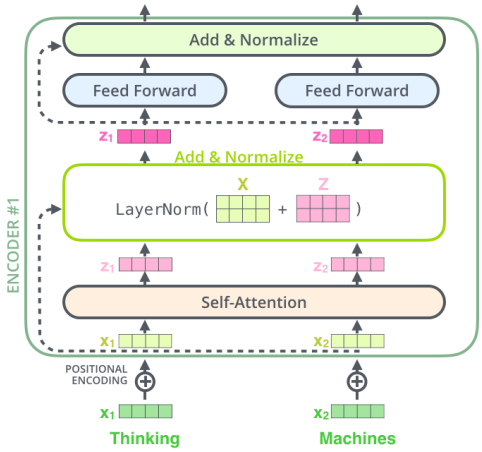


Figure 2.3.12: Illustration of Encoder. From [51]

Multi-Head Self-Attention Instead of performing a single attention operation, the Transformer model employs multiple attention heads to capture different representation subspaces at different positions. This allows the model to attend to information from different representation subspaces at different positions simultaneously, enhancing its ability to understand complex dependencies in the data. To this end, for each attention head, the input X is linearly transformed by multiplying it with learnable weight matrices W^Q,

W^K, and W^V respectively, in order to produce the query (Q), key (K), and value (V) matrices upon which the attention operation is performed.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$$

$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

In this formulation, W_i^Q, W_i^K, and W_i^V are the weight matrices for the i^{th} attention head for queries, keys, and values respectively, and W^O is the output weight matrix that combines the outputs of all attention heads.

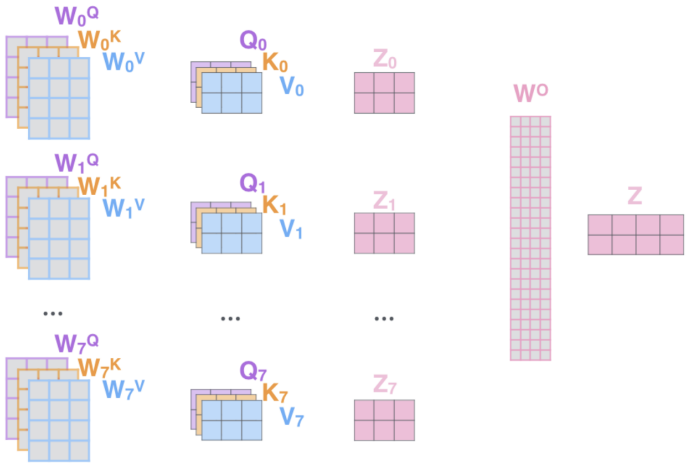


Figure 2.3.13: Illustration of multiple Heads of the transformer. From [51]

Position-wise Feed-Forward Networks Each layer in the encoder and decoder contains a fully connected feed-forward network, which is applied to each position separately and identically.

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

Detailed Structure of Decoder The decoder also consists of N identical layers. In addition to the two sub-layers that are also present in each encoder layer, the decoder inserts a third sub-layer, between the multihead attention block and the feedforward block, which performs multi-head attention over the encoder's output.

Masked Self-Attention In the decoder, the self-attention layer is modified to prevent positions from attending to subsequent positions. This masking ensures that predictions for position i can depend only on the known outputs at positions less than i.

Normalization and Residual Connections Each sub-layer in the encoder and decoder, including self-attention, feed-forward networks, and the additional encoder-decoder attention in the decoder, is equipped with normalization and residual connections, promoting faster training and mitigating the vanishing gradient problem.

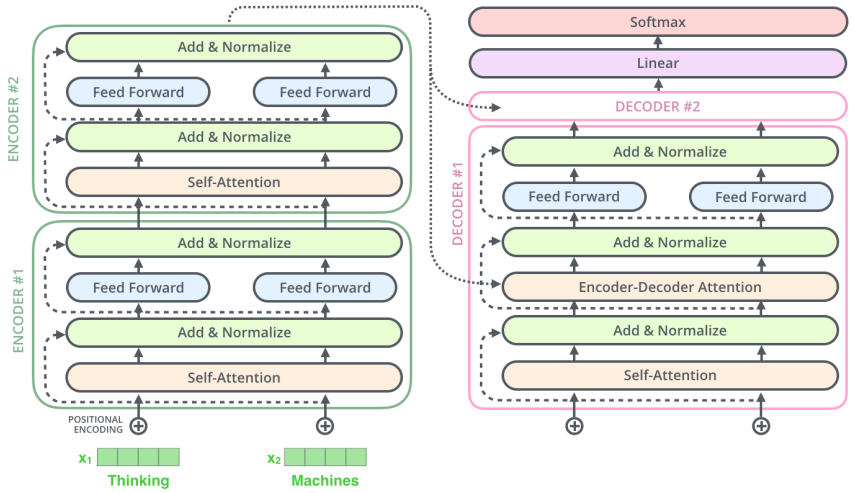


Figure 2.3.14: Illustration of Encoder and Decoder. From [51]

Application in Music Processing

Transformers find applications in various music processing tasks such as music generation, transcription, and source separation [52, 32]. Their ability to handle long-range dependencies makes them particularly suited for capturing musical structure and context.

Correlation to Music: The Transformer's self-attention mechanism can analyze the temporal structure of music, enabling it to understand complex relationships and patterns in musical compositions. At the same time as discussed in [33] it can be used for modelling cross domain relationships between spectral and audio representations.

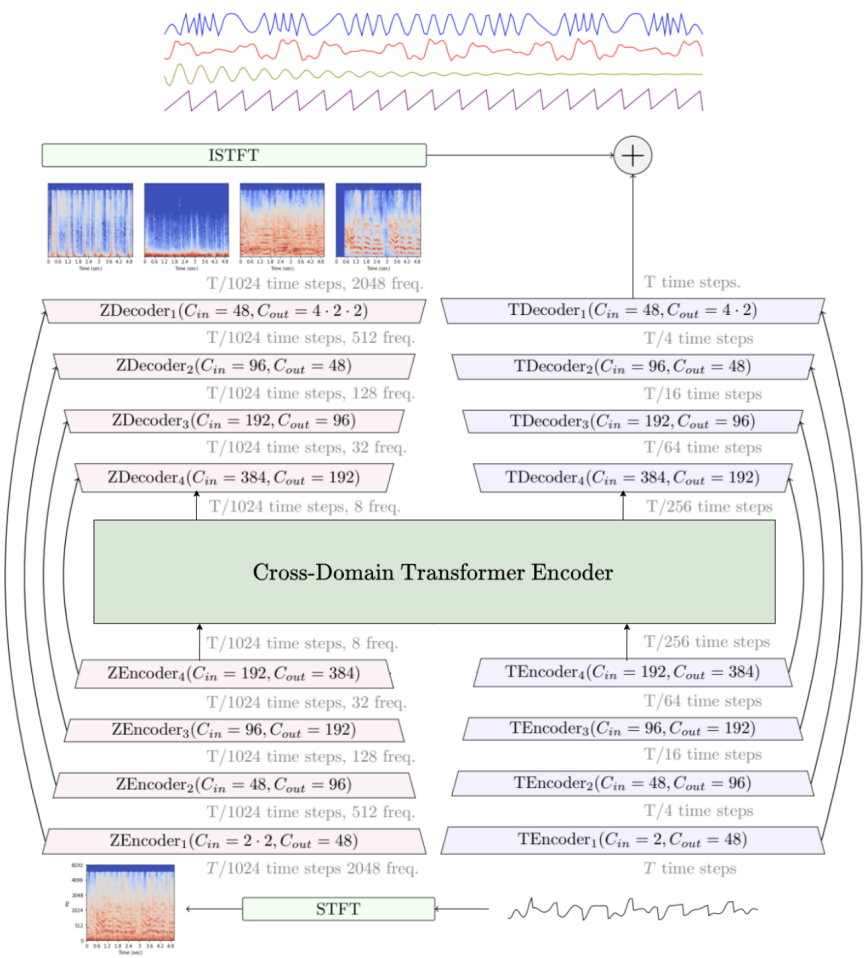


Figure 2.3.15: Cross Domain Transformer Encoder. From [33]

2.3.4 Benes Networks

Overview of Benes Networks

Benes Networks, known from their application in packet routing tasks in computer networks, have recently found their place in the realm of neural network architectures for processing long sequences. These networks are characterized by their unique structure, consisting of interleaved shuffle and switch layers. The shuffle layers are responsible for permuting the signals, while the switch layers consist of switches that can either swap two adjacent signals or leave them unchanged. This distinctive arrangement allows Benes Networks as depicted in Fig. 2.3.16 to efficiently route signals from input to output.

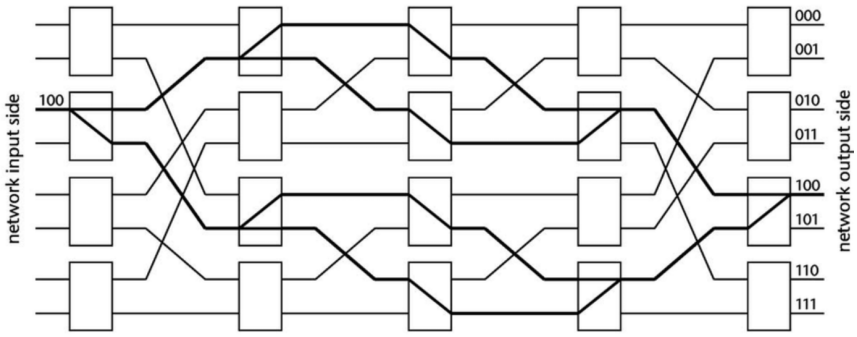


Figure 2.3.16: Illustration of Benes Network. From [11]

Neural Shuffle-Exchange Networks

In the context of neural networks, the Benes Network structure is adapted in the form of Neural Shuffle-Exchange Networks. These networks replace each switch of a Benes Network with a Switch Unit, a learnable 2-to-2 function as depicted in Fig. 2.3.17. The input to these networks is a sequence of a specific length, with each element being a multi-dimensional vector. The network's first layer consists of a series of Switch Units, each processing a pair of adjacent sequence elements. Following this are the shuffle layers, which permute the inputs based on a defined permutation pattern, such as the perfect shuffle permutation.

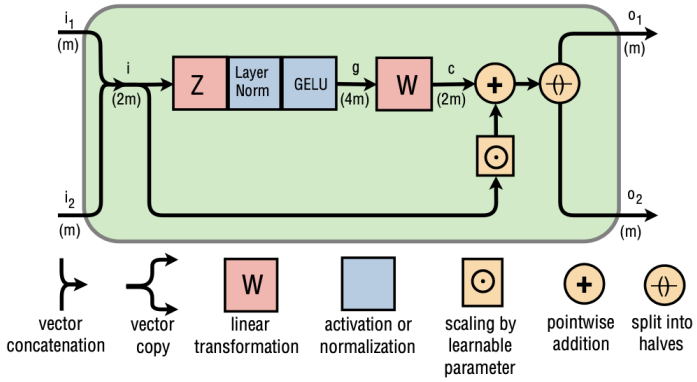


Figure 2.3.17: Residual Switch Unit. A number of feature maps (m) is shown in parentheses. Depicted here with the default of hidden layer being 2× larger than the input (4m being the size of the hidden layer and 2m the size of the input) [12].

Functioning and Applications

The combination of regular Shuffle-Exchange Networks followed by reversed Shuffle-Exchange Networks, which reshuffle the input sequence to the right by 1 element after the learnable switch operation, forms a Benes block. This block can connect any input to any output, allowing the network to have a receptive field encompassing the entire sequence without any bottlenecks. Such networks can process sequences of significant lengths, up to millions of elements, making them particularly suitable for tasks where long-range dependencies are crucial, such as in music transcription or sequence processing applications where traditional methods like attention mechanisms are less effective due to their computational complexity.

Advantages in Music Processing

The Residual Shuffle-Exchange Network, a variant of the Neural Shuffle-Exchange Network, demonstrates state-of-the-art performance in tasks like music transcription. This network's architecture, which uses signif-

icantly fewer parameters compared to other models, highlights its efficiency and effectiveness in processing long sequences, a critical requirement in music processing applications [12]. We will analyze this architecture in depth in Sec.5.

Chapter 3

Literature Review

Contents

Source separation is a fundamental subfield of audio signal processing, and its principles and methods are used in many research areas. Yet, the vast majority of its literature orbits around separating sources with different timbres, a scenario distinct from the challenges posed by monotimbral source separation. In this literature review, as we explore the fundamental principles of general source separation in depth —spanning methodologies, data augmentation techniques, architectures, and evaluation protocols— we also underscore the unique facets of monotimbral separation. This includes specialized considerations like permutation-invariant training, the imperative for conditioning, and more. Our intent is to both present a comprehensive overview of source separation at large and to spotlight where the literature potentially lacks in addressing the specific challenges of separating sources with identical or near-identical timbral characteristics.

3.1 Traditional Signal Processing Approaches

Source separation, as previously highlighted, is a complicated issue with various dimensions. One foundational way to categorize it is based on the count of sources and sensors. When the number of sensors exceeds or matches the sources, we term the scenario as over-determined or determined, in that order. Conversely, if sensors are outnumbered by sources, we define the issue as under-determined.

For the former categories, matrix factorization strategies, particularly those anchored in Independent Component Analysis (ICA) [53, 54], have demonstrated considerable success. ICA is designed to divide a signal into additive components, assumed to be *non-Gaussian* and *statistically independent*. This technique represents a mixed signal from n sensors, $\mathbf{x}$, as a product of an $n \times p$ mixing matrix with linearly independent columns, $\mathbf{A}$, and p statistically independent vectors, $\mathbf{s}$:

$$\mathbf{x} = \mathbf{A}\mathbf{s}. \tag{3.1.1}$$

The objective of ICA is to determine an unmixing matrix that closely aligns with the pseudoinverse of $\mathbf{A}$, denoted as $\mathbf{W} = \mathbf{A}^{+}$, ensuring that the inferred components $\mathbf{u}$ maintain as much statistical independence as feasible.

$$\mathbf{u} = \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{A}\mathbf{s}. \tag{3.1.2}$$

However, ICA's elegance falls short in monaural source separation, predominantly because it mandates more sensors than sources.

For the under-determined scenarios, especially the single channel challenge, which is central to this study, traditional techniques can be divided into three overarching classes:

- Spectral Decomposition Based
- Model Based
- Computational Auditory Scene Analysis (CASA) Based

In **spectral decomposition**, an input mixture's representation is dissected into fundamental elements, subsequently organized into separate sets symbolizing the distinct sources. The signal's representation could adopt varied forms, yet commonly, it's the magnitude or power spectrogram of the mixture, derived via the STFT. The uniqueness of these methods lies in the decomposition and grouping criteria.

One such method is the *Independent Subspace Analysis (ISA)*, a broader application of ICA. Here, the basis elements of a group can co-exist without the statistical independence assumption, but this constraint persists among elements of divergent groups. A technique hinging on ISA, as discussed in [55], is deployed to fragment the mixture spectrogram into independent source domains, which are then reverted to obtain the separated sources.

The *Non-Negative Matrix Factorisation (NMF)* is another decomposition strategy. As inferred from its designation, NMF insists on non-negativity across all matrices. In the decomposition equation:

$$\mathbf{V} = \mathbf{W}\mathbf{H}, \tag{3.1.3}$$

where $\mathbf{V}$ represents the known matrix, $\mathbf{H}$ is the matrix of basis vectors, and $\mathbf{W}$ is the weight matrix.

All these matrix elements uphold non-negativity. This non-negative stipulation lends the technique a perceivable significance, often absent in most matrix decompositions. Specifically, non-negative weights ascertain an exclusively additive amalgamation of the basis elements, and the non-negative basis vectors eliminate potential mutual cancellations. Given their inherent non-negativity, magnitude and power spectrograms are apt for this technique. The learned matrices $\mathbf{W}$ and $\mathbf{H}$ encapsulate valuable information for source separation. Specifically, $\mathbf{W}$ encodes the spectral profiles of distinct sources, while $\mathbf{H}$ reveals the temporal activation patterns corresponding to the predefined number of sources. The number of sources to be separated can be explicitly defined by choosing the inner dimension of the matrices. Let $\mathbf{V}$ be the observed matrix, $\mathbf{H}$ the matrix of basis vectors, and $\mathbf{W}$ the weight matrix, with dimensions $(F \times T)$, $(S \times T)$, and $(F \times S)$, respectively, where F is the number of features, T is the number of time points, and S is the user-defined number of sources.

The objective during the algorithm's training is to minimize the error between the reconstructed matrix $\hat{\mathbf{V}} = \mathbf{WH}$ and the initial observed matrix $\mathbf{V}$. After training, to extract each sound source k, the source matrix can be calculated as:

$$\hat{\mathbf{V}}_k = \mathbf{W}_k \mathbf{H}_k^T$$

where $\mathbf{W}_k$ is the k-th column of $\mathbf{W}$ and $\mathbf{H}_k$ is the k-th row of $\mathbf{H}$. This decomposition allows for the isolation of individual sound sources from the observed data, providing flexibility and control over the source separation process.

A technique built on NMF is discussed in [56, 57, 58], where an additional component promoting temporal continuity is integrated during the weight and basis vector matrices' estimation.

CASA aims to replicate the human auditory system's decoding process. Employing psychoacoustical indicators like harmonicity and onset-offset time, CASA strategies [59, 60] generate streams grounded in pitch closeness. Nonetheless, these techniques stumble when segregating sources that overlap at the same pitch.

In **model-based** strategies, generative templates of the source signals are crafted to facilitate the separation process. Given that these models extrapolate parameters from solo samples, they are highly susceptible to the recording surroundings. Such models might hinge on Hidden Markov Models (HMM), as illustrated in [61].

3.2 Machine Learning Approaches

The progression of deep learning has catalyzed a surge in wholly supervised methods. This section aims to explore the spectrum of DNN methodologies, from non-score informed techniques that utilize classification and leverage time and frequency, to score-informed approaches and the application of CNNs, illustrating the breadth and depth of DNN applications in current research.

3.2.1 Non-Score-Informed Techniques

Classification

Depending on the domain of data processing, we have waveform-centric techniques that leverage the 1D "natural" audio data representation, and spectrogram-centric methods that employ a 2D time-frequency data transformation. This transformation might be predetermined, like the Short Time Fourier Transform (STFT) magnitude, or one that's independently discerned.

For strategies utilizing spectrograms, based on signal estimation, there's direct estimation, which learns the source signal spectrograms directly, and indirect estimation, where the model deduces a 2D mask for every source. This mask is element-wise multiplied with the input spectrogram to deduce the original source signal. An illustration of a *binary* masking producing a source estimate in Fig. 3.2.1. Common masking applied for source separation **is not** binary but continuous.

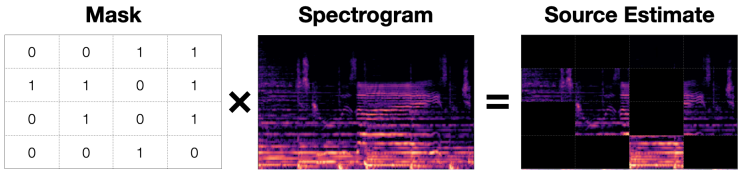


Figure 3.2.1: Illustration of binary masking [13]

Signal estimation methodology doesn't majorly dictate the model's efficacy, although masking techniques have a maximal constraint defined by the Ideal Binary Mask (IBM) and Ideal Ratio Mask (IRM). In contrast, direct estimation can, in theory, flawlessly retrieve the source signals. As outlined in [62], the IBM and IRM for time-frequency signals are distinctly characterized. However, each choice concerning the computation domain comes with its pros and cons.

Time-frequency domain data are essentially more compact than waveforms, as a result models demand less computation, shortening training durations. Essential details, like temporal dependencies, can be harnessed by simpler models, leading in smaller parameter footprints. Plus, 2D representation models can adapt methodologies from the well-explored realm of image processing. Conversely, adopting the STFT magnitude for time-frequency representation isn't without pitfalls. Primarily, the STFT isn't tailored for source separation. Most approaches neglect the signal's phase when estimating sources, thus decreasing optimal performance by omitting crucial information. Consequently, the phase of the split signals isn't deduced. To bypass this, some methods presuppose the source phase matches the mixture's [63], or they approximate it using the Griffin-Lim algorithm [64]. Unlike magnitude, estimating phase via DNNs is tricky, given its cyclic nature, leading to inconsistencies at junctures (like when values range between minus pi and pi, the juncture being pi). Phase unwrapping could be a potential remedy, but it shifts the challenge, enlarging the value span, making phase estimation by DNNs still challenging. Nevertheless, the significance of phase data in tasks like speech amplification [65] and audio separation is undeniable. Indeed, [66] reports promising outcomes by innovatively turning the regression challenge of phase estimation into a classification one.

Techniques Leveraging Time-Frequency (2D) Representations

Utilizing the time-frequency domain, especially via the STFT, has become common in a majority of audio applications. Given this, it's unsurprising that the DNN-based source separation research ventured into this domain. This preference, combined with many efficient techniques preferring this domain, is why our literature exploration commences with these methodologies. Initially, due to constraints in resources and the growing stages of machine learning tools, the focus was on relatively straightforward and shallow neural networks. In [67], one of the foundational forays into speech separation using DNNs, a basic network was employed to combine single-frame estimates from NMF in a nonlinear manner. This non-linearity in basis vector combinations was seen as enhancing separation potential, while introducing non-linear activation functions between layers seemed to boost the model's expressive capability. Conversely, in [68], the approach involved feeding the network with an assortment of adjacent frames to offer temporal context.

As years progressed, DNNs evolved in depth and complexity, incorporating diverse layers to refine separation efficiency. Recognizing that audio signals could have extended temporal dependencies, researchers integrated recurrent layers into designs to adeptly manage lengthy frame sequences. Huang et. al [14] explored deep recurrent neural networks (DRNN) and their varied temporal connections. Jannson et al. [15] introduced a U-Net [16] adaptation for spectrogram-based music segregation as illustrated in Fig. 3.2.2. This deep autoencoder consists of several 2D convolutional layers, functioning at varied scales through upsampling and downsampling, targeting both micro and macro patterns.

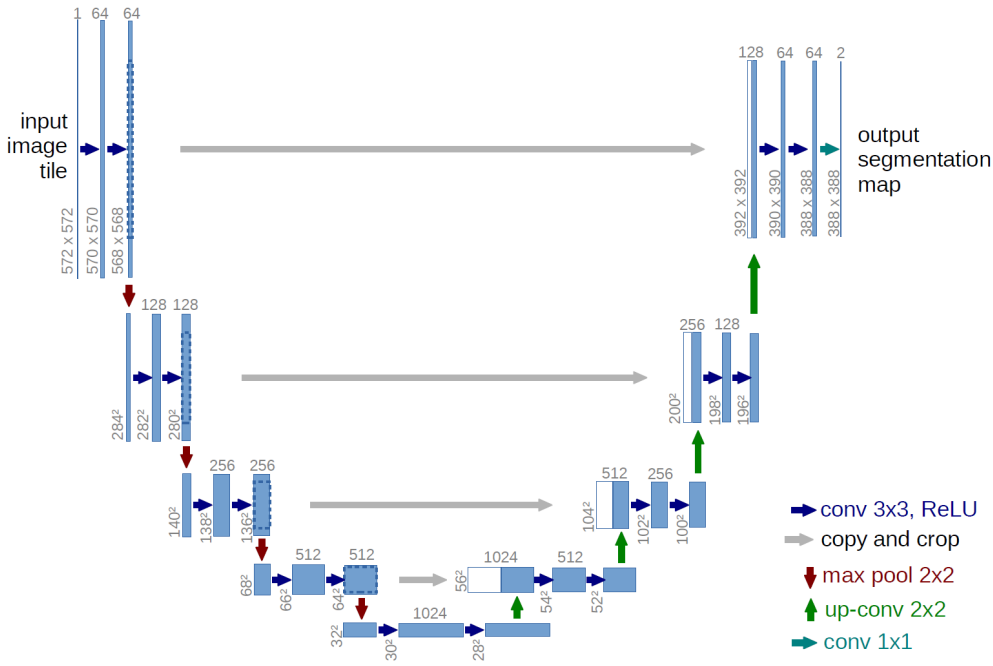


Figure 3.2.2: U-Net Architecture for Source Separation From [16]

Methods using Time/Waveform (1D) Representations

Techniques within the waveform domain distinguish themselves by conventional feature extraction frontends, like the Short-Time Fourier Transform (STFT), in favor of handling data in its one-dimensional form. This approach encompasses either employing learned transformations to derive latent representations or directly processing the waveform data end-to-end. A key advantage of operating in the waveform domain is the preservation of phase information, which is often lost in magnitude-only representations used in other methods.

These 1D methodologies do not imply that the resultant latent representations are one-dimensional; rather, it suggests that the data undergoes processing in a manner akin to multi-channel one-dimensional signals. This is reflected in the activation maps of various layers being one-dimensional. In practice, where traditional methods might apply two-dimensional convolutions, these techniques utilize one-dimensional convolutions to process the data.

Central to the discussion of 1D techniques are two architectures: TasNet [17] and Wave-U-Net [4]. Each has significantly influenced the landscape of audio processing, particularly in tasks like speech enhancement and music separation.

TasNet employs a strategy where the input waveform is transformed into a latent representation through an encoder, which is then manipulated by a separator network to isolate source signals as illustrated in Fig. 3.2.3. This process involves:

- Encoding the input signal into a mixture of basis vectors and weights,
- Applying masks to these weights via the separator to isolate individual sources,
- Reconstructing the source signals through a decoder using the masked weights.

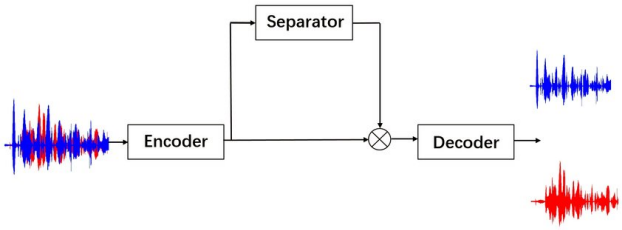


Figure 3.2.3: TasNet Architecture for Source Separation From [17]

Originally, TasNet featured a separation module comprising a deep LSTM network to capture temporal dependencies, augmented by a fully connected layer. Variations on this architecture have experimented with different separator designs to enhance performance, often maintaining the encoder and decoder components unchanged. Innovations include modifications to incorporate RNNs that process both feature and channel dimensions, the introduction of Temporal Convolutional Networks (TCNs) for efficiency, and even meta-learning approaches [69, 70] for separator adaptation.

Wave-U-Net, conversely, provides a simpler yet effective architecture adapted from the U-Net model for 1D audio signals illustrated in Fig. 3.2.4. It consists of:

- A downsampling path to distill multi-scale information,
- A bottleneck layer for dense representation processing,
- An upsampling path that restores the signal to its original dimension, leveraging skip connections for feature integration.

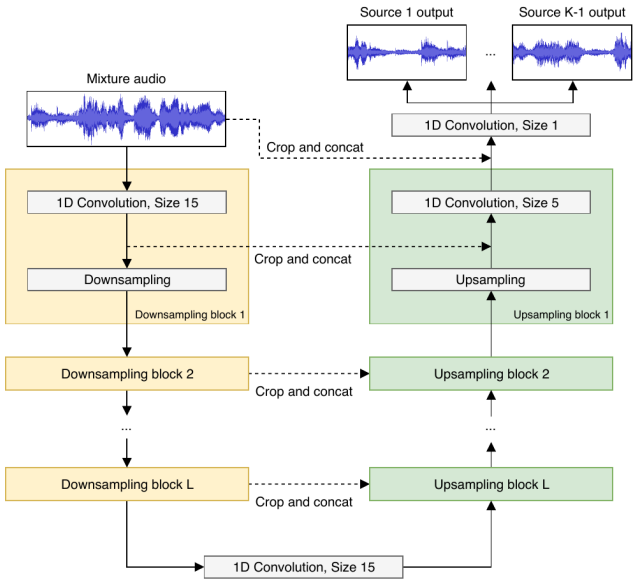


Figure 3.2.4: Wave-U-Net Architecture for Source Separation From [4]

Subsequent research on Wave-U-Net has explored modifications to its core components [71, 72], aiming to leverage long-range temporal information, introduce attention mechanisms for feature selection, and refine the downsampling process to preserve information fidelity.

Both TasNet and Wave-U-Net exemplify the versatility and potential of waveform-based processing techniques in audio signal analysis, demonstrating how direct manipulation of the waveform can yield significant insights and improvements in various audio processing tasks.

In addition to TasNet and Wave-U-Net, the Demucs [73] architecture represents another significant advancement in the realm of audio processing techniques. Initially, Demucs utilized the power of the 1D time domain for signal processing. However, as the architecture evolved, it uniquely positioned itself by simultaneously utilizing both the temporal and spectral domains of signals. This dual-domain approach allows for a more comprehensive analysis and reconstruction of audio data, showcasing the architecture's innovative integration of time and frequency information. Since the Demucs model, due to its unique approach in leveraging audio signals, forms the the cornerstone of the research and practical exploration undertaken in this thesis, it will be thoroughly analyzed in a later chapter 6.

3.2.2 Score-Informed-Techniques

Score-informed music source separation leverages musical scores to enhance the accuracy of separating individual instruments or voices from a mixed audio signal. This approach contrasts with traditional source separation methods that rely solely on the audio signal. We can characterize the activity labels of each note as soft labels. Most of the practices using soft labels are done within DNN Architectures like CNNs.

Score-informed techniques offer several advantages over traditional source separation methods:

- **Improved Accuracy:** Leveraging score information leads to more accurate separation of instruments, particularly in complex polyphonic music.

- **Efficient Handling of Polyphony:** These techniques are particularly effective in handling polyphonic textures found in classical and chamber music. Versatility in Applications: Score-informed separation is applicable in a range of contexts, including remixing, karaoke systems, and music analysis tasks.

Convolutional Neural Networks (CNNs)

As explored in "Monaural Score-Informed Source Separation for Classical Music Using Convolutional Neural Networks" [74] CNNs can be effectively used for score-informed source separation. This approach involves using score-informed constraints in a convolutional neural network (CNN) architecture to improve source separation. The method first derives training features from audio files and their corresponding scores. One notable characteristic of the scores is that each score is a series of *harmonic partial* trying to mimic not only the base frequency of a note but also its harmonics. These features, known as score-based soft masks and *score-filtered spectrograms*, are then used to train the CNN. The model is trained on synthetic renditions of music scores and is capable of separating real-life performances based on these scores. The paper demonstrates that this approach achieves better performance, and is less computationally intensive compared to score-informed Non-negative Matrix Factorization (NMF) systems. The use of score labels in this context helps the model in accurately separating sources by providing additional contextual information about the timing and frequency of notes played in the music, leading to more effective and efficient source separation.

At the same time a notable work in "Improved Separation of Polyphonic Chamber Music Signals by Integrating Instrument Activity Labels" [18] presents another score informed music source separation, particularly focusing on polyphonic chamber music. The system uses a U-Net architecture, modified from the Demucs model, for separation in the time domain. This architecture includes several encoder and decoder blocks with bidirectional LSTMs or GRUs in the bottleneck. The number of output tracks is set based on the desired instrument number. Time-dependent instrument activity labels, indicating whether a specific instrument is playing at any given time, are integrated into the network. This integration occurs at various stages of the encoder-decoder architecture, with a focus on combining these labels with encoder inputs. Integrating instrument activity labels improves the system's ability to separate sources by providing additional context. The architecture overview is illustrated in the Fig. 3.2.5

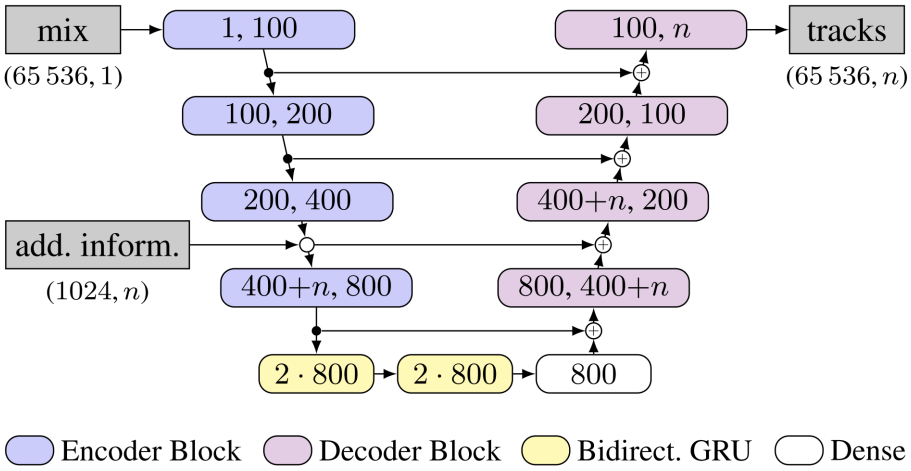


Figure 3.2.5: Schematic model structure with channel numbers of each layer for additional information integration. From [18]

The approach was tested using both simulated and real instrument activity labels, showing significant improvement in separation quality over traditional methods that don't use such labels. The paper explores both a joint model (predicting all source signals simultaneously) and independent models for each instrument, highlighting their flexibility and robustness. The integration of activity labels before the deepest encoder block is shown to yield the best results. Experiments demonstrate the effectiveness of this approach with real and simulated label errors, showing that the model maintains robustness even with some label inaccuracies. The results show significant improvements in separation quality compared to traditional approaches, validating the effectiveness of integrating time-dependent instrument activity labels.

Cheuk . [75] present an advanced framework called "Jointist" for music transcription and source separation. This framework integrates three key modules: Instrument Recognition (fIR), Transcription (fT), and Source Separation (fMSS) as illustrated in Fig. 3.2.6.

Figure 3.2.6: Jointlist Architecture. From [75]

Jointist handles multi-instrument transcription by being aware of different instruments and their specific characteristics. It uses Transformer networks for instrument recognition, combining a CNN front end with a transformer back end. The transcription module is inspired by the onsets-and-frames model, modified to condition on instrument vectors. The source separation module utilizes predicted piano rolls and the STFT spectrogram for separation. The paper demonstrates how Jointist effectively improves performance in transcription and separation tasks, and explores its application in areas like downbeat, chord, and key estimations, as well as music classification. The experiments show that Jointist, with its joint training approach and modular design, offers significant advancements in handling complex polyphonic music scenarios.

Introduction to Music Transcription In the following chapters of this thesis, we will delve into experiments and discussions surrounding various transcription models that will help us with the score informed separation. To ensure thoroughness and provide a foundational understanding, we will introduce the task of

music transcription. **Music transcription** is the process of converting a music audio signal into a symbolic representation, typically in the form of a musical score or a MIDI file.

Yang et al. [76] introduced the Complex Transformer, an approach that leverages complex numbers to enhance the Transformer model's capacity for sequence modeling, particularly for tasks involving audio signals such as music transcription. The model's architecture is designed to handle complex-valued input, enabling it to process the naturally complex-valued representations of audio signals obtained after a Fourier Transform. Notably, the Complex Transformer achieved state-of-the-art results

Building upon the foundation of Transformers for sequence modeling, Gardner et al. [52] presented MT3, a multi-task, multi-track music transcription model that utilizes a general-purpose Transformer to transcribe an array of musical instruments from various transcription datasets. MT3's innovation lies in its unified training framework, which significantly improves transcription accuracy for instruments with fewer resources while maintaining high performance for those more abundant in the training data. This is achieved through a tokenization scheme and a flexible output vocabulary inspired by the MIDI specification, enabling the model to represent complex musical compositions accurately.

3.3 Data Augmentation in Deep Neural Networks for Audio Processing

Data augmentation is a pivotal strategy in the training of deep neural networks, especially when the available dataset is limited. By artificially enhancing the dataset's size and diversity, augmentation techniques can mitigate overfitting, thereby improving a model's generalization capability to unseen data. This becomes particularly crucial in scenarios where additional data acquisition is challenging or impossible. Data augmentation has a critical role, especially when training recurrent neural networks with limited data [19]. Data augmentation contributes to the creation of better-generalizing separating networks.

While the concept of data augmentation bolstering the performance of DNNs isn't novel and has been highlighted in various works like [77, 78], particularly in the realm of music information retrieval tasks, the specific techniques employed can vary based on the application and the data's nature. In addition to traditional techniques, our approach also incorporates a novel augmentation method, *OppositePanning*, which specifically addresses the realistic scenario of instrument panning in audio mixtures.

1. **FlipChannels:** This technique involves randomly swapping the left and right audio channels for each instrument. Given the stereo nature of many audio recordings, this technique can introduce variability without distorting the essence of the sound.

2. **Shift:** Randomly shifting channels is another useful method. It helps the model learn invariant representations that aren't overly reliant on specific channel placements, thereby enhancing the robustness of the trained model.

3. **Remix:** A creative technique where, within a single batch, one instrument from a song is interchanged with the same instrument from a different song. This can simulate diverse musical combinations and scenarios, challenging the model to identify and separate instruments even in previously unheard mixes.

4. **Scale:** Randomly scaling the signal with a multiplier, often drawn from a range, for example, [0.5, 1.25], can introduce variations in amplitude. This is instrumental in ensuring that the model doesn't overfit to specific amplitude levels and can generalize well across varied volume levels.

3.4 Monotimbral Source Separation: A Closer Look

Audio source separation, at its core, aims to disentangle individual sound sources from a composite audio mix. This objective can further branch out into several specialized tasks like speech separation, speech enhancement, and the popular music source separation, each with its unique challenges and characteristics.

Differentiating Speech and Music Separation

Sarkar et al. [27] helps elucidate some of the nuances and distinctions within the music separation domain. notably:

1. *Speech Separation Domain*
    - Tasks in this realm often include speech denoising, multi-speaker separation, and dereverberation. The challenges here primarily revolve around handling multiple speakers, background noises, and the intricate sound reverberations in varying environments.
2. *Music Separation Domain:*
    - Historically, music separation research has been predominantly aimed at the demixing challenge, as underscored by the MUSDB dataset's popularity.
    - The demixing challenge mainly concerns separating vocals, bass, and drums from mixed and mastered pop songs. The profound success of deep learning architectures in this area has showcased the feasibility of source separation on a commercial scale.
    - However, the success in this realm has inadvertently overshadowed other vital areas within music source separation. This has led to a narrowed perception where music source separation is mostly equated with the task of separating vocals, drums, and bass stems from mastered tracks.

Venturing into Monotimbral Ensembles

The challenge here is two-fold:

1. **Spectral Overlap:** Unlike the broader music demixing challenge, which involves instruments with distinct spectro-temporal cues (like vocals and drums), monotimbral ensembles frequently comprise instruments that operate within similar frequency ranges. This, coupled with their harmonized nature, results in an intense spectral overlap, making their separation much more complex.
2. **Label Ambiguity:** Often, monotimbral ensembles might contain multiple sources from the same instrument family. This presents an ambiguity in labeling, further complicating the separation task.

   Given these intricacies, separating monotimbral ensembles encapsulates the complexities of both speech and music separation [20, 79]. The overlap in frequency, the harmonically correlated structure, and label ambiguities make this task uniquely challenging.

3.5 The Need for Permutation Invariant Training

In the realm of audio processing, the challenge of source separation is a multifaceted one, especially when the sources in question share similar characteristics. Classical guitar duets can be likened to the complexities faced in speaker separation. Just as two speakers may have closely matched tonal qualities, making their voices challenging to distinguish, two classical guitars can create complex and challenging-to-separate sound patterns. Given this parallel, it becomes evident that the methodologies and techniques honed for speaker separation in the realm of deep learning and audio processing can offer invaluable insights for our research. By delving into the strategies used for speaker separation, we can adapt and refine these methods, tailoring them to the unique nuances of classical guitar duets, and thereby advancing the field of musical source separation.

Speaker-independent multi-talker speech separation presents a significant challenge due to the inherent label ambiguity or permutation problem. Historically, only a handful of deep learning-based studies have ventured into addressing this issue. Weng et al. [79] achieved commendable results by leveraging instantaneous energy to navigate the label ambiguity challenge. Their method employed a two-speaker joint-decoder with a speaker switching penalty. However, this approach is intrinsically tied to the decoder, making it a challenge to scale beyond two speakers. On the other hand, Hershey et al. [80] made strides with the deep clustering (DPCL) technique. This method involved training an embedding for each time-frequency bin to optimize a segmentation criterion. During the evaluation phase, each bin was mapped into an embedding space, and a clustering algorithm was subsequently applied to partition the time-frequency bins. While this method has shown promise, it operates under the assumption that each bin is exclusive to a single speaker. This assumption, although frequently accurate, can sometimes be sub-optimal. Moreover, integrating this method with other techniques, especially those in the complex-domain separation realm, poses challenges [20].

In light of these challenges, a novel training criterion, termed "permutation invariant training" (PIT), has been proposed for speaker-independent multi-talker speech separation. Unlike traditional perspectives that view speech separation as either a multi-class regression or a segmentation problem, PIT approaches it as a genuine separation challenge. The primary strategy of PIT is to first ascertain the optimal output-target assignment and subsequently minimize the error based on this assignment. This method offers a direct solution to the long-standing label permutation problem, which has been a significant roadblock in the evolution of deep learning techniques for speech separation [20]. **Permutation Invariant Training** (PIT) [27] is an approach that we have to consider. It is particularly advantageous when dealing with label ambiguities. As the monotimbral ensembles can have multiple sources from the same instrument family, traditional training might struggle with assigning correct labels due to the inherent similarities among these sources. PIT addresses this by ensuring the model is invariant to different permutations of the output labels, thus reducing the ambiguity in assignments and enhancing the overall separation quality.

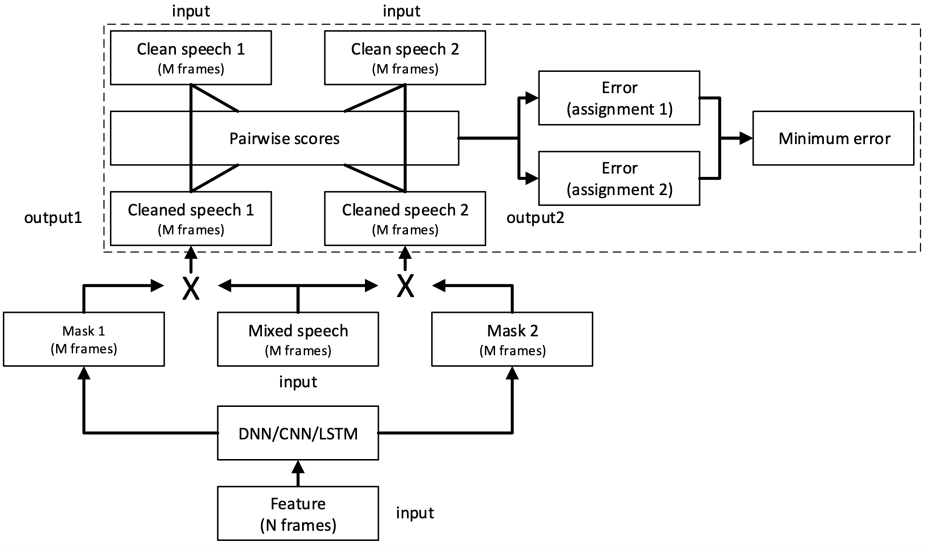


Figure 3.5.1: The two-talker speech separation model with Permutation Invariant Training (PIT) [20].

The proposed solution, as illustrated in Fig. 3.5.1, underscores two pivotal innovations: permutation invariant training (PIT) and segment-based decision making. The model treats reference source streams as a set, rather than an ordered list, ensuring consistent training outcomes irrespective of the source order. During the inference phase, the mixed speech is the sole available data. Speech separation is executed for each input meta-frame, estimating an output meta-frame with multiple frames of speech for each stream. Due to the inherent nature of PIT, the output-to-speaker assignment might fluctuate across frames. However, this can be mitigated and potentially enhanced by overlaying a speaker-tracing algorithm atop the network's output.

In conclusion, while the broader domain of music separation has seen significant advancements, particularly in the demixing challenge, there remains a vast landscape of challenges within monotimbral ensemble separation that demands focused research and innovative techniques like PIT.

3.6 Datasets

As highlighted previously, the evolution of music source separation methods has shifted from traditional digital signal processing (DSP) techniques to fully supervised end-to-end deep neural networks. The availability of well-constructed datasets for training and evaluation is now an indispensable component of this research.

3.6.1 Existing Datasets for Music Source Separation

In response to the demand for more comprehensive and reliable datasets, researchers have developed dedicated collections tailored for music source separation tasks. Prominent examples include MUSDB18-HQ [21], MedleyDB [22], Slakh [23], URMP [24], MIR-1K[25], GuitarSet[26], EnsembleSet[27] each designed with specific objectives and rich annotations.

All the aforementioned datasets can be divided into two primary categories:

Multitimbral Datasets

This subset encompasses sources with pronounced timbral distinctions, typically representing disparate instruments like bass, drums, and vocals. The unique sonic characteristics of each instrument make them discernible from one another.

MedleyDB: MedleyDB dataset was primarily curated to facilitate research on melody extraction and includes 122 songs, totaling approximately 7.17 hours of audio. It offers stereo-format recordings with a sampling rate of 44.100 Hz, accompanying each song with processed stems, raw audio, metadata, melody f0 annotations, instrument activations, and genre information.

DSD100: The DSD100 dataset is featuring 100 full-length tracks split evenly into a training set and a test set, cumulatively offering around 10 hours of audio. Each track is presented in stereo format with a standard sampling rate of 44.100 Hz. Uniquely, the dataset provides the individual stems for vocals, bass, drums, and other accompaniments, allowing for detailed analysis and separation tasks. Additionally, the DSD100 dataset includes comprehensive metadata and mixing information, although it does not provide melody annotations or genre classifications.

MUSDB18-HQ: The MUSDB18 dataset, widely used in music separation research, combines data from various sources, including MedleyDB and DSD100. It comprises 150 songs spanning different musical genres, totaling 10 hours of high-quality audio. MUSDB18 includes full-length stereo recordings and, notably, offers four individual stems (vocals, bass, drums, and "other") for each song to facilitate multi-instrument separation.

Slakh: Synthesized Lakh (Slakh), a multi-track audio dataset, was designed for tasks such as music source separation and multi-instrument transcription. It contains 2.100 tracks, amounting to approximately 145 hours of mixture data. These tracks are synthesized from the Lakh MIDI Dataset v0.1 [81] using professional-grade virtual instruments and aligned MIDI files with 34 instrument classes.

MIR-1K: MIR-1K is a dataset created for singing voice separation, consisting of 1000 song clips. Each clip includes music accompaniment in one channel and singing voice in the other. It offers manual annotations for pitch contours, unvoiced frames, lyrics, and vocal/non-vocal segments. Additionally, the dataset provides speech recordings of the lyrics, and the clips vary in duration from 4 to 13 seconds. In total, MIR-1K spans 133 minutes and is derived from 110 karaoke songs. These songs were chosen from a pool of 5000 Chinese pop songs and performed by 8 females and 11 males, most of whom lack professional music training.

URMP: URMP (University of Rochester Multi-Modal Musical Performance) is a dataset designed for analyzing musical performances from both audio and visual perspectives. It encompasses 44 simple multi-instrumental music pieces, constructed by combining separately recorded performances of individual instrument tracks. Each piece in the dataset includes the musical score in both MIDI and PDF formats, high-quality individual instrument audio recordings in WAV format, and assembled videos in MP4 format. The videos feature a 1080P resolution with a frame rate of 29.97 FPS and arrange the instrument players horizontally, following the score's track order. Additionally, the dataset provides frame-level pitch trajectories and note-level transcriptions for individual tracks in ASCII delimited text format.

EnsembleSet: EnsembleSet is a chamber ensemble dataset crafted to address the challenges in source separation research. This dataset leverages Spitfire Audio's "BBC Symphony Orchestra" sample library to create realistic digital renditions of chamber ensemble scores derived from MIDI transcriptions (RWC Classical Music Database [82]) and lilypond scores (Mutopia) [83].

EnsembleSet comprises a selection of 9 classical pieces, encompassing string quartets, clarinet quintets, piano trios, and piano quintets from the RWC Classical Music Database. Additionally, it includes 71 compositions

from Mutopia, featuring various chamber ensembles with string quartets at the core. The core of this dataset is Spitfire Audio's BBC Symphony Orchestra Sample Library, capturing individual instruments with multiple microphones in a studio setup. It simulates high-quality recordings of chamber ensemble pieces, faithfully reproducing the nuances of real performances.

In summary, EnsembleSet offers 6 hours and 9 minutes of multi-instrument, multi-microphone data, primarily focusing on string ensembles, with additional woodwind and brass instruments. Each song is paired with its corresponding MIDI file, containing essential articulation information.

Monotimbral Datasets

Contrasting the multitimbral subset, this category encapsulates sources with homogeneous or closely related timbral attributes, either emanating from identical instruments or from those with pronounced sonic similarities.

GuitarSet: A comprehensive dataset that offers top-notch guitar recordings paired with extensive annotations and metadata. The unique use of a hexaphonic pickup during guitar recording allows us to capture not only individual string recordings but also significantly streamline the costly annotation process, resulting in rich and detailed annotations.

GuitarSet comprises recordings of diverse musical excerpts performed on an acoustic guitar, complemented by meticulously synchronized annotations. These annotations encompass pitch contours, precise string and fret positions, chord progressions, beats, downbeats, and intricate playing styles.

This dataset features a total of 360 excerpts, each approximately 30 seconds in duration. These 360 excerpts are derived from various combinations, including performances by six different players across 30 lead sheets. Notably, there are two versions for each performance: comping following the sheet and soloing with an improvised performance. Musicians initially record the comping version and later overlay their solo performance on top of their comping. The 30 lead sheets are generated from a wide spectrum of musical styles, including Rock, Singer-Songwriter, Bossa Nova, Jazz, and Funk. Additionally, they encompass three distinct chord progressions: 12 Bar Blues, Autumn Leaves, and Pachelbel Canon, all performed at both slow and fast tempi.

The audio recordings are captured using the hexaphonic pickup technology, delivering separate signals for each guitar string. This innovation enables automated note-level annotation, a valuable feature for researchers and enthusiasts alike. Musicians are provided with lead sheets and accompanying backing tracks that reflect the correct style, complete with a drum kit and bass line. To ensure the highest audio quality, each excerpt is recorded using both the hexaphonic pickup and a Neumann U-87 condenser microphone as a reference. For each of the 360 excerpts, there is an associated file containing 16 annotations. These encompass pitch data, including pitch contours and MIDI note annotations for each of the six strings. Additionally, annotations cover beat positions, tempo, and chord progressions, providing a wealth of information for in-depth music analysis.

Extracting Monotimbral Datasets from Multitimbral: In a multitimbral dataset featuring diverse orchestral instruments, the creation of monotimbral datasets can be achieved by selectively extracting specific instrument families. For instance, in the case of stringed instruments such as violins, violas, and cellos, a monotimbral dataset can be formed by exclusively acquiring these instruments and omitting others from different categories. This focused selection ensures a dataset dominated by a single timbre. Such practices can be applied to datasets like the URMP or the EnsembleSet.

Our endeavor, which focuses on the separation of classical guitar duets, naturally aligns with the monotimbral category. Upon scrutinizing available datasets, "GuitarSet" emerges as the most congruous to our research requirements due to its specific orientation towards guitar soundscapes. Nevertheless, a significant portion of existing datasets is predominantly geared towards scenarios where source signals exhibit notable timbral differences, such as drums, vocals, and bass. In contexts like guitar duets, identifying datasets that capture recordings of homogeneous instruments proves to be an intricate task. The challenge is accentuated when seeking datasets apt for classical guitar separation, considering the instrument's inherently polyphonic character. As of now, a dataset meticulously tailored to polyphonic recordings of identical instruments, especially classical guitars, remains elusive.

3.7 Evaluation Metrics for Source Separation

The field of Blind Audio Source Separation (BASS) [84] has been a focal point of research for many years, producing numerous successful techniques with commendable results. Yet, due to the inherently subjective and complex nature of this task, assessing performance and comparing different methods necessitates the adoption of widely accepted and high-quality evaluation metrics.

Historically, various metrics have been employed, such as Inter-Symbol Interference (ISI) [85] or the Mean Squared Error (MSE) between L2-normalized source signals. While these metrics are relevant, they possess limitations. One of the most critical shortcomings is their treatment of the desired signal s_{bj}, considering it recovered up to permutation and gain, without accounting for other forms of distortion. Moreover, these metrics offer a singular performance value, failing to differentiate between various error sources like sensor noise (e_{noise}), source interferences (e_{interf}), spectral correctness, and the introduction of unrelated artifacts. Such distinction is crucial for accurate technique assessment, especially since different applications may prioritize one type of error over another. Errors in separation tasks can be categorized into three groups: sensor noise (e_{noise}), interference from other sources (e_{interf}), and disruptive artifacts (e_{artif}), with the latter being particularly detrimental. Consequently, a technique may score highly in metrics while still delivering suboptimal perceived performance, depending on the balance of error terms.

To address these challenges, the BSS Eval toolkit [86] has emerged as a valuable resource. Initially developed for MATLAB, it has gained widespread usage within the Python community through the museval package. The toolkit introduces metrics such as the Source to Distortion Ratio (SDR), Source to Interference Ratio (SIR), and Source to Artifacts Ratio (SAR) [84] , which can be configured to accommodate time-invariant filter and gain adjustments, aligning more closely with specific application requirements.

In the computation of these metrics, it is assumed that the estimated source signal s_{bj} can be decomposed into four terms: $s_{\text{bj}} = s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}$. This decomposition relies on orthogonal projections of the source signals onto subspaces defined by the source signals and/or sensor noise. As a result, the metrics are defined as follows:

Source to Distortion Ratio (SDR): Serves as a measure of overall signal separation quality, capturing both the interference and artifacts in the separated signal relative to the desired source. The SDR is defined as:

$$\text{SDR} = 10\log_{10}\frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}\|^2}.$$

Source to Interference Ratio (SIR): Quantifies the clarity of separated sources, specifically focusing on how well other interfering sources have been excluded from the separated signal. The SIR is defined as:

$$\text{SIR} = 10\log_{10}\frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}}\|^2}.$$

Source to Artifacts Ratio (SAR): Indicates the presence of auditory artifacts, assessing the amount of distortion or unwanted sounds introduced during the separation process that are not related to interference. The SAR is defined as:

$$\text{SAR} = 10\log_{10}\frac{\|s_{\text{target}}\|^2}{\|e_{\text{artif}}\|^2}.$$

Source Image to Spatial Distortion Ratio (ISR): Measures the amount of spatial distortion or filtering distortion that the separation algorithm introduces. The ISR is defined as:

$$\text{ISR} = 10\log_{10}\frac{\|e_{\text{interf}}\|^2}{\|e_{\text{spillover}}\|^2}.$$

Among these metrics, SDR is typically considered the most crucial, as it aligns closely with human perception. Despite its widespread adoption, the BSS Eval toolkit has its limitations, particularly concerning allowed distortions and the correlation between metrics and human perception. One concern is that the permissible distortions within the toolkit can alter the reference signal substantially, potentially matching any estimated

signal, leading to issues of objectivity and credibility in evaluating different algorithms. To address this, a scale-invariant version of the metric SDR have been proposed in [87], enhancing and redefining the traditional metrics.

Scale-Invariant Source to Distortion Ratio (SI-SDR): An enhancement of the traditional SDR, SI-SDR is less sensitive to the scaling of the estimated signal, making it a more robust metric for evaluating the true quality of source separation, especially in scenarios where the amplitude or volume of the separated sources might vary. The SI-SDR is defined as:

$$\text{SI-SDR} = 10 \log_{10} \frac{\|\alpha s_{\text{target}}\|^2}{\|\alpha s_{\text{target}} - s_{\text{est}}\|^2},$$

where $\alpha = \arg\min_\alpha |\alpha s - \hat{s}|^2$. The optimal scaling factor for the target is obtained as $\alpha = \frac{\hat{s}^T s}{\|s\|^2}$, and s_{est} is the estimated source signal.

In summary, the evaluation of source separation methods relies on robust metrics like SDR, SIR,ISR and SAR, offered by the BSS Eval toolkit. However, ongoing efforts are essential to address potential limitations and better align these metrics with human perception, ensuring more accurate and reliable evaluations of source separation algorithms. For this specific research we are focusing on the SDR and SI-SDR while at the same time calculating the rest of the metrics.

Chapter 4

Creating Datasets for Monotimbral Source Separation

Contents

Given the lack of datasets addressing the needs of monotimbral source separation, particularly for scenarios involving classical guitars or similar instruments, our research aims to bridge this gap. We intend to embark on the creation of dedicated datasets that encompasse polyphonic recordings of the same instrument, facilitating the development and evaluation of source separation techniques for such challenging and musically rich contexts. In doing so, we aim to provide a valuable resource for researchers and practitioners in the field of music source separation.

4.1 Dataset Creation: GuitarDuets

In this section, we present the process of creating a unique and authentic dataset for monotimbral music source separation by recording two classical guitars performing together. This dataset harnesses the interaction and interplay between real guitarists, capturing the intricacies of live performances. The recording process involved meticulous setup, performance considerations, and data collection to ensure a dataset for further analysis and training of our source separation models.

Recording Setup

The recording process began with the setup of a suitable recording environment to achieve optimal audio quality. We selected a quiet and acoustically treated room to minimize background noise and undesirable reverberations. Each classical guitar was positioned using high-quality condenser microphones (Presonus PM-2) to capture the respective sounds. During the recording process, four different classical guitars were used to ensure a diverse range of timbres in our dataset. Despite meticulous placement and adjustments, we encountered a challenge with microphone leakage, where the sound from one guitar's performance inadvertently bled into the microphone capturing the other guitar. This unintended crossover of sound presented a significant concern, as it could potentially compromise the isolation of the individual guitar tracks, impacting the quality and accuracy of the source separation dataset. In addressing the issue of source bleeding in microphones, we recorded a specialized test set that is free from such leakage. This set, consisting of seven tracks, was created to ensure the absence of cross-feed between microphones. We leveraged this dataset for the model's testing and validation phases to provide an accurate assessment of its separation capabilities.

Performance Considerations and Data Collection

Creating a comprehensive and expressive dataset required careful planning of the guitar duet performances. We curated a selection of diverse classical music pieces, representing various styles, tempos, and complexities. The guitarists practiced extensively to ensure synchronized performances while allowing for artistic interpretation and expressive dynamics. Emphasis was placed on capturing the inherent interactions between the two instruments, such as harmonies, counterpoint, and complementary melodies.

During the recording sessions, we captured multiple takes of each piece to introduce variability in the dataset. This allowed us to incorporate different interpretations and slight variations in performances, akin to the natural imperfections present in live music. We ensured consistency in data collection by maintaining the same recording setup and conditions for all pieces.

Preprocessing

Before integrating the recorded data into the source separation pipeline, we performed essential preprocessing steps to ensure data consistency and compatibility. This involved normalization of audio levels, removal of any unwanted artifacts, and segmenting the recordings into appropriate samples for the source separation task. Special care was taken to preserve the natural dynamics and timbres of the guitars during preprocessing. We exported each individual guitar performance as a 44.100 Hz 16-bit WAV file in both stereo and mono formats. Subsequently, to create the mixed audio files, we simply summed the two guitar files representing each duet.

In the following table there is an overall overview of the tracks constituting the dataset recording along with their duration.

Figure 4.1.1: Marios and Orpheas playing at a concert.

Recording Name	**Duration (seconds)**
Valses Poeticos 1 (Enrique Granados)	85.5
Valses Poeticos 1 2nd	97.5
Valses Poeticos 2	92.0
Valses Poeticos 2 2nd	25.0
Valses Poeticos 2 3rd	49.5
Valses Poeticos 3	88.5
Valses Poeticos 3 2nd	86.5
Valses Poeticos 4	116.0
Valses Poeticos 5	46.5
Valses Poeticos 5 2nd	46.0
Valses Poeticos 6	60.0
Valses Poeticos 7	87.0
Valses Poeticos 8	49.0
Valses Poeticos 9	91.0
Valses Poeticos 10	91.5
Summer Garden Suite 1 Opening (Sergio Assad)	82.0
Summer Garden Suite 1 Opening 2nd	73.0
Summer Garden Suite 2 Summer Garden	156.0
Summer Garden Suite 2 Summer Garden 2nd	142.0
Summer Garden Suite 3 Farewell	175.0
Summer Garden Suite 3 Farewell 2nd	168.0
Summer Garden Suite 4 Butterflies	175.0
Tango 1 (Astor Piazzolla)	338.0
Tango 1 2nd	331.0
Tango 2	297.5
Tango (N. Mavroudis)	154.0
demo1	34.5
demo2	66.0
demo3	49.5
demo4	27.0
demo5	41.5
demo6	22.5
demo7	27.0
demo8	47.5
Total Duration	**58.6 minutes**

Table 4.1: Dataset Recordings and Durations

Conclusion

The dataset created through the recording of two classical guitars playing together offers an authentic and expressive representation of real-world guitar duets. Unlike synthesized or MIDI-based datasets, this collection captures the subtleties and nuances of live performances, providing a rich resource for training and evaluating music source separation models. The combination of meticulous recording setup, thoughtful performance considerations, and careful data collection ensures that the dataset reflects the artistry and complexities of classical guitar duets. It is important to note that during the recording process, we encountered a microphone bleeding issue which may pose challenges and should be taken into consideration for any future research or experiments utilizing this dataset.

4.2 Dataset Creation for Music Source Separation: Leveraging Native Instruments Plugin and MIDI Scores

In this section, we present a comprehensive account of our dataset creation process, where we combine the expressive power of Native Instruments' SESSION GUITARIST - PICKED NYLON plugin [29], which constitutes a virtual instrument generating classical guitar sounds, and the musical notation data from MIDI scores sourced from the MuseScore community [30]. As discussed in the preceding section, the dataset comprising real recordings is insufficient in duration to constitute a comprehensive training set for neural network models. Consequently, we are developing this synthetic dataset. The generation of synthetic data is not only more efficient and less time consuming but also serves to augment our initial dataset. Moreover, this synthetic compilation allows us to conduct comparative analyses between real and synthetic datasets, thereby evaluating the model's ability to generalize across these two distinct domains.

Introduction to Virtual Instruments

A virtual instrument, in the context of music production and digital audio technology, refers to a software-based emulation of traditional musical instruments or synthesizers. Unlike physical musical instruments, which require physical interaction and acoustic sound generation, virtual instruments exist entirely within the digital realm, residing as software plugins within Digital Audio Workstations (DAWs).

Throughout history, the development of virtual instruments has been shaped by technological advancements. The early efforts in the 1980s to utilize MIDI for computer-music communication laid the foundation for virtual instrument development. As computers gained processing power and storage capabilities, developers began creating basic software synthesizers, marking the initial steps in the world of virtual instruments. The 1990s saw significant progress with sample-based technology, resulting in virtual instrument plugins that could convincingly emulate acoustic instruments. The 2000s witnessed a paradigm shift with the rise of Digital Audio Workstations (DAWs) and standardized plugin formats, fueling the proliferation of high-quality virtual instruments that continue to redefine modern music production [88].

Today, virtual instruments play a vital role in the music-making process, democratizing access to an extensive array of sounds and unleashing creativity in music production. The seamless integration of virtual instruments within modern DAW workflows has revolutionized the way artists craft their compositions, enabling them to explore diverse timbres, styles, and sonic landscapes, all within the digital domain. Whether through sampled-based or synthesis-based technologies, virtual instruments have become indispensable tools, empowering musicians, composers, and producers to express their artistic visions with unprecedented flexibility and versatility.

There are two primary types of virtual instruments: **sample-based** and **synthesis-based**. **Sample-based** virtual instruments rely on extensive collections of recorded audio samples from real instruments. These high-fidelity samples capture the nuances and sonic characteristics of acoustic instruments, such as pianos, guitars, strings, and brass. When triggered, these samples play back the recorded sounds, producing realistic and expressive tones. Sample-based virtual instruments offer a vast selection of instruments, articulations, and playing styles, allowing musicians to access a wide range of lifelike sounds without the need for physical instruments or dedicated hardware.

On the other hand, **synthesis-based** virtual instruments generate sounds algorithmically, simulating the behavior of synthesizers and electronic instruments [28]. Through various synthesis techniques, such as sub-

tractive synthesis, additive synthesis, FM synthesis, and wavetable synthesis [89], these virtual instruments can create an extensive array of electronic sounds, pads, leads, and textures. Synthesis-based virtual instruments provide artists with the flexibility to craft entirely original and imaginative sounds, unleashing boundless creative possibilities.

Native Instruments Plugin: "Session Guitarist - Picked Nylon"

The Native Instruments plugin, "Session Guitarist - Picked Nylon" [29], is a sample-based virtual instrument meticulously designed to capture the nuances of a nylon-stringed guitar. Its clean and mellow timbres make it an ideal production tool for a wide range of music genres, from classical to contemporary styles like bossa nova and flamenco. The plugin's remarkable flexibility and top-tier performance controls enable artists to achieve their desired sounds effortlessly. To create this plugin, master guitar-maker Lisa Weinzierl crafted the instrument using rare preamps and a carefully controlled studio environment, ensuring a realistic tone [29]. The plugin offers a rich library of strummed chords, picked arpeggios, and delicate riffs, performed with various playing styles by a top-class session player.

Figure 4.2.1: Native Instruments Plugin: "Session Guitarist - Picked Nylon" [29].

MIDI Scores from the MuseScore Community

In the realm of music production and digital audio technology, music representations play an important role in capturing musical data in digital format. A music representation, also known as a music file format, is a standardized way to store musical information, such as pitch, duration, tempo, and dynamics, in a digital file. These representations allow music to be seamlessly transferred, edited, and reproduced across various digital platforms and software applications.

Over the years, numerous music representations have emerged, each tailored to specific purposes and requirements. Some popular, audio based formats include WAV (Waveform Audio File Format), AIFF (Audio Interchange File Format), FLAC (Free Lossless Audio Codec), and MP3 (MPEG Audio Layer III), among others. These formats vary in terms of audio quality, compression techniques, and storage efficiency.

Among the various music representations, MIDI (Musical Instrument Digital Interface) stands out as one of the most widely used and versatile formats. Unlike audio-based formats, MIDI files do not contain actual audio waveforms; instead as stated in Sec. 2.2.3, they store musical instructions and data that represent the performance of instruments and their interactions within a composition. MIDI files are lightweight, highly editable, and do not suffer from audio degradation due to compression, making them ideal for music composition, arranging, and sequencing.

The flexibility of MIDI allows composers and producers to manipulate individual musical elements, such as notes, velocities, and articulations, with precision and ease. MIDI data can be easily edited and reorganized, making them a powerful tool for composing and arranging complex musical pieces. Furthermore, MIDI's ability to separate musical data from sound generation allows users to apply different virtual instruments or synthesizers to interpret the MIDI data, yielding diverse and customizable sound options.

Due to its widespread compatibility and efficiency, MIDI has become the industry standard for various musical applications, including music production, video game soundtracks, film scoring, and live performances.

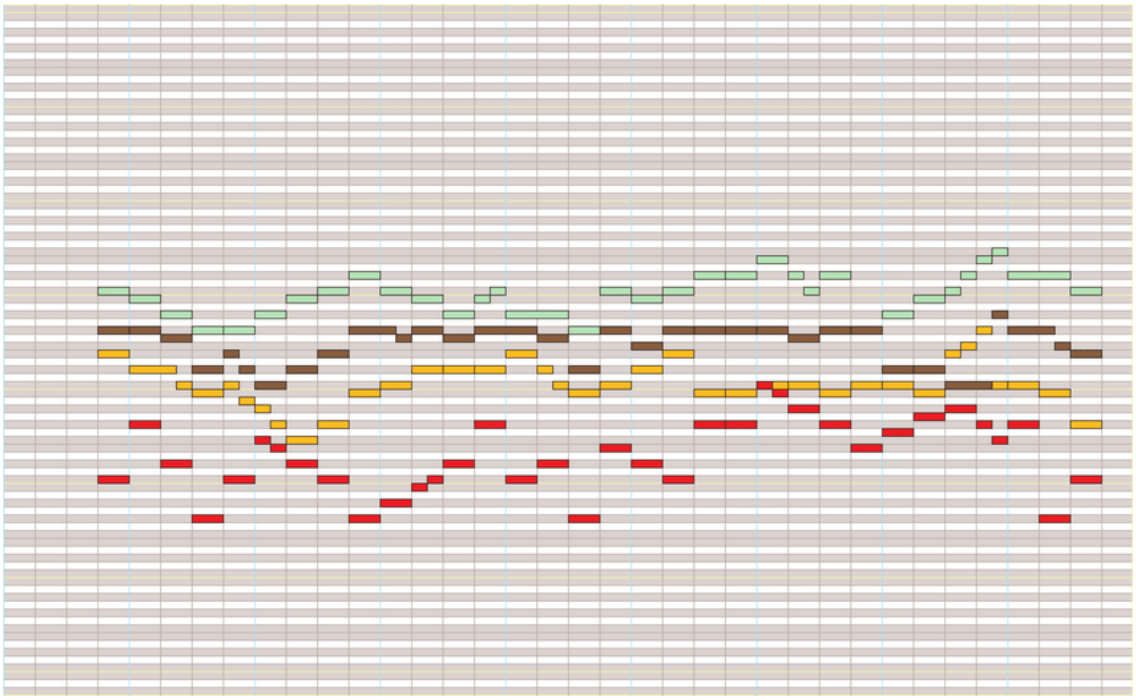


Figure 4.2.2: Piano roll of a MIDI file [90].

MIDI's ubiquity in digital music production workflows highlights its adaptability and universal appeal to musicians, composers, and producers worldwide.

The MuseScore community [30] is an invaluable resource for musicians and researchers alike. It hosts an extensive collection of guitar duet MIDI scores, presenting musical notation and performance data. MIDI (Musical Instrument Digital Interface) files encode musical events, such as pitch, duration, and velocity, making them an excellent representation of the musical score. From this vast repository, we handpicked a selection of guitar duet MIDI scores, ensuring a diverse and representative set of classical music pieces.

Digital Audio Workstation (DAW): Logic Pro X

To transform the MIDI scores into realistic guitar performances, we used Logic Pro X, a sophisticated Digital Audio Workstation (DAW) trusted by musicians and producers worldwide. Logic Pro X [31] provides a professional environment for music production, enabling seamless integration with virtual instruments and audio processing tools. By importing the MIDI files into Logic Pro X, we gained access to a rich array of creative possibilities for shaping the sound of the virtual instruments.

MIDI-Based Sound Creation

Within Logic Pro X, we loaded the SESSION GUITARIST - PICKED NYLON plugin three times, creating three distinct instances, each with a unique timbral setting. This configuration allowed us to simulate three separate guitars, each contributing a distinct sound to the duet ensemble. With this meticulous setup, we ensured a wide range of timbres for each guitar part in the duet, closely mirroring real-world scenarios where guitarists select different instruments to achieve specific tones and styles.

For each MIDI file, we selected two out of the three distinct timbral settings from the plugin, creating a *diverse combination* of guitar duets. This careful selection process resulted in an expansive dataset. The resulting duets showcased an array of musical styles, dynamic performances, and harmonious interplay between the two guitars.

To finalize the dataset, we exported each individual guitar performance as a 44.100 Hz 16-bit WAV file in both stereo and mono formats. Subsequently, to create the mixed audio files, we simply summed the two guitar files representing each duet.

In the following table there is an overall overview of the tracks consisting the dataset recording along with their duration.

Track Number	**Track Name**	**Duration (seconds)**
Track1	Bach, Minuet in G major, BWV Anh 114	120.0
Track2	Bach Prelude n3 BWV 935	120.0
Track3	Blind Guardian The Bard's Song	45.818
Track4	Unkoown (No named Provided by MuseScore)	32.0
Track5	Unkoown (No named Provided by MuseScore)	36.0
Track6	Unkoown (No named Provided by MuseScore)	38.571
Track7	Marcello/Bach - Concerto in D minor	109.5
Track8	Duo en Sol op.27 n°8 - Ferdinando Carulli	57.6
Track9	BWV 304 Bach. J.S. Choral; Eins ist noth, ach Herr, dies Eine	93.103
Track11	Sir Edward Elgar - Pomp and Circumstance March No.1	374.4
Track12	Sibelius Etude Op.76 No.2	192.0
Track13	The Police - Every Breath You Take	214.839
Track14	Jordon Drumgoole - Four Short Seasons for Guitar Duet	300.632
Track15	Gerald Schwertberger - Blue and Rythmic Duets	577.92
Track16	Unkoown (No named Provided by MuseScore)	500.909
Track17	J.O.Marques: Six Easy Duets for Guitars - No.1 in C major	56.048
Track18	J.O.Marques: Six Easy Duets for Guitars - No.2 in G major	92.857
Track19	J.O.Marques: Six Easy Duets for Guitars - No.4 in F major	106.667
Track20	J.O.Marques: Six Easy Duets for Guitars - No.6 in C major	89.302
Track21	Suite c minor (BWV997) - Preludio for tenor	154.884
Track22	Mazurka - Francesco Tarrega (1852 - 1909) - Duo	63.717
Track23	Milonga Guitar Duo	105.366
Track24	NIGHTWISH - Ever Dream	83.137
Track25	Poco Allegretto - Ferdinando Carulli (1770 - 1841) - Duo	94.815
Track26	Recuerdos de la Alhambra - Francisco Tárrega	240.0
Track27	Scherzino Mexicano	141.639
Track28	Unkoown (No named Provided by MuseScore)	129.836
Track29	Unknown	63.066
Track30	Unknown	57.81
Track31	Unknown	151.579
Track32	Unknown	122.553
Track33	Terpsichore - Duo op.45 - José Ferrer y Esteve	208.0
Track34	Louis Moreau Gottschalk - The Dying Poet	261.818
Track35	Ferdinando Carulli Trois Noctures op.90	727.183
Track36	Unkoown (No named Provided by MuseScore)	604.337
Total Duration		**106 minutes**

Table 4.2: NI Dataset and Durations

Below in Fig. 4.2.3, we present spectrograms of both a real guitar recording from our dataset, and a synthetic guitar counterpart. Examination of these spectrograms reveals that the spectrogram on the right exhibits increased noise levels which can be attributed to microphone-related noises and the phenomenon of instrument source leakage.

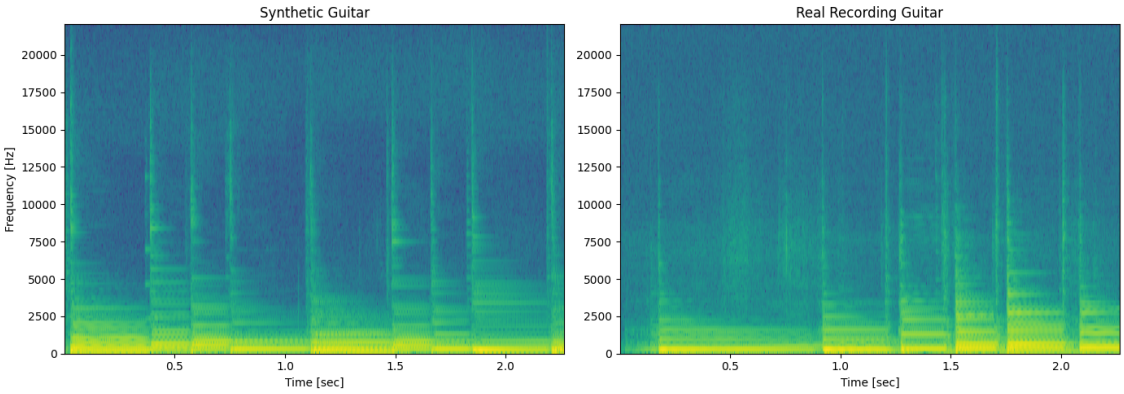


Figure 4.2.3: Comparison between synthetic (left) and real (right) data.

Chapter 5

Residual Shuffle-Exchange Music Transcription Network And Experiments

Contents

5.1 Overview of the Residual Shuffle-Exchange Network

5.1.1 Benes Network Foundation

The Residual Shuffle-Exchange Network (RSE) evolves from the foundational structure known as the Benes network. This network is a key component within the RSE architecture and serves as a pivotal mechanism for facilitating the process of shuffling and exchanging features within a deep neural network. The Benes network is renowned for its ability to efficiently handle permutations of inputs, which is a fundamental operation in tasks such as sorting and routing networks.

In the context of the RSE, the Benes network underpins the model's ability to manage and learn from the high-dimensional data involved in music transcription. It comprises two Benes blocks, which are critical for the network's capacity to develop rich representations of the input data. These blocks are arranged in a manner that allows for a deep and complex transformation of features, setting the stage for the subsequent application of the Residual Switch Unit (RSU).

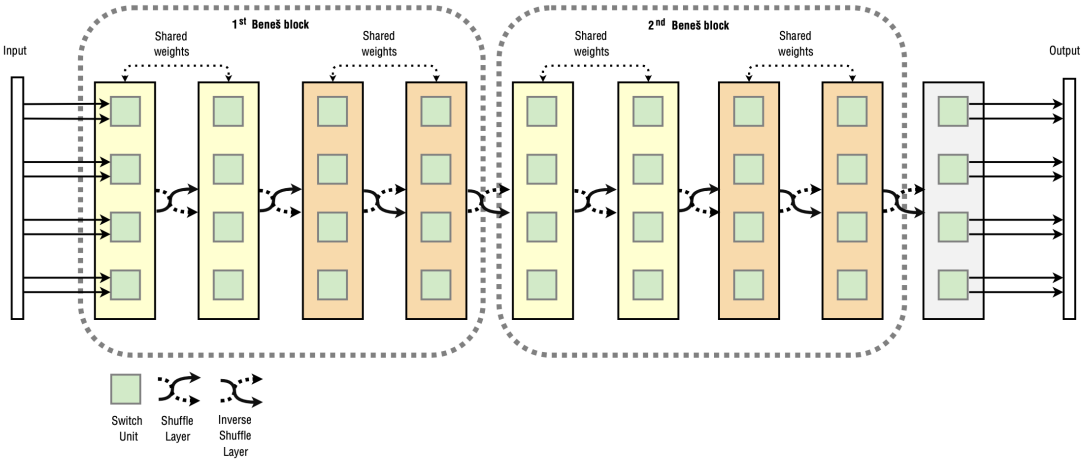


Figure 5.1.1: Residual Shuffle-Exchange network with two Benes blocks and eight inputs.[12]

By replacing the standard Switch Unit with the RSU, the RSE network leverages the Benes network's inherent strengths while enhancing its ability to learn and generalize from audio data. This modification is instrumental in achieving the feature processing necessary for long sequences. The subsequent section on the RSU will delve into the specifics of this replacement and its implications for the network's performance.

5.1.2 Residual Switch Unit (RSU)

The RSU is the cornerstone of the RSE network, distinguishing it from its predecessors. It is constructed on a residual network basis, incorporating Gaussian Error Linear Units (GELU) and Layer Normalization. The RSU design is inspired by the feed-forward block seen in the Transformer architecture. It processes pairs of input vectors, producing output vectors of corresponding size, with each vector representing the number of feature maps in the network. The RSU's linear transformations, followed by Layer Normalization and GELU activation, ensure effective feature processing and stability in training deep networks [12].

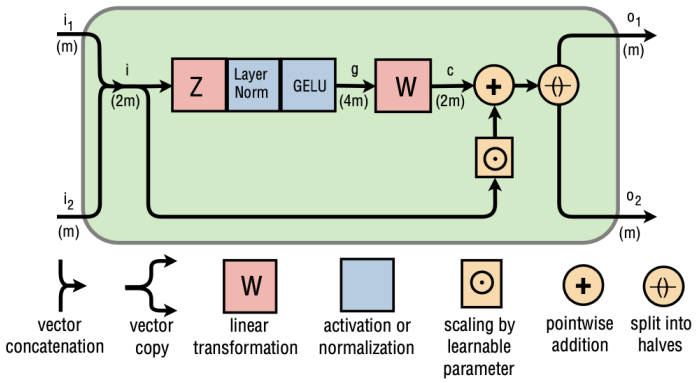


Figure 5.1.2: Residual Switch Unit. A number of feature maps (m) is shown in parentheses. Depicted here with the default of hidden layer being 2× larger than the input (4m being the size of the hidden layer and 2m the size of the input) [12].

Mathematical Formulation

The RSU's formulation involves multiple components as shown in eq. 5.1.1:

- Two linear transformations applied to the feature dimensions.
- Layer Normalization and GELU activation.
- A final output calculation using sigmoid and scalar multiplication operations.

$$\begin{aligned} \mathbf{i} &= [\mathbf{i}_1, \mathbf{i}_2] \\ \mathbf{g} &= \text{GELU}(\text{LayerNorm}(\mathbf{Zi})) \\ \mathbf{c} &= \mathbf{Wg} + \mathbf{B} \\ [\mathbf{o}_1, \mathbf{o}_2] &= \sigma(\mathbf{S}) \odot \mathbf{i} + \mathbf{h} \odot \mathbf{c} \end{aligned} \tag{5.1.1}$$

This design enables a more straightforward architecture compared to the gated mechanisms in previous models, facilitating easier training and potentially more robust performance in various applications [12].

5.1.3 Incorporation of Strided Convolutions

For tasks with a significant discrepancy in information content, such as the MusicNet dataset provides because of the polyphonic nature of the instruments included and the multple notes played at the same time, the RSE network utilizes strided convolutions. These convolutions are applied before the main network, serving to increase the number of feature maps while reducing the sequence length. This approach not only aligns the information content more appropriately but also speeds up the processing without sacrificing accuracy [12].

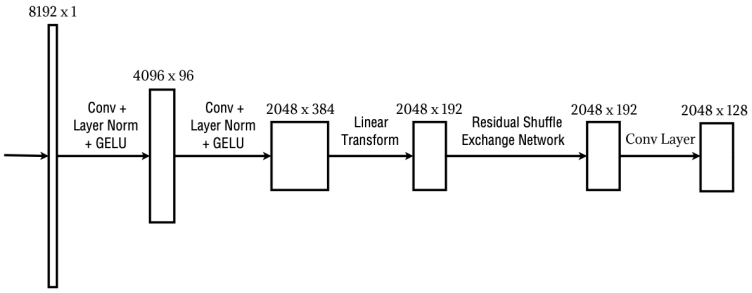


Figure 5.1.3: The architecture with two prepended convolutions employed for the MusicNet Dataset.[12]

5.2 Music Transcription Application of RSE Network

The RSE network's implementation in TensorFlow has demonstrated its adaptability and effectiveness in various tasks. The network architecture has been fine-tuned to optimize performance without leading to overfitting, showcasing its robustness [12].

5.2.1 Performance on Algorithmic Tasks

In algorithmic tasks, where a small change in input can significantly alter the output, the RSE network has shown promising results. It has been evaluated on tasks like long binary addition, multiplication, and sorting. These tasks are benchmarks for assessing a model's capacity to develop and manage long-term dependencies [12].

5.2.2 MusicNet Dataset Performance

A notable application of the RSE network is in the MusicNet Dataset [91], which involves the classification of notes played at each time step in a waveform. The network has shown remarkable performance in this multi-label classification task, achieving an Average Precision Score (APS) of 78.02% on a certain window size. This performance highlights the RSE network's capability in handling complex, long-sequence data [12].

Compared to other architectures listed on the MusicNet leaderboard at Papers With Code [92], the RSE network not only outperforms competitors in APS but also does so with a substantially smaller model size. For instance, the Complex Transformer, which follows the RSE network in performance, has an APS of 74.22% but requires nearly four times as many parameters (11.61M compared to RSE's 3.06M). This efficiency highlights the RSE network's advantage, which can deliver top-tier performance in music transcription tasks without the computational expense typically associated with larger models.

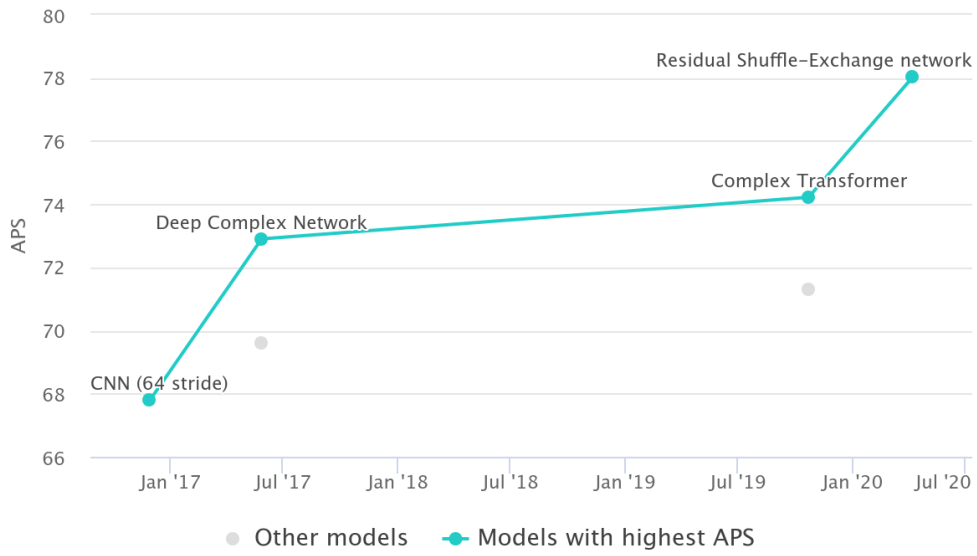


Figure 5.2.1: Papers With Code Leaderboard in the task of Music Transcription on the MusicNet Dataset with Average Precision Score [92].

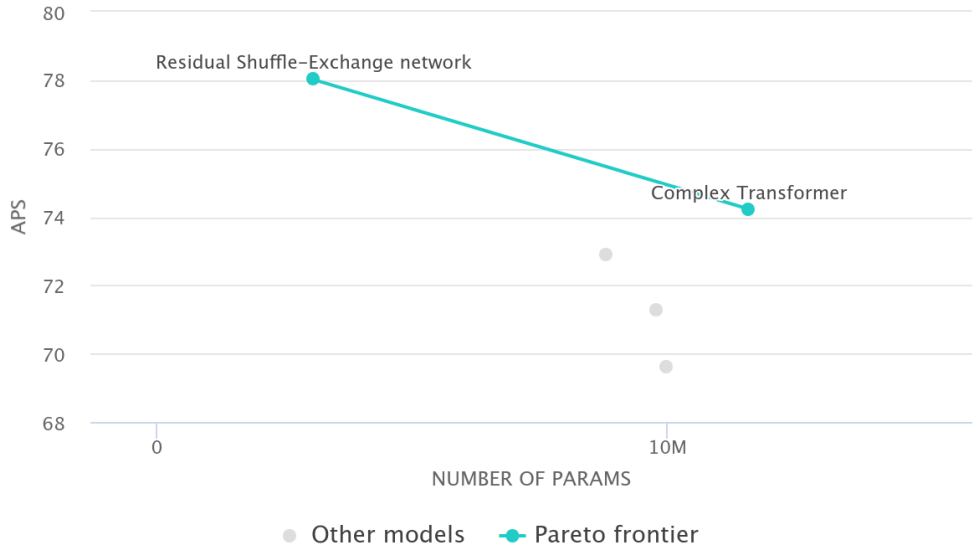


Figure 5.2.2: Papers With Code Chart in the task of Music Transcription on the MusicNet Dataset APS to number of parameters [92].

The above Fig. 5.2.2 of the RSE Network and the Complex Transformer illustrating the parameter footprint in comparison to the performance, underlines the RSE network's superior capability in translating complex auditory information into accurate transcriptions with fewer parameters. This efficiency is critical for practical applications where computational resources may be limited.

5.2.3 MusicNet Dataset Overview

MusicNet is a comprehensive dataset consisting of hundreds of freely-licensed classical music recordings. These recordings are composed by 10 composers and involve 11 different instruments, offering a diverse range of classical music pieces. The dataset is enriched with detailed instrument recorded at a sample rate of 44.100

Hz and note annotations given in the format of csv files containing the notes and the corresponding samples of the audio that the note is being played, amounting to over 1 million temporal labels across all tracks. It encompasses 34 hours of chamber music performances, recorded under various studio and microphone conditions [91].

5.2.4 Evaluation Metric for Music Transcription

Identification of notes in an audio segment is formulated as a multi-label classification problem. We assign each audio segment a binary label vector $\mathbf{y} \in \{0,1\}^{128}$, where the 128 dimensions correspond to the MIDI note numbers. A value of $y_n = 1$ indicates the presence of note n at the midpoint of the audio segment $\mathbf{x}$.

Let $f : \mathcal{X} \to \mathcal{H}$ be a function that maps the audio segment to a feature map. We train a model to predict a label vector $\hat{\mathbf{y}}$ given the feature map $f(\mathbf{x})$, optimized for mean square loss. The prediction $\hat{\mathbf{y}}$ can be interpreted as a multi-label estimate of the notes in $\mathbf{x}$ by choosing a threshold c and predicting label n if $\hat{y}_n \geq c$.

Precision, Recall and Average Precision Score

Music transcription models are evaluated using three metrics: precision, recall as stated in Ch. 2, and average precision score (APS).

These metrics are parameterized by the note prediction threshold c, which is varied to construct precision-recall curves. The APS is then calculated as the area under these curves:

$$\text{APS} = \int_0^1 \text{Precision}(r)\, dr \tag{5.2.1}$$

Here, r is the recall rate, and $\text{Precision}(r)$ is the precision as a function of recall. This score provides a single-figure measure of the model's overall performance across all threshold settings.

5.3 Modifications to Existing Architecture

Our approach transforms the single-output music transcription architecture of the Residual Shuffle-Exchange Network (RSE) into a dual-output system. This system is adept at generating separate transcriptions for each guitar within a mixed audio input, tailored specifically to unravel the complexities of audio separation and transcription from two guitars. A key rationale behind opting for a transcription-focused architecture over a traditional separation model lies in the inherent advantages of transcription systems in handling distinct notes. Given that transcription architectures have already demonstrated commendable performance in delineating individual notes, our strategy leverages this strength for enhanced separation efficiency. By accurately transcribing all notes present in the audio mix, our model aims to simplify the task of attributing each note to the correct guitar. The goal is for the model to learn and understand the natural correlations and mutual exclusivities among guitar notes. For example, the model can learn how playing a certain note on one guitar can often preclude the simultaneous playing of certain other notes on the same instrument. We believe that achieving such a goal and uncovering these correlations is more challenging for a separation model focused solely on distinguishing sounds. In contrast, a transcription-based model, which outputs the notes being played as a binary vector $\mathbf{y} \in \{0,1\}^{128}$ and is designed with features targeting this specific objective, offers a clearer pathway to understanding these correlations. The use of vectors in the transcription model provides a more intuitive and direct means to analyze and grasp the intricate relationships between simultaneous notes on a guitar.

We have tried two different approaches in modifying the architecture:

5.3.1 Modification 1

In the first modification, we adapted the initial architecture to enhance its ability to differentiate between the two instruments playing simultaneously. The architecture retains the foundational structure of Residual

Shuffle Exchange Network dedicated to MusicNet Dataset implementation but introduces a bifurcated pathway after the linear transformation layer. Each path is dedicated to one of the two instruments, integrating a separate Residual Shuffle Exchange Network for each. This split allows the model to focus on the unique characteristics of each instrument, leveraging the potential of the Residual Shuffle Exchange Network to capture and transcribe the subtleties of each instrument's contribution to the audio mix. The final stage in each pathway consists of a convolution layer that is specifically fine-tuned to condense the rich feature maps into a 128-dimensional output. This dimensionality corresponds to the 128 possible MIDI notes, thereby enabling the model to capture the full spectrum of notes each instrument can produce.

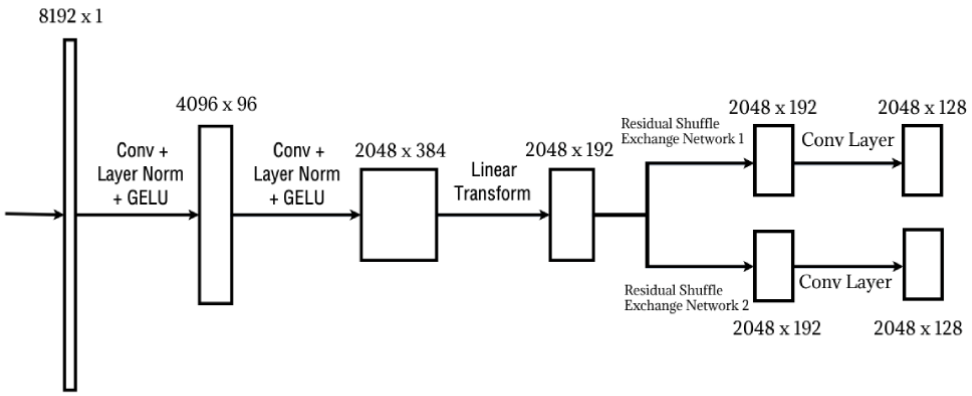


Figure 5.3.1: Modification showing the two different RSEs used.

5.3.2 Modification 2

In the second modification, we implemented a simpler adaptation of the initial Residual Shuffle Exchange Network. The core of the network, including the Residual Shuffle Exchange component, remains unchanged to preserve the integrity of feature extraction performed by the original design. However, an alteration has been made at the concluding stage of the model, the last convolutional layer has been reconfigured to output twice the original dimensionality. Instead of creating two separate pathways for each instrument, this architecture entrusts a single, convolution layer with the responsibility of segregating the features pertinent to each instrument from the shared feature set produced by the RSE. This convolutional layer is adept at assigning the extracted features to the appropriate instrument, outputting a 2x128-dimensional representation that encapsulates the probability distribution for all 128 MIDI notes across both instruments.

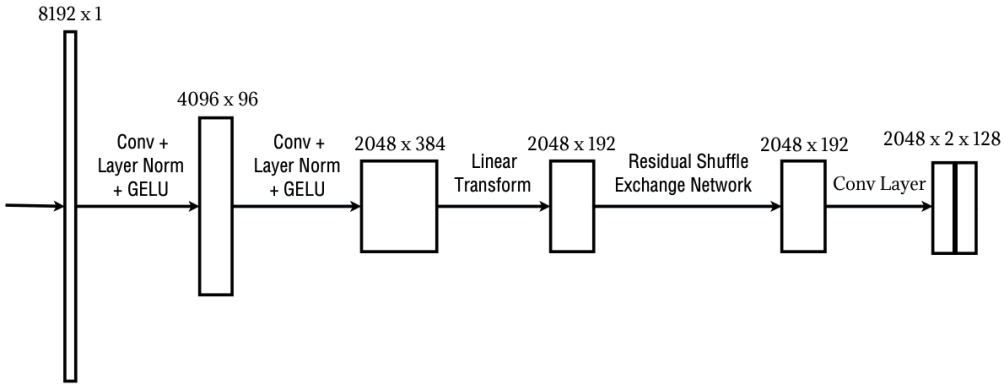


Figure 5.3.2: Modification 2 showing the alteration in the last convolution layer.

5.3.3 Experiments and Results

In this section, we delve into the experimental outcomes, where we applied the model configuration as outlined in the original paper. The RSE architecture processes the audio signal by transforming the waveform into its frequency representation. For any given input audio with a sampling rate of 11.000 Hz, the goal is to identify the musical notes it contains. The method begins by segmenting the audio into frames of 8.192 samples each. Within these frames, it computes the Fast Fourier Transform (FFT) to convert the time-domain signal into frequency-domain information. This step is followed by creating a sequence comprising 4.096 samples of the real parts and 4.096 samples of the imaginary parts of the FFT result. Utilizing these features, the network performs inference on the central portion of the initial 8.192-sample frame to predict the notes present. The analysis advances across the audio by moving the frame by a hop size of 128 samples, repeating this process for subsequent segments. The network's output is a 128-element vector, representing the probabilities of each of the 128 MIDI notes being played. To derive a final, actionable prediction, these probabilities are subjected to a thresholding process to binarize the outcomes, indicating the presence or absence of each note.

We employed a permutation invariant training algorithm within the TensorFlow framework. Our training leveraged two datasets: the widely recognized GuitarSet and a custom dataset comprising native instruments.

To align with the MusicNet dataset's format, the labels from the GuitarSet (.jams files) were transformed into CSV format. Similarly, MIDI files from our native instrument dataset were converted to the same format, ensuring uniformity in data processing. All audio files, originally sampled at 44.100 Hz, were resampled to 11.000 Hz and converted to mono for consistency and computational efficiency.

The datasets were split in a ratio of 0.8 for training, 0.1 for validation, and 0.1 for testing. This partitioning was crucial to evaluate the model's performance comprehensively across various scenarios, ensuring a robust and well-generalized learning process.

Ablation Study on PIT Loss and Modifications

Table 5.1: Ablation Study

APS SCORE	GuitarSet	NI Dataset Test
Modification 1 no PIT	10.06%	59.03%
Modification 1 PIT	13.14%	62.99%
Modification 2 no PIT	12.37%	60.58%
Modification 2 PIT	13.29%	63.97%

The initial experiment was designed not only to compare the performance and efficiency between two architectural modifications but also to test whether the integration of permutation invariant training (PIT) loss contributes to improving the model's effectiveness. While both modifications yielded similar accuracy, the second modification demonstrated a slight edge. Notably, it not only surpassed the first in accuracy but also exhibited enhanced efficiency, attributed to its reduced parameter count. Unlike the first modification, which doubled the parameters by duplicating the residual shuffle exchange network, the second modification simply altered the last convolution layer to produce the desired output. This strategic change resulted in significantly faster inference and training times. Consequently, for subsequent experiments, the second modification will be adopted as the preferred architecture. Moreover, the inclusion of PIT loss indeed enhanced the model, contributing to its overall performance in both modifications. Consequently, for subsequent experiments, the integration of PIT loss will be adopted.

From this experiment, it is inferred that incorporating a linear convolution at the end of the network is effective in capturing and distinguishing individual notes played by each guitar. This suggests that linear convolution is a potent mechanism for the in-depth analysis required in music transcription.

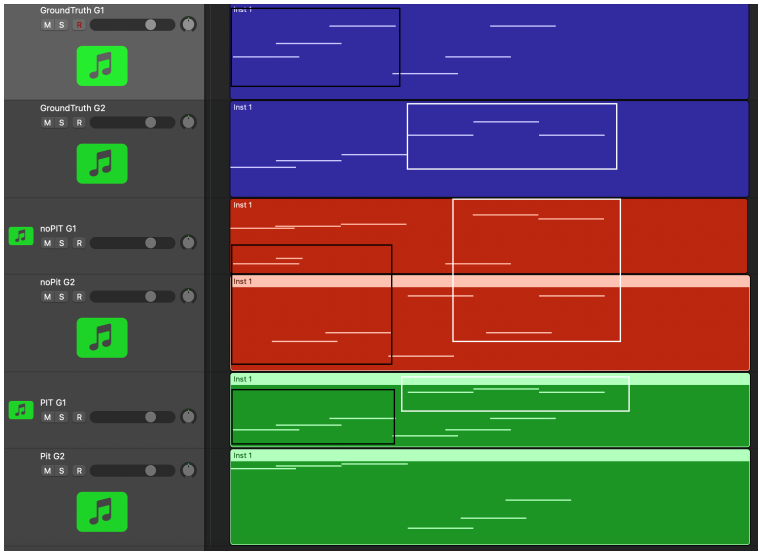


Figure 5.3.3: PIT Training difference with no PIT train. Figure is showing gorundtruth scores (Blue), Estimated Scores with no PIT (Red), Estimated Scores with PIT (Green)

In the above Figure 5.3.3, we observe the ground truth piano rolls for each guitar depicted in blue, alongside the predictions from two algorithms: one that did not utilize Permutation Invariant Training (PIT), shown in red, and one that did employ PIT. The parts enclosed within the black and white boxes represent specific sections from each guitar. It is evident that the model without PIT struggled to consistently assign melody notes to the correct guitar, switching notes between guitars mid-phrase. This issue largely stems from the fact that without PIT, the computed loss for weight optimization is higher and less representative of the desired outcome.

Conversely, the predictions made by the algorithm incorporating PIT demonstrate a marked improvement in maintaining the continuity of a musical phrase within the same guitar. This indicates that PIT was effective in training the model to better understand and preserve musical phrases, assigning them accurately to the appropriate guitar throughout a piece. This distinction underscores the significance of PIT in enhancing model performance for tasks requiring a nuanced understanding of musical structure and consistency in source separation.

Cross-Dataset Experiments

Table 5.2: Cross Dataset Experimentds' Results

MOD2 - APS SCORE	GuitarSet	NI Dataset Test
Train NI - Finetune GuitarSet	71.81%	7.99%
Train GuitarSet	72.77%	7.65%
Train GuitarSet + NI Dataset	68.62%	62.62%

The first experiment revealed a pronounced "forgetfulness" in the model when fine-tuned with a different dataset. After training on the NI dataset and subsequent fine-tuning on the GuitarSet, the model's performance on the GuitarSet test set was suboptimal, indicating a loss of information acquired during initial training. This behavior points to the model's tendency to rapidly overwrite learned features, a phenomenon that necessitates further investigation into memory retention during fine-tuning.

Similarly, the second experiment highlighted the challenges of dataset transferability. Training on real recordings (GuitarSet) did not translate well to performance on the synthetic dataset (NI Dataset), suggesting a significant gap between the model's learning on real versus synthetic data. This underlines the importance of dataset diversity in training for robust model generalization.

In the third experiment, the combination of the NI Dataset and GuitarSet for training yielded a model with performance metrics close to the highest APS scores achieved when trained exclusively on either dataset. This blend of datasets enhanced the model's robustness and its ability to generalize across varied data.

Moving forward, the architecture from the second modification, trained on the combined dataset, will be employed to feed the score-informed separation model. This approach aims to leverage the collective strengths of both datasets to cultivate a transcription model capable of dealing with a broader spectrum of audio inputs. The ultimate goal is to enhance the accuracy and reliability of music transcription, facilitating its application in complex tasks such as score-informed source separation.

Chapter 6

Demucs Architecture And Experiments

Contents

6.1 History of Demucs

Demucs (Deep Extractive Music Source Separation) emerged as a pivotal solution for the music source separation task, particularly on the MUSDBHQ dataset. Over the years, multiple iterations of Demucs have been introduced [73, 33, 32]. This section provides an exploration of the progressive development of the Demucs architecture over time.

6.1.1 Origins of Demucs

The initial version of Demucs, as delineated in the paper "Music Source Separation in the Waveform Domain" [73] sought to challenge the prevailing notion that models should predominantly operate in the spectrogram domain. This research hypothesized that operating solely in the waveform domain could still yield satisfactory results.

The first work of the Demucs was focused on **Adaptation of Conv-Tasnet** [17]. Originally tailored for monophonic speech separation at 8.000 Hz, Conv-Tasnet was reconfigured for stereophonic music source separation at 44.100 Hz. While it surpassed prior methods, achieving an SDR of 5.7, it fell short of the IRM oracle's 8.2 SDR. Despite its accuracy, Conv-Tasnet exhibited audio artifacts, especially in drums and bass sources. To address Conv-Tasnet's limitations, Demucs was introduced. Drawing inspiration from music synthesis models, Demucs utilizes a U-net architecture, merging a convolutional encoder with a transposed convolution-based decoder. Integral components included a bidirectional LSTM, exponentially increasing channels with depth, and gated linear units as activation functions.

Architectural Blueprint

Demucs' architecture comprises of a Convolutional Auto-encoder:

- **Encoder:** Constituted of six convolutional blocks. Each block employs a convolution with a kernel size of 8 and stride of 4, augmented with another convolution of kernel size 1, leading to the employment of gated linear units (GLUs). These GLUs enhance model depth and expressivity at minimal computational expense. The encoder is depicted in Fig. 6.1.2

- **Bi-directional LSTM:** This LSTM bridges the encoder and decoder, using a linear layer to reduce the channel count to match the encoder's output.

- **Decoder:** Essentially a reverse-engineered encoder. After processing through the blocks, the sources are synthesized only at the final layer, with each channel producing its corresponding waveform. The decoder is depicted in Fig. 6.1.2

Mirroring the Wave-U-Net [4] approach, Demucs uses skip connections between encoder and decoder blocks of the same index. These connections afford direct access to the original signal, enabling the direct transfer of the input signal's phase to the output. The Demucs architecture is illustrated in Fig. 6.1.1

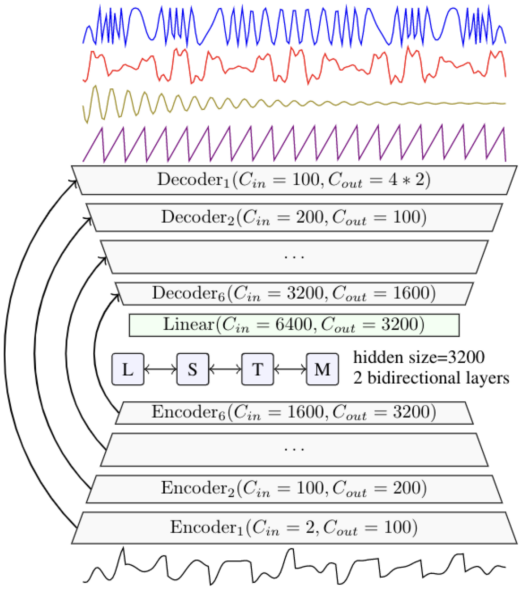


Figure 6.1.1: Demucs architecture with the mixture waveform as input and the four sources estimates as output. Arrows represents U-Net connections [73].

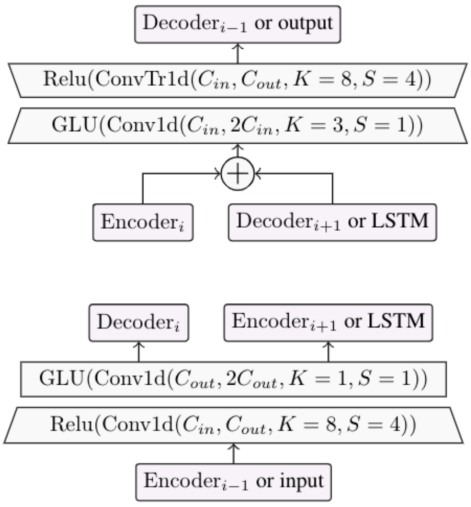


Figure 6.1.2: Detailed view of the layers Decoder on the top and Encoder on the bottom. Arrows represent connections to other parts of the model. For convolutions, Cin (resp Cout) is the number of input channels (resp output), K the kernel size and S the stride [73].

6.1.2 Hybrid Demucs Architecture

Recent developments in Demucs have advanced the hybridization of spectrogram and waveform source separation. While theoretically, spectrogram and waveform models should have no difference, in practice, with constrained datasets like MUSDB's 100 songs, inductive biases have a significant impact. Various auditory artifacts arise depending on the domain utilized.

Défossez et al. [33] expanded the Demucs architecture for end-to-end hybrid waveform/spectrogram domain source separation. Integrating the original U-Net architecture, the model introduced parallel branches: temporal (time) and spectral (frequency). Further enhancements include compressed residual branches featuring dilated convolutions, LSTM, and localized attention.

Architectural Blueprint

Built on the U-Net encoder/decoder structure, integrated with a BiLSTM for long-range context. Symmetrically designed encoder and decoder layers process 44.100 Hz audio, with 6 layers each.

Hybrid Approach:

- **Temporal Branch:** Processes input waveform similarly to standard Demucs, with GELU activations supplanting ReLU.
- **Spectral Branch:** Processes the spectrogram obtained from a Short-Time Fourier Transform (STFT) to match the temporal branch output. Convolutions are applied frequency-wise to manage frequency dimensions, while keeping the temporal dimension intact (and equal to the number of frames of the STFT).
- **Shared Encoder/Decoder Layer:** Summation of the temporal and spectral representations undergo further encoding and decoding. The result feeds into both temporal and spectral decoders. The architecture is highlighted by dual U-Net structures with respective skip connections.

- **Output:** The spectral branch undergoes an Inverse Short-Time Fourier Transform (ISTFT) and combines with the temporal branch for the final prediction.

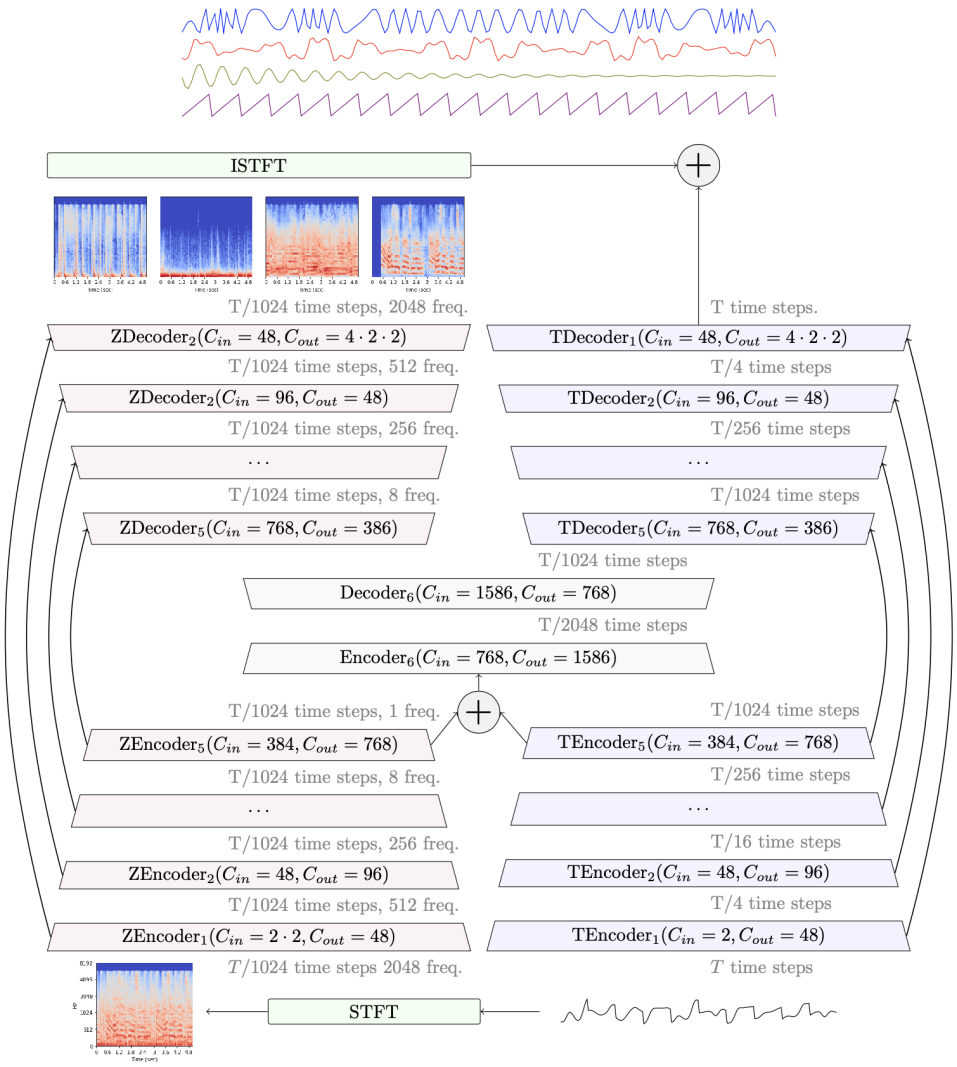


Figure 6.1.3: Hybrid Demucs architecture. The input waveform is processed both through a temporal encoder, and a spectral encoder; in the second case the input undergoes through the STFT. The two representations are summed when their dimensions align. Both decoder branches are built symmetrically to their respective encoders. The output spectrogram goes through the ISTFT and is summed with the waveform outputs, giving the final model output. The Z prefix is used for spectral layers, and T prefix for the temporal ones [33].

Technical Features

- **Alignment:** Ensured alignment between spectral and temporal representations regardless of input length, leveraging convolution padding techniques from models like MelGAN [93].
- **Spectrogram Representation:** Both complex number and amplitude spectrogram representations were investigated. Regardless of the chosen representation, the final loss is applied in the waveform domain.
- **Compressed Residual Branches:** Introduced between two convolution layers, these branches use dilated convolutions. Respective to time dimension, two such branches exist per encoder layer. Local

attention and a 2-layer BiLSTM provide long-range context for the 5th and 6th encoder layers.

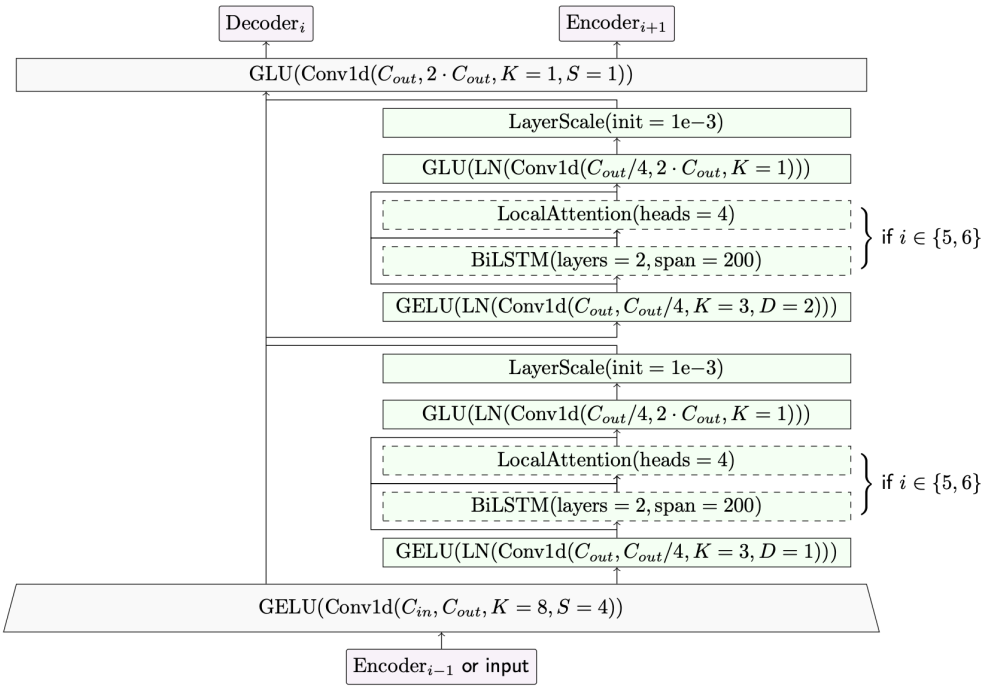


Figure 6.1.4: Representation of the compressed residual branches that are added to each encoder layer. For the 5th and 6th layer, a BiLSTM and a local attention layer are added [33].

- **Local Attention:** A modification of the conventional attention mechanism, local attention introduces a controllable penalty, that penalizes attending to positions that are far away. This dynamic approach, which contrasts fixed penalty systems in Natural Language Processing [94], proves innovative in the audio domain.

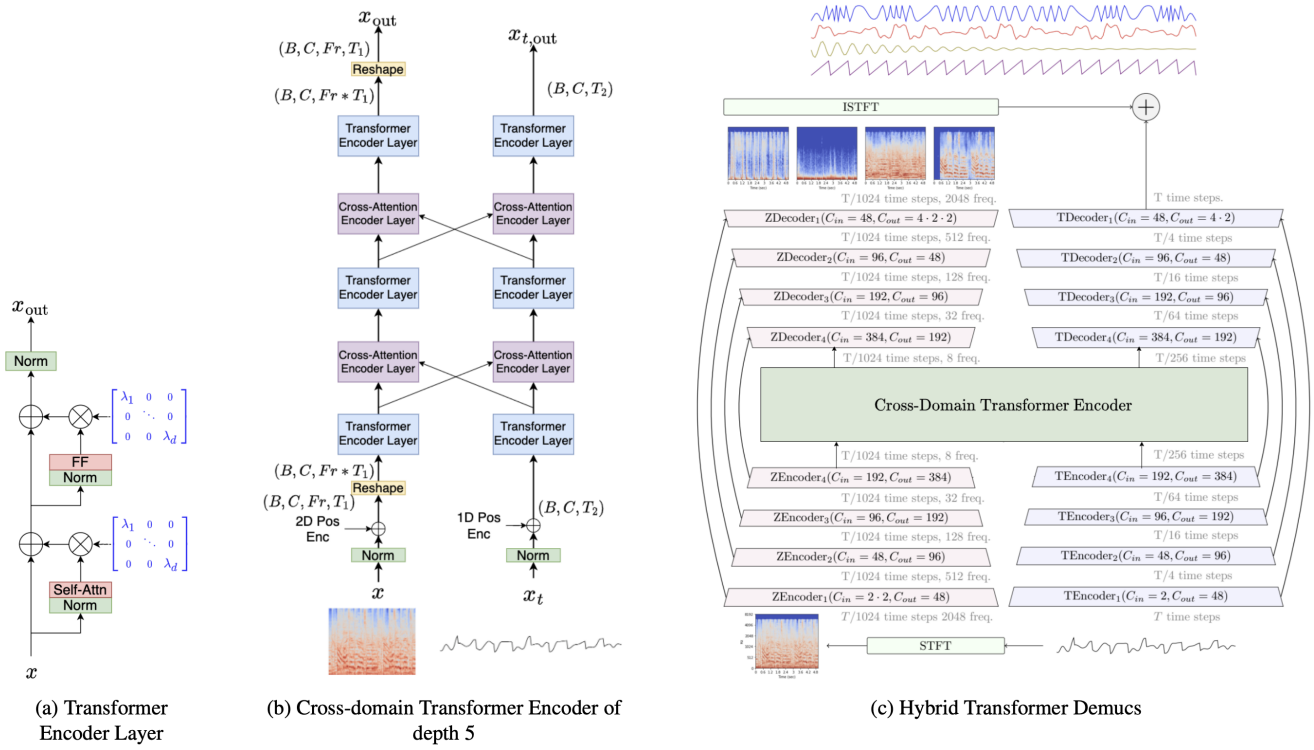
6.1.3 Hybrid Transformer Demucs for Music Source Separation

In the evolving field of Music Source Separation (MSS), a pertinent inquiry emerges regarding the significance of long-range contextual information versus localized acoustic features. Recent advancements in other domains [95, 96] has shown that attention-based Transformers are quite effective at handling long sequences of data. Défossez et al. [32] introduces the Hybrid Transformer Demucs (HT Demucs), a fusion of temporal/spectral bi-U-Net premised on the original Hybrid Demucs. The novel architecture incorporates a cross-domain Transformer Encoder, integrating self-attention in each domain as well as cross-attention inter-domain.

Architectural Blueprint

The foundational Hybrid Demucs consists of dual U-Nets, operating in both the time and spectrogram domains, each featuring five encoder and decoder layers. The layers converge post the fifth encoder, followed by a shared sixth layer. The primary decoder layer is also shared, branching into both temporal and spectral domains. The spectral output, post an inverse Short Time Fourier Transform (ISTFT), is merged with the temporal output, producing the model's prediction. The HT Demucs retains the initial four layers from its predecessor but changes the internal encoder-decoder layers. Unlike the original Hybrid Demucs, which requires careful parameter optimization to synchronize temporal and spectral data, the cross-domain Transformer Encoder exhibits adaptability with heterogeneous data.

Transformer Enhancements: As delineated in Sec. 6.1.5, the architecture showcases a single self-attention Encoder layer from the Transformer. Employing both layer (each token is independently normalized) and time-layer (all the tokens are normalized together) normalizations, it facilitates a stable training environment, synergizing with Layer Scale [97]. A consistent dimension of 384 is maintained, with auxiliary linear layers adapting to the Transformer's internal dimensionality as needed. The attention mechanism is comprised of 8 heads, while the hidden state size of the feed-forward network quadruples that of the transformer. The cross-domain encoder integrates self and cross-attention layers, both in spectral and waveform domains, augmented with 1D and 2D sinusoidal encodings.



(a) Transformer Encoder Layer

(b) Cross-domain Transformer Encoder of depth 5

(c) Hybrid Transformer Demucs

Figure 6.1.5: Details of the Hybrid Transformer Demucs architecture. (a): the Transformer Encoder layer with self-attention and Layer Scale. (b): The Cross-domain Transformer Encoder treats spectral and temporal signals with interleaved Transformer Encoder layers and cross-attention Encoder layers. (c): Hybrid Transformer Demucs keeps the outermost 4 encoder and decoder layers of Hybrid Demucs with the addition of a cross-domain Transformer Encoder between them [32].

With burgeoning sequence lengths, there's a concomitant surge in memory consumption and computational latency. To counteract this, the design incorporates sparse attention kernels introduced in the *xformer* package [98], aligned with a Locally Sensitive Hashing (LSH) blueprint to dynamically regulate the sparsity configuration. The final model variant, termed Sparse HT Demucs, attains a sparsity magnitude of 90%, achieved through multiple rounds of LSH.

6.2 Experimental Evaluation

In the subsequent section, we delineate the systematic procedures undertaken in our experimental process. This includes the methodologies employed for model adaptation and optimization, the inherent challenges faced, and the rigorous measures adopted to maintain the integrity and fidelity of our results. This segment aims to provide a comprehensive overview of the practical dimensions underpinning our research.

6.2.1 Methodology

Non-Score Informed Experiments

For our purposes, we carefully fine-tuned the model to perform the task of separating classical guitar duets. Firstly, the final layer of the model underwent a modification to output just two stereo channels, with each channel representing one of the two classical guitars. This streamlining from four to two channels allows for more targeted separation.

We made several refinements to the demucs hybrid transformer, optimizing it to operate on 4-second segments. This optimization strikes a balance between computational efficiency and the ability to capture relevant musical nuances. To enhance the model's robustness, we employed various data augmentation techniques, ensuring the model encounters a diverse range of input variations, thereby improving its generalization capability. To enhance the model's robustness, we employed various data augmentation techniques. These included traditional methods such as FlipChannels, Shift, Remix, and Scale, as well as our *OppositePanning* technique described in Sec. 6.2.1. OppositePanning, introduces varying stereo images by adjusting the panning positions of pairs of guitars.

In alignment with the integrity of the original recordings, the sample rate for the stereo waveforms was maintained at 44.100 Hz, preserving the fidelity of the authentic musical essence.

A key challenge in source separation of instrumental duets is addressing the inherent permutation ambiguity, since in contrast to the general music source separation case, the two interplaying guitars cannot be distinguished by particular timbral characteristics/profiles. To overcome this, we implemented permutation invariant training, a strategic approach that effectively mitigates this challenge by factoring in the potential source permutations during training.

During the training phase, our chosen loss function was a weighted sum of the L1 loss (0.8) and the sum loss (0.2).

$$\text{Weighted Loss} = 0.8 \times L_1 + 0.2 \times \text{Sum Loss}$$

The weights were determined based on optimal outcomes from preliminary tests. The incorporation of the sum loss ensures the model recognizes that both outputs should collectively approximate the input. To manage inherent permutation ambiguity in source separation, this loss function was further augmented with a permutation-invariant wrapper. This wrapper recalculates the most favorable loss for both possible output permutations at each iteration, guiding parameter updates accordingly. The Adam Optimizer, with a learning rate set at 0.0003, was employed, consistent with the original Demucs model. Model training spanned approximately 100 epochs, each lasting roughly an hour. It's noteworthy that the most efficacious models in terms of validation loss consistently emerged between the 70th and 100th epochs. We partitioned our data in an 80-20 train-validation split for all the following experiments with all the datasets.

For performance assessment, we employed metrics such as SDR, SI-SDR, SAR, ISR and SIR. Adhering to the median-of-medians protocol outlined in [73], these metrics were computed for both guitars of each segment. By first determining the median score for each song segment and subsequently deriving the median from these per-song scores across the entire test set, we were able to represent our evaluation with a singular, consolidated value. An illustration of the whole pipeline can be seen in Fig. 6.2.1

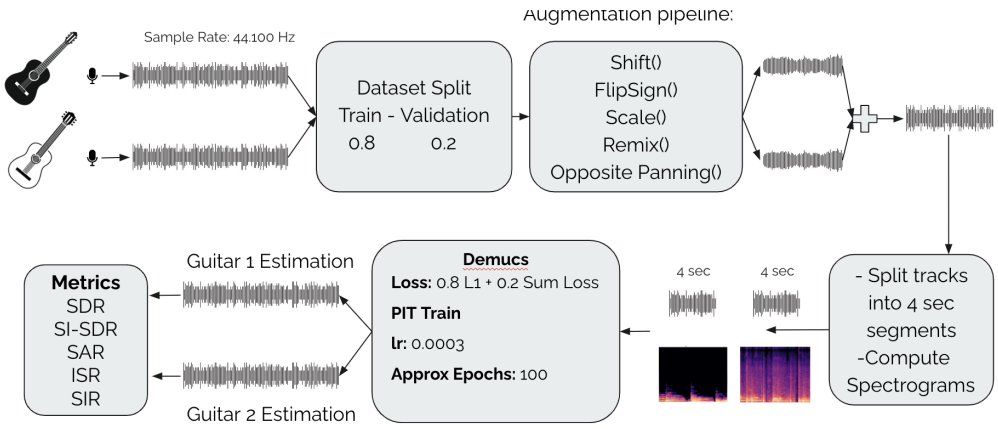


Figure 6.2.1: An overview of the pipeline utilized for training our variant of demucs in the task of guitar duet separation.

In our experimental results, we present an exhaustive exploration of various trainings conducted using a composite of datasets. These encompass GuitarDuets Dataset, Native Instruments Dataset, GuitarSet, and a segment of URMP.

Score-Informed Experiments

At this stage we propose the development of a pipeline composed of dual models. The first model would intake the combined sounds of the two guitars and strive to generate a binarized piano roll representation for each individual guitar. Afterward, the second model combines the mixed audio and the generated piano rolls to create separate audio files for each guitar.

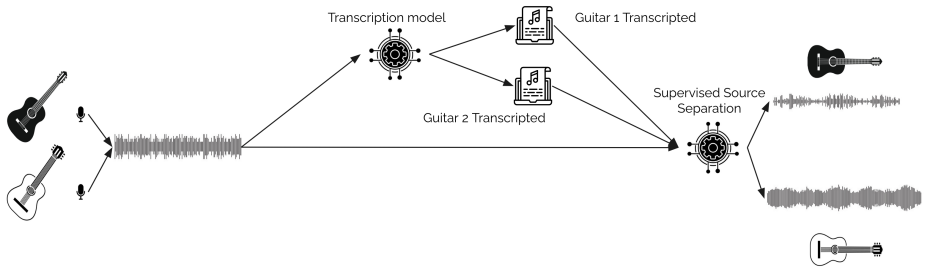


Figure 6.2.2: Proposed System Overview

This approach leverages the intrinsic correlations that are anticipated to emerge when training the primary model. By extracting piano rolls from the mixture, the model can potentially unravel the intricate dynamics and interdependencies exhibited by the two guitars in tandem. Furthermore, by incorporating the piano roll data, we enrich the source separation process with a profound layer of musical context. This representation doesn't merely differentiate the two sounds; it captures the temporal progression and pitch nuances of the guitars. This, in turn, permits the models to reference the melodic and harmonic architectures [99] when disentangling the intertwined audio sources, opening a promising avenue for enhanced accuracy and fidelity in source separation tasks.

For the purpose of the following experiments we are going to utilize the RSE (Residual Shuffle Exchange Network) as trained and analyzed in Sec 5.

Integration of Activity Labels into Demucs Architecture The Demucs architecture, as depicted in Fig. 6.1.5, processes audio segments of four seconds in duration. To facilitate the separation task, activity

labels, which are binary vectors indicating the presence or absence of each of the 128 MIDI notes during small temporal frames, are concatenated to the input at specific points in each branch of the architecture.

Time Domain Concatenation In the time domain branch, the activity labels are concatenated after the third TEncoder layer. The binary vector for each guitar has a dimensionality of 128 $\times$ samples of 4 seconds segment, yielding a combined shape of 256 $\times$ samples of 4 seconds segment for both guitars. The dimensions of the encoder outputs before concatenation are as follows:

- After TEncoder layer 0: Size($[4, 48, 44100]$)
- After TEncoder layer 1: Size($[4, 96, 11025]$)
- Pre-concatenation TEncoder layer 2: Size($[4, 192, 2757]$)
- Post-concatenation: Size($[4, 448, 2757]$)

Post concatenation, the dimensions are updated to torch.Size($[4, 448, 2757]$), accounting for the additional label information. The activity labels are resampled from 44100 samples to match the 2757 sample time resolution of the encoder at this stage.

Frequency Domain Concatenation The activity labels are integrated into the frequency domain branch after the second ZEncoder layer. The concatenation process is tailored to preserve the distinct identity of the 128 MIDI notes, which is critical for the model's ability to distinguish between different notes during the sound separation task. The shape of the activity labels for concatenation is $2 \times 128 \times$ samples of 4 seconds segment, aligning with the two guitars' MIDI notes. The dimensions of the encoder outputs before and after concatenation are itemized below:

- After ZEncoder layer 0: Size($[4, 48, 128, 173]$)
- Pre-concatenation ZEncoder layer 1: Size($[4, 96, 128, 173]$)
- Post-concatenation: Size($[4, 98, 128, 173]$)

This adjustment increases the channel dimension from 96 to 98, allowing the model to process the note information alongside the audio data. To ensure compatibility with the encoder's frequency resolution, the activity labels are resampled from their original sample rate of 44.100 Hz to match the encoder's frequency resolution of 173.

The concatenation on this stage allows the preservation of the distinct 128 notes without the need for downsampling, crucial for maintaining separation fidelity.

OppositePanning Augmentation

The implementation of the *OppositePanning* augmentation technique is motivated by the prevalent use of panning in modern audio recordings. In productions where multiple instruments are present, panning is a common technique to create spatial depth and a fuller sound image. This is particularly relevant when separating instruments like classical guitars, which are often spatially positioned in a mix.

Since our datasets consists of raw recordings, no such panning procedure has been applied, presenting a scenario that is not entirely representative of real-world conditions. To address this, *OppositePanning* is introduced as an augmentation strategy. This technique artificially creates scenarios where two guitars are panned differently, closely mimicking real recording conditions. An illustration of the augmentation can been seen in Fig. 6.2.3

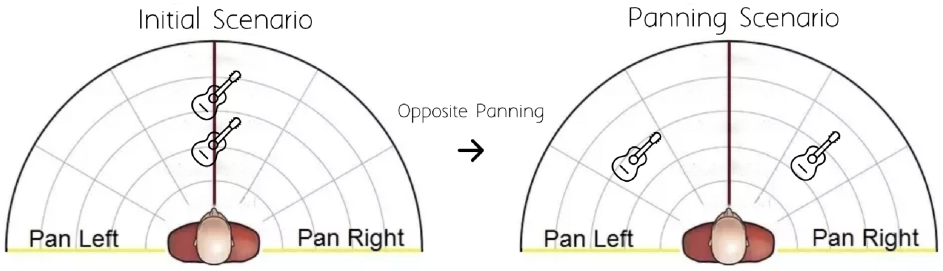


Figure 6.2.3: Illustration of Opposite Panning Augmentation.

By training the model on data augmented with such spatial variations, it becomes adept at handling real-world scenarios where panning plays a significant role in the audio mix. This augmentation thus aims to bridge the gap between the dataset's limitations and the practical realities of audio production, ensuring that the model is well-equipped to perform in typical, panned environments.

6.2.2 Experiments

Traditional Approach NMF

In the pursuit of advancing music transcription and source separation technologies,we recognized the necessity of establishing a reliable baseline, particularly given the scarcity of prior research in the specific context of monotimbral source separation with polyphonic instruments. To address this gap and set a foundational benchmark, we turned to a classic Digital Signal Processing (DSP) approach, specifically the Non-negative Matrix Factorization (NMF) algorithm, for our initial baseline tests.

The choice of NMF as a baseline test is informed by its simplicity, interpretability, and established track record in handling audio separation tasks [100]. By decomposing a mixed audio signal into a set of basis components and their corresponding weights, NMF provides a clear framework to understand how different sounds contribute to the overall mixture. This characteristic makes it an ideal starting point for our experiments, offering a straightforward yet effective method for initial assessments.

In our baseline tests with NMF, we aim to separate and transcribe the audio of two guitars from mixed signals. The performance of NMF in this context sets a foundational benchmark against, which we can measure the advancements brought about by our modified dual-output architecture. This comparison is vital to demonstrate the incremental improvements and justify the need for more sophisticated models in scenarios where the interplay of sounds is more complex than what NMF can effectively handle.

For the following table we are going to use the acronyms for the evaluation metrics so we repeat the definitions as outlined in Section 3.7:

SDR Source to Distortion Ratio

SI-SDR Scale-Invariant Source to Distortion Ratio

SAR Source to Artifacts Ratio

SIR Source to Interference Ratio

ISR Source Image to Spatial Distortion Ratio

Table 6.1: NMF Algorithm Results on myTestSet

Metric	Guitar 1	Guitar 2
SDR	-2.663	-1.890
SI-SDR	-41.364	-41.364
SAR	-34.257	-34.596
SIR	-0.011	0.020
ISR	6.801	-1.912

Non-negative Matrix Factorization (NMF) yields very low values, primarily due to its inherent limitations in capturing the intricate characteristics of audio signals. NMF operates by decomposing a spectrogram into a set of basis components and their corresponding activations, attempting to reconstruct the original signal through this simplified representation. However, when dealing with sources that share closely matched timbres, such as classical guitars, the basis components generated by NMF may be too generic or "pure" to effectively distinguish between the subtle differences in sound characteristics that define each source. As a result, NMF might excel in tasks where the sources have significantly different timbral qualities but fall short in scenarios requiring the discrimination of finer auditory details.

6.2.3 Demucs Non Score Informed Experiments

In our pursuit to optimize the performance of monotimbral music source separation, we conducted an experiment to determine the most effective version of the Demucs architecture for our specific application, all the different versions where trained and tested on the GuitarDuets Dataset.

Table 6.2: Revised Results for Different DEMUCS Architectures

Metric	Initial (Our Implementation)	Hybrid	Hybrid Transformer
SDR	G1: 3.94 G2: 0.15	G1: 4.06 G2: 0.80	G1: 5.14 G2: 0.99
SI-SDR	G1: 0.27 G2: 0.30	G1: 0.63 G2: 0.71	G1: 1.80 G2: 1.81
SAR	G1: 6.57 G2: 0.78	G1: 6.67 G2: 0.83	G1: 7.72 G2: 1.42
SIR	G1: 9.22 G2: 3.84	G1: 9.19 G2: 3.94	G1: 10.19 G2: 4.94
ISR	G1: 5.82 G2: 0.04	G1: 5.61 G2: 0.03	G1: 6.40 G2: 0.57

Based on the above table, it is evident that the hybrid transformer version of Demucs yielded the best results. This conclusion is drawn not only from the better performance metrics associated with this version but also from the qualitative evaluation of the audio outputs. When listening to the separated sources, it was observed that the hybrid transformer Demucs introduced the least amount of noise, making it the preferred choice for our project. This finding underscores the importance of selecting an appropriate model architecture that balances computational efficiency with high-quality audio separation.

For the following experiments we are going to use the **DEMUCS Hybrid Transformer** version.

Training on GuitarDuets

Table 6.3: Results for Model Trained on GuitarDuets

Metric	Test GuitarDuets	Test NI Dataset	Test Guitarset
SDR	G1: 5.140 G2: 0.991	G1: 2.814 G2: 0.933	G1: 1.771 G2: 2.324
SI-SDR	G1: 1.803 G2: 1.806	G1: -7.877 G2: -9.312	G1: -2.615 G2: -2.624
SAR	G1: 7.719 G2: 1.417	G1: 4.026 G2: 3.157	G1: 5.696 G2: 5.605
SIR	G1: 10.186 G2: 4.935	G1: 6.852 G2: 7.104	G1: 7.121 G2: 7.152
ISR	G1: 6.400 G2: 0.571	G1: 8.814 G2: 3.776	G1: 2.132 G2: 3.097

The metrics for the three datasets, as shown in the above table 6.3, although they seem relatively high, are not satisfactory. They highlight the model's challenges in accurately distinguishing the timbres of different guitars. Particularly within GuitarDuets, the slightly enhanced SDR for the first guitar implies that the model possesses a certain degree of recognition capability, more so when the guitar primarily functions as a solo instrument. This suggests that isolating a single melody becomes more feasible when it is accompanied by another guitar playing patterns.

From the above table we can infer that the model's performance is not uniform across the various datasets. The discrepancy in performance can be attributed to the inherent differences in the datasets' composition. Specifically, the NI Dataset, being synthetic, poses generalization challenges for the model, which is trained on real data from GuitarDuets, indicating a potential shortfall in the model's ability to adapt from real to synthetic data scenarios. Conversely, the GuitarSet's lack of timbral distinction between its parts—since both are played by the same guitar—complicates the model's decision-making process based on timbre alone. Informal subjective listening of the outputs generated from GuitarDuets corroborate that while the model is capable of enhancing the dominant parts for each guitar and diminishing the others, it does not achieve a flawless separation. This limitation becomes markedly evident in the GuitarSet test set, where the model demonstrates a relative proficiency in isolating the accompaniment guitar. The accompaniment guitar's consistent role likely aids the model in this aspect, yet it significantly struggles to distinguish the solo guitar. The observed performance variance underscores the critical need for model enhancements to better generalize across different data forms and to more effectively discern and isolate timbral characteristics in complex musical compositions.

Training on GuitarSet

Table 6.4: Results for Model Trained on Guitarset

Metric	Test on GuitarDuets	Test on NI Dataset	Test on Guitarset
SDR	G1: 4.580 G2: 1.323	G1: 2.454 G2: -0.125	G1: 7.880 G2: 7.551
SI-SDR	G1: -4.223 G2: -4.178	G1: -2.574 G2: -2.716	G1: -8.135 G2: -8.074
SAR	G1: 8.314 G2: 6.627	G1: 2.640 G2: 2.776	G1: 9.416 G2: 10.521
SIR	G1: 7.552 G2: 8.939	G1: 5.110 G2: 7.361	G1: 13.420 G2: 13.669
ISR	G1: 8.115 G2: -0.310	G1: 6.978 G2: 2.086	G1: 10.845 G2: 10.007

In this experiment, a particular model distinguishes itself as the most effective, exhibiting commendable performance on the GuitarSet dataset, yet showing limitations when applied to GuitarDuets and the synthetic NI Dataset without modifications from earlier experiments. The inherent structure of the GuitarSet, with one guitar serving as accompaniment and another performing a relatively straightforward solo, primarily influences this result. The solo parts, often consisting of single, higher-pitched notes, contrast with the accompaniment's rhythmic patterns, which vary harmonically. This distinct separation between solo and accompaniment within the GuitarSet simplifies the model's task, making it considerably easier than identifying guitars based on their unique timbral qualities. Furthermore, it's important to note, as previously mentioned, that the GuitarSet lacks timbral differences between the two guitars played simultaneously, limiting the model's ability to generalize to datasets where timbral characteristics, rather than musical roles, are essential for distinguishing between guitars. Informal subjective listening confirm the model's proficiency in the GuitarSet, where it achieves near-perfect separation between the accompaniment and solo parts. This capability also extends to test dataset of GuitarDuets, where the model effectively discerns melody from accompaniment, a feature particularly beneficial for classical compositions characterized by a clear division between these roles. However, the model faces difficulties with more complex classical guitar duets, where the intertwining of parts blurs the lines between accompaniment and solo roles. This highlights a significant challenge: developing models capable of nuanced differentiation in scenarios where musical roles are not distinctly defined, thus requiring a more sophisticated approach to recognize and isolate intertwined musical elements.

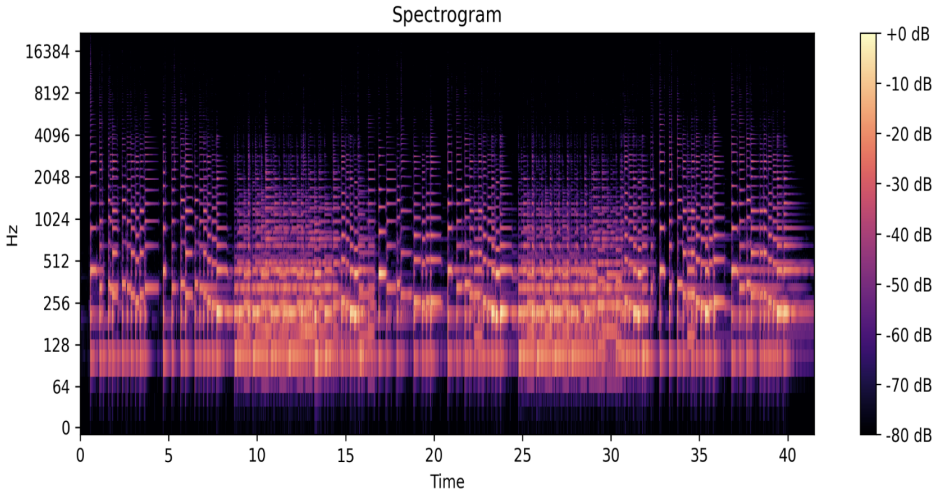


Figure 6.2.4: Clear Prediction from GuitarSet trained model.

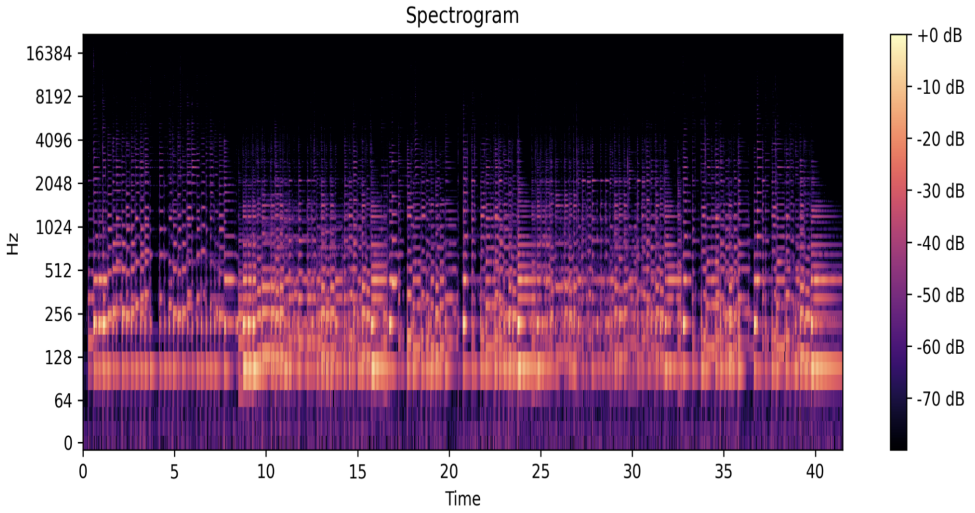


Figure 6.2.5: Noisy prediction from GuitarDuets trained model.

In the above Figures 6.2.4 and 6.2.5, we observe two distinct outcomes from models trained on different datasets. For the computation of the spectrograms presented above, and subsequent figures, standard parameters were employed as defined by the *plt.specgram* function in matplotlib. Specifically, a Hanning window was used with a length of 256 samples. The moving stride, was set to 128 samples, the Discrete Fourier Transform (DFT) length, matched the window length of 256 samples.

The first Figure 6.2.4 showcases a model trained on the GuitarSet dataset, which yields a clear and accurate prediction of a melody. This result reflects the inherent quality of the GuitarSet data, characterized by distinct melodies from a single guitar and the absence of microphone bleeding, allowing the model to learn and separate melodies effectively without introducing artifacts.

Conversely, the second Figure 6.2.5 presents predictions from a model trained on the GuitarDuets dataset. Here, the output is noticeably noisier compared to that of the GuitarSet model. This disparity can be attributed to the distinct structure and nature of the GuitarDuets dataset, which unlike GuitarSet, contains instances of microphone bleeding. This aspect of the dataset introduces additional complexity during training, challenging the model's ability to cleanly separate the source signals. This comparison underscores the significant impact dataset characteristics have on the performance of source separation models, highlighting the importance of dataset selection and preparation in training effective models for music source separation.

Training on GuitarDuets and GuitarSet

Table 6.5: Results for Model Trained on GuitarDuets + Guitarset

Metric	Test on GuitarDuets	Test on NI Dataset	Test on GuitarSet
SDR	G1: 4.200 G2: 1.206	G1: 2.865 G2: 1.485	G1: 7.992 G2: 8.303
SI-SDR	G1: -0.334 G2: -0.340	G1: -2.345 G2: -3.487	G1: -11.403 G2: -11.367
SAR	G1: 8.839 G2: 10.670	G1: 3.698 G2: 4.188	G1: 8.902 G2: 10.192
SIR	G1: 6.235 G2: 7.732	G1: 5.398 G2: 7.403	G1: 14.505 G2: 16.088
ISR	G1: 5.908 G2: -1.427	G1: 5.079 G2: 1.142	G1: 12.862 G2: 12.424

In this experiment, merging the two datasets—GuitarDuets and the GuitarSet—appeared to have minimal impact on the model's performance on the GuitarSet test set. This could be attributed to the GuitarSet's larger size, offering the model more learning opportunities. Furthermore, as previously discussed, separating guitars in the GuitarSet is a comparatively simpler task. The metrics for GuitarDuets remained consistent with prior experiments. Informal listening corroborated these findings, with results for the GuitarSet mirroring the previous experiment and GuitarDuets showing consistent outcomes across all experiments.

Training on Native Instruments Dataset

Table 6.6: Test Results for NI Dataset training

Metric	Test GuitarDuets	Test NI Dataset	Test Guitarset
SDR	G1: 2.672 G2: 0.012	G1: 2.247 G2: 3.272	G1: 1.325 G2: 2.307
SI-SDR	G1: -0.841 G2: -0.815	G1: -9.130 G2: -7.672	G1: -3.004 G2: -2.945
SAR	G1: 4.706 G2: 8.491	G1: 4.283 G2: 3.511	G1: 5.794 G2: 4.569
SIR	G1: 3.989 G2: 8.612	G1: 10.609 G2: 5.811	G1: 6.874 G2: 5.813
ISR	G1: 6.368 G2: -2.957	G1: 7.816 G2: 10.043	G1: 1.030 G2: 3.306

In our experiment, we assessed the performance of a model trained on synthetic data from the NI Dataset, subsequently applying this model to both real and synthetic test datasets, including GuitarDuets and the Guitarset. The results, as outlined in the provided table, indicate a underperforming outcome, compared to models trained on real data. This underperformance is particularly evident in the SDR and SI-SDR metrics across all test datasets, suggesting a limitation in the model's ability to generalize from synthetic to real data contexts effectively.

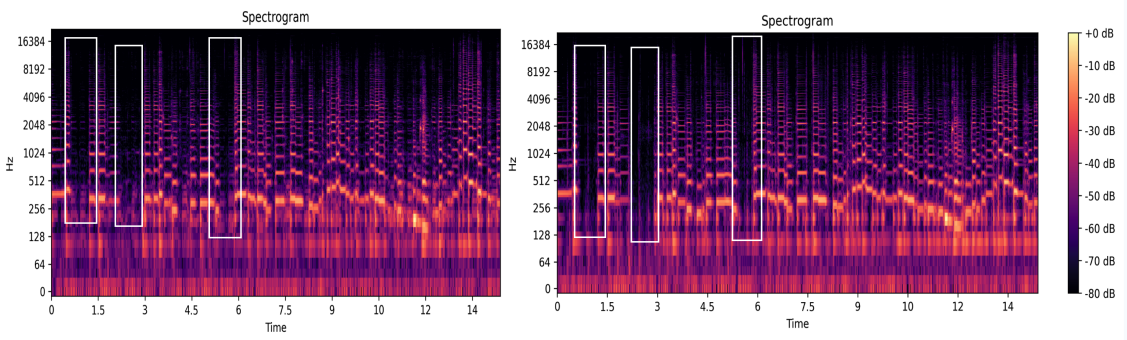


Figure 6.2.6: Illustration of noise on predicted Data.

In the above figure 6.2.6 we have a comparison between two models' performance in music source separation on the same real recording track: one trained on synthetic data and the other on real-world recordings. The right side of the figure shows the output of the model trained on synthetic data, where we observe the introduction of noisy artifacts. In contrast, the left side, depicting the output from the model trained on real data, demonstrates a cleaner separation with fewer artifacts. This visual evidence highlights the challenge of domain mismatch, where training on synthetic data can lead to the incorporation of artifacts when the model is applied to real-world data. The comparison underscores the importance of training models on data that closely matches the target domain to minimize the introduction of unwanted noise and improve the quality of source separation.

Training on GuitarDuets + Native Instruments

Table 6.7: Test Results for GuitarDuets + NI training

Metric	Test GuitarDuets	Test NI Dataset	Test Guitarset
SDR	G1: 5.988 G2: 0.934	G1: 5.016 G2: 4.185	G1: 1.537 G2: 1.880
SI-SDR	G1: 2.370 G2: 2.362	G1: -14.114 G2: -13.439	G1: -3.272 G2: -3.285
SAR	G1: 8.835 G2: 0.893	G1: 4.641 G2: 3.027	G1: 4.732 G2: 3.412
SIR	G1: 11.777 G2: 4.271	G1: 11.280 G2: 8.667	G1: 7.460 G2: 6.447
ISR	G1: 7.229 G2: 2.215	G1: 14.055 G2: 11.992	G1: 2.525 G2: 3.852

The integration of both real and synthetic data for training presents an approach to overcoming the limitations observed when models are trained exclusively on either type of dataset. This experiment aimed to evaluate the efficacy of such a combined dataset, incorporating real data from GuitarDuets with synthetic data from the NI Dataset, in enhancing model generalization across different test sets, including GuitarDuets, the NI Dataset, and the Guitarset.

The results, as depicted in the provided table, illuminate the impact of this hybrid training approach. For the test sets of GuitarDuets and the Guitarset, the model demonstrates a performance that closely approximates, and in some aspects, surpasses the outcomes observed with training solely on GuitarDuets. Notably, it significantly outperforms the models trained exclusively on synthetic data. This improvement underscores the potential benefits of diversifying training data sources to bridge the gap between real and synthetic environments. However, the gains in performance, while meaningful, are not as substantial as one might anticipate considering the combination of datasets.

This moderate enhancement suggests that while merging real and synthetic datasets can indeed provide a richer learning context, leading to improved model adaptability, the inherent differences in the nature of these datasets pose challenges. The real data capture the complex characteristics of live musical performances, whereas the synthetic data offer a more controlled but less varied representation of musical sounds.

Training on GuitarDuets + Native Instruments + GuitarSet

Table 6.8: Test Results for GuitarDuets + NI + GuitarSet training

Metric	Test GuitarDuets	Test NI Dataset	Test Guitarset
SDR	G1: 4.306 G2: 0.765	G1: 4.233 G2: 3.612	G1: 3.264 G2: 3.365
SI-SDR	G1: -6.161 G2: -6.174	G1: -0.153 G2: -0.961	G1: -4.086 G2: -4.078
SAR	G1: 7.988 G2: 1.964	G1: 3.486 G2: 2.654	G1: 3.105 G2: 3.820
SIR	G1: 10.596 G2: 4.518	G1: 11.064 G2: 7.852	G1: 7.413 G2: 8.216
ISR	G1: 5.873 G2: 1.146	G1: 10.536 G2: 10.086	G1: 4.993 G2: 5.125

The previous experiment, which involved training a model on a comprehensive dataset consisting of GuitarDuets, the NI Dataset, and the Guitarset, demonstrated a model that achieves a relatively consistent performance across the three test datasets. This consistency is noteworthy, considering the distinct characteristics and challenges posed by each dataset. However, the performance does not reach the peak levels

observed in previous experiments where the model was trained on individual or pairs of datasets. This outcome suggests a trade-off between consistency and peak performance when expanding the training data scope.

A key observation from this experiment is the model's ability to adapt to the varying natures of the datasets it was exposed to. On the one hand, the training has created a model with broad understanding, allowing it to perform with a degree of competence across all test cases. On the other hand, the mixing of data sources appears to have introduced a level of complexity that prevents the model from achieving the high performance observed when training was more focused. Specifically, the blending of synthetic and real data, combined with the Guitarset's lack of timbral variation and fixed musical roles, seems to have resulted in a model that, while versatile, may be somewhat confounded by the inherent contradictions and nuances of the aggregated datasets.

Transfer Learning from URMPStrings to GuitarDuets

We explored the potential of transfer learning by leveraging a segment of the URMP dataset that housed similar instruments. Specifically, we extracted recordings featuring at least two instruments from the violin, viola, and cello categories. For any given recording, we synthesized all viable pairs from these instrument groups. For instance, a recording encompassing violin, oboe, cello, viola, and piano would yield the pairs: (violin, cello), (violin, viola), and (cello, viola). This methodology expanded our dataset with three distinct tracks from a single recording. With this enriched 3-hour dataset, we initially trained our model. Upon achieving satisfactory performance with the same experimental setup as the guitar duets separation, we proceeded to finetune the model using our specific guitar recordings.

Table 6.9: Experiment Results

BEST MODEL	URMP Test	MyTestSet
SDR	Ins1: 8.634 Ins2: 9.457	G1: 1.633 G2: 2.609
SI-SDR	Ins1: -17.708 Ins2: -17.649	G1: -2.440 G2: -2.444
SAR	Ins1: 9.875 Ins2: 9.637	G1: 2.906 G2: 6.088
SIR	Ins1: 16.378 Ins2: 18.725	G1: 5.113 G2: 8.250
ISR	Ins1: 20.030 Ins2: 19.029	G1: 1.621 G2: 2.817

The metrics indicate commendable performance on the SDR metric by the model on the URMP dataset. This can be attributed to the dataset's simplicity, where at most two notes are played concurrently by the two strings. This reduces the model's task to distinguishing between two notes, rather than discerning the nuanced timbres of two instruments. Informal subjective listening for the URMP dataset confirms the model's proficiency in differentiating between the two strings, despite some residual overlap. For the results from GuitarDuets using transfer learning, the model appears to recognize the roles of each guitar, amplifying each part, albeit without achieving complete separation.

Comparative Analysis of SDR Metrics in Source Separation

In the field of music source separation, the evaluation of separation quality is often quantified using metrics such as the Source to Distortion Ratio (SDR) and Scale-Invariant Source to Distortion Ratio (SI-SDR). While these metrics have been extensively used in studies focusing on the separation of different instruments, their behavior in the context of separating sources with similar timbral characteristics remains less explored. Given that most prior work involves instruments with distinct timbres, direct comparison of SDR values may not be appropriate for our study, which focuses on two classical guitars sharing very similar timbral properties. Notably, despite obtaining relatively high SDR values in comparison to general music source separation literature, our subjective listening evaluations reveal that the actual perceptual quality of the separations does not align with these quantitative assessments.

Methodology To address the challenges presented by the similar timbral characteristics of sources, our experiments entailed the creation of synthetic mixtures featuring two distinct guitar sounds with closely aligned timbres. To guarantee a fair comparison across all tests, all signals involved in these experiments were normalized. The methodology involved systematically varying the mixing ratio to simulate varying levels of source separation, with this mixing ratio specifically referring to the construction of a signal that serves as the estimate. The 'reference' in this context denotes the unadulterated signal. This precise approach allowed for the creation of a controlled environment to explore how metric responses might differ when applied to mixtures of instruments with significant timbral overlap compared to those with distinct timbres. Through a setup involving mixtures of different instruments, we aimed to investigate any potential disparities in metric responses attributable to timbral similarities.

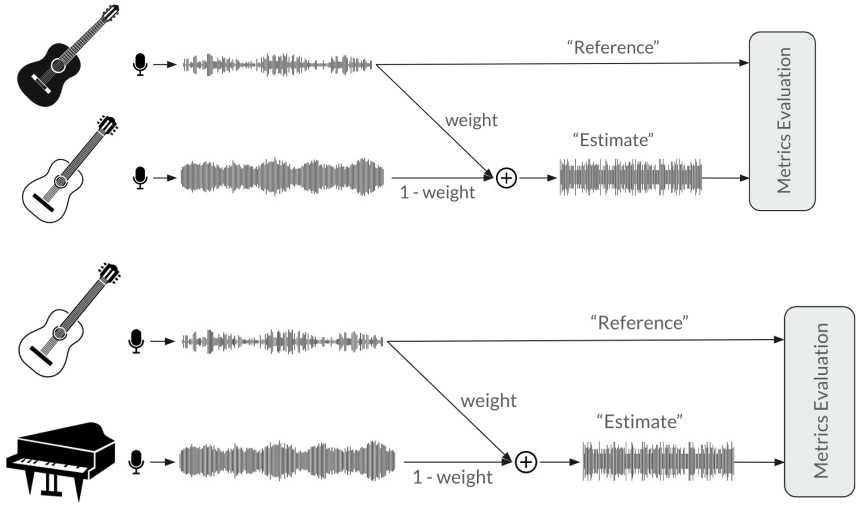


Figure 6.2.7: Methodology of metrics evaluation.

Experimental Findings The results, illustrated in Fig. 6.2.8, indicate that the SDR and SI-SDR values for the guitar mixtures are consistently higher than those obtained from mixtures of different instruments. Specifically, the same SDR values were observed in guitar mixtures even when one guitar was 0.3 times lower in volume compared to the different instruments. This suggests that the similarity in timbre between the two guitars introduces a challenge for the metrics to accurately assess the quality of separation.

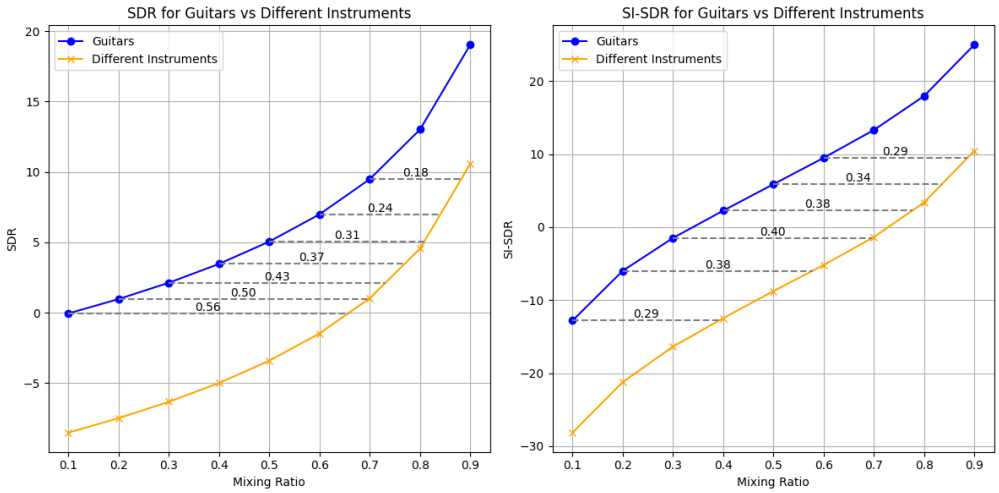


Figure 6.2.8: Comparative analysis of SDR and SI-SDR metrics for mixtures of two classical guitars versus different instruments.

Conclusions From the observed data, we conclude that SDR and SI-SDR values are influenced by the similarity of timbral characteristics between the sources. Consequently, these metrics should not be directly compared to those from other research in audio source separation involving different instruments. It should be noted that our approach of using a weighted mixture to simulate source separation is an initial approximation and not a full representation of the complex processes involved in actual separation algorithms. Despite this simplification, our results provide preliminary insights into the metrics' behavior. Specifically, Fig. 6.2.8 demonstrates that on average, an SDR value equivalent to that of a mixture of different instruments can be obtained when the weight of one guitar in the mixture of two guitars is 0.3 times lower. This suggests that higher SDR or SI-SDR values in our study should not necessarily be interpreted as superior separation performance relative to studies with more diverse instrument separations. This underlines the importance of considering the nature of the source material when evaluating and comparing source separation algorithms.

Demucs Score Informed Experiments

The initial experiment is designed to elucidate the optimal method for integrating activity labels of notes into our network, given that the Demucs architecture operates across both frequency and temporal domains. Using the exact same experimental setup as the previous experiments with the modification as outlined in the former section, our investigation will explore the efficacy of assigning labels within a singular domain, across both domains, or a combination, in enhancing the model's capacity for sound separation. For this purpose, we will employ the NI Dataset—a synthetic dataset generated from MIDI files, which contains the exact timing and duration of note playbacks. This approach aims to precisely determine the most effective strategy for label incorporation within the network's architecture to achieve superior sound isolation outcomes.

Table 6.10: Comparison of NI Dataset Results Across Different Branches

Metric	Time Branch	Freq Branch	Freq and Time Branch
SDR	G1: 3.965 G2: 4.367	G1: 4.747 G2: 3.703	G1: 4.756 G2: 4.842
SI-SDR	G1: -9.226 G2: -10.311	G1: 0.543 G2: 0.492	G1: 1.572 G2: 1.054
SAR	G1: 4.824 G2: 3.009	G1: 4.840 G2: 3.606	G1: 4.592 G2: 4.457
SIR	G1: 12.312 G2: 8.209	G1: 10.468 G2: 10.243	G1: 11.475 G2: 10.592
ISR	G1: 12.218 G2: 12.252	G1: 11.659 G2: 12.376	G1: 14.373 G2: 13.078

The experiment evaluated three distinct approaches to data concatenation within the Demucs structure: solely within the time domain, exclusively in the frequency domain, and a hybrid approach that integrates data across both domains. The analysis, as detailed in the accompanying table, reveals that the hybrid approach of concatenating data in both the frequency and time domains achieves the most promising outcomes. This performance can be attributed to the inherent design and strengths of the Demucs architecture 6.1.3, which has historically shown improved efficiency when leveraging both domains concurrently.

Demucs, from its inception to its latest iteration, has consistently demonstrated significant gains in performance by utilizing information from both the frequency and temporal domains. This architecture is adept at discerning which domain's information is more pertinent for a given task, thus effectively reconstructing the sound by drawing on the strengths of both domains. The results of this experiment underscore the rationale behind the superior performance observed when employing a hybrid concatenation approach: by engaging both domains, the model can access a more comprehensive dataset, enabling a better sound separation. This finding not only reinforces the architecture's versatility but also highlights the importance of a multifaceted approach in achieving optimal sound isolation results.

In the pursuit of refining the Demucs architecture's capacity for sound separation, the subsequent experiment focuses on the utility of activity labels, specifically the role of RSE's transcription model accuracy in enhancing

the Demucs framework. We investigate whether the Demucs architecture can benefit not only from ground-truth labels but also from high-accuracy predictions of note labels provided by a transcription model. Given that transcription models inherently introduce artifacts, this experiment evaluates the advantage of employing a Residual Sound Exchange (RSE) model trained on the NI dataset to generate near-perfect activity labels, which are then used to inform the Demucs model.

Table 6.11: Comparison of NI Dataset Results for Label Quality

Metric	No Labels	Almost Perfect Labels	Perfect Labels
SDR	G1: 2.247 G2: 3.272	G1: 3.527 G2: 2.049	G1: 4.756 G2: 4.842
SI-SDR	G1: -9.130 G2: -7.672	G1: -1.974 G2: -2.336	G1: 1.572 G2: 1.054
SAR	G1: 4.283 G2: 3.511	G1: 3.520 G2: 3.115	G1: 4.592 G2: 4.457
SIR	G1: 10.609 G2: 5.811	G1: 10.580 G2: 7.443	G1: 11.475 G2: 10.592
ISR	G1: 7.816 G2: 10.043	G1: 7.900 G2: 8.561	G1: 14.373 G2: 13.078

The analysis of label quality on the Demucs architecture, as summarized in the table above, offers insights into the model's performance with respect to different label accuracies. While the integration of almost perfect labels from the Residual Sound Exchange Network (RSE) transcription model does not markedly influence the Signal-to-Distortion Ratio (SDR), it is evident that these labels significantly enhance the Scale-Invariant Signal-to-Distortion Ratio (SI-SDR). This metric is particularly crucial as it better reflects the perceptual quality of the audio separation, suggesting that the precision of activity labels plays a pivotal role in achieving higher fidelity in sound isolation tasks.

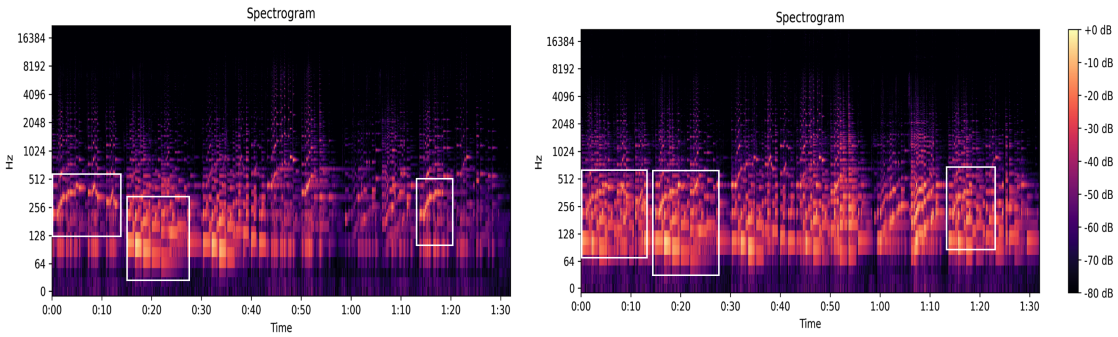


Figure 6.2.9: Prediction with almost perfect labels (left) and No Labels (right).

In the above figure 6.2.9, we observe the significant impact of predicted labels on the performance of the source separation algorithm. On the left side of the figure, the algorithm, equipped with labels, demonstrates a heightened level of certainty for specific melodies or accompanying parts. This distinction is noticeably clearer when compared to the image on the right, where the separation quality is inferior. The enhanced clarity and accuracy in identifying and isolating melodies or accompaniments in the labeled scenario result in a higher Scale-Invariant Signal-to-Distortion Ratio (SI-SDR) score. Other metrics, such as Signal-to-Artifact Ratio (SAR), Signal-to-Interference Ratio (SIR), and Image-to-Spatial Ratio (ISR), exhibit marginal variations, indicating that the label quality predominantly affects the SI-SDR metric within the scope of this experiment.

Table 6.12: Median Metrics for GuitarDuets with RSE trained on NI and Guitarset

Metric	Values
SDR	G1: 5.399 G2: 1.166
SI-SDR	G1: 1.657 G2: 1.664
SAR	G1: 7.815 G2: 2.101
SIR	G1: 11.280 G2: 5.158
ISR	G1: 6.808 G2: 0.589

These are the final results from the transcription model trained on NI Dataset combined with GuitarSet and then subsequently the score informed demucs architecture trained on GuitarDuets with soft labels creating from the aforementioned transcription architecture. In the following section we are going to have an overall overview of the models' performance on GuitarDuets test set.

6.2.4 Discussion

Overall Overview of myTestSet

For the following table we are going to use the abbreviations:

GD GuitarDuets Dataset

GS GuitarSet Dataset

NI NI Dataset

GS SI GuitarDuets Score Informed

Table 6.13: Comprehensive Test Results across Different Training Sets

Metric	URMP	GD+GS+NI	GD+GS	GS	NI	GD	GD SI	GD+NI
SDR	G1: 1.633 G2: 2.609	G1: 4.306 G2: 0.765	G1: 4.200 G2: 1.206	G1: 4.580 G2: 1.323	G1: 2.672 G2: 0.012	G1: 5.140 G2: 0.991	G1: 5.399 G2: 1.166	G1: 5.988 G2: 0.934
SI-SDR	G1: -2.440 G2: -2.444	G1: -6.161 G2: -6.174	G1: -0.334 G2: -0.340	G1: -4.223 G2: -4.178	G1: -0.841 G2: -0.815	G1: 1.803 G2: 1.806	G1: 1.657 G2: 1.664	G1: 2.370 G2: 2.362
SAR	G1: 2.906 G2: 6.088	G1: 7.988 G2: 1.964	G1: 8.839 G2: 10.670	G1: 8.314 G2: 6.627	G1: 4.706 G2: 8.491	G1: 7.719 G2: 1.417	G1: 7.815 G2: 2.101	G1: 8.835 G2: 0.893
SIR	G1: 5.113 G2: 8.250	G1: 10.596 G2: 4.518	G1: 6.235 G2: 7.732	G1: 7.552 G2: 8.939	G1: 3.989 G2: 8.612	G1: 10.186 G2: 4.935	G1: 11.280 G2: 5.158	G1: 11.777 G2: 4.271
ISR	G1: 1.621 G2: 2.817	G1: 5.873 G2: 1.146	G1: 5.908 G2: -1.427	G1: 8.115 G2: -0.310	G1: 6.368 G2: -2.957	G1: 6.400 G2: 0.571	G1: 6.808 G2: 0.589	G1: 7.229 G2: 2.215

Upon analyzing the results from the various datasets, several key insights emerge regarding the performance of models in the music source separation task for two classical guitars.

Firstly we aim to analyze the best Performance by Metric SDR (Signal to Distortion Ratio): The combination of 'GuitarDuets+NI Dataset' for the first guitar (G1) yielded the highest SDR at 5.988. This suggests that the model trained on this combined dataset was most effective in overall signal separation quality for the soloing guitar. The inclusion of the NI Dataset, which is generated from a classical guitar plugin, likely provided additional information that enhanced the model's ability to distinguish between the two guitars.

SI-SDR (Scale-Invariant Signal to Distortion Ratio): Some of the datasets demonstrated negative SI-SDR values, indicating challenges in achieving scale-invariant separation, while others had low positive values. However, again the combination 'GuitarDuets + NI Dataset' dataset performed the least poorly, with values of 2.370 for G1 and 2.362 for G2. For this combination, the proximity of SDR values to SI-SDR suggests that the model adeptly preserves the scaling of the original sources, ensuring consistent separation quality.

An observation is the pronounced disparity between SDR and SI-SDR metrics, with the latter often registering lower values. This is an outcome that was expected given the analysis we carried out in Sec. 6.2.3. This divergence underscores the presence of separation errors transcending mere amplitude scaling, potentially encompassing distortions, interferences, or other artifacts. While a satisfactory SDR might suggest an overall competent separation, the SI-SDR accentuates more granular inaccuracies when amplitude variations are neutralized. Such a pronounced metric gap serves as a diagnostic indicator: consistent discrepancies across datasets could signal the model's challenges with specific separation nuances or heightened sensitivity to certain interferences. Conversely, a minimal difference between SDR and SI-SDR may be indicative of the model's robustness in source separation.

SAR (Signal to Artifacts Ratio): The combination of 'GuitarDuets + GuitarSet' for G2 achieved the highest SAR at 10.670. This indicates that this model introduced the fewest auditory artifacts during the separation process for the second guitar.

SIR (Signal to Interference Ratio): 'GuitarDuets Score Informed' dataset excelled in this metric, with values of 11.280 for G1 and 5.158 for G2. This suggests that the model trained on this dataset was most adept at clarifying separated sources, minimizing interference from other sources.

ISR (Intereference to Spillover ratio): The combination of 'GuitarDuets+NI Dataset' yielded the highest ISR values, with 7.229 for G1 and 2.215 for G2. This indicates that this model was most effective in minimizing the presence of auditory artifacts specific to the image (or spectral representation) of the source.

Overall Best Model:

The combination of "GuitarDuets + NI Dataset" appears to be the most promising achieving consistent results throughout all the metrics. This suggests that supplementing the model with a comprehensive dataset collection can enhance the model's ability to distinguish between classical guitar timbres.

The nature of each dataset that the model is trained on, plays a pivotal role in the model's performance. For instance, the clear distinction between accompaniment and solo roles in the 'GuitarSet' simplifies the separation task. In contrast, 'GuitarDuets', being real recordings with microphone bleed, presents a more complex scenario, leading to relatively lower metrics. The 'NI Dataset', generated from a virtual instrument, offers a cleaner, more controlled environment, which when combined with real recordings, can enhance the model's generalization capability. The 'URMP-Transferred' dataset, while not directly analogous to the guitar separation task, still offers valuable insights into timbral differences, aiding in the separation process.

Chapter 7

Conclusion

Discussion In this thesis, we delve into the task of monotimbral music source separation with a focus on guitar duets. This area distinguishes itself from general music source separation due to its unique challenge of separating sources that share identical timbral characteristics, particularly when considering the classical guitar's polyphonic nature. Before delving into the discussion, we summarize the key contributions of this work as follows:

- **Dataset Creation:** We introduced two monotimbral datasets specifically designed for this study. The first dataset consists of real classical guitar duet recordings, while the second is a synthetically generated dataset of duets. These resources are crucial for advancing research in monotimbral source separation.
  - **Merging Datasets:** It was demonstrated that merging the two classical guitar datasets could lead to improved results, showcasing the potential for dataset augmentation in enhancing model performance.
- **Architectural Modifications:** Existing architectures for separation and transcription were adapted to address the challenges of monotimbral source separation.
- **Pipeline Architecture:** A pipeline architecture that combines the tasks of separation and transcription was developed, facilitating a more integrated approach to addressing the problem of source separation.
- **Cross-dataset Evaluation:** Extensive evaluation was conducted across both real and synthetic datasets under various conditions.
- **Comparative Analysis:** We performed a comparative analysis of music source separation metrics, highlighting the need for metrics that are more sensitive to timbral differences in the context of monotimbral source separation.
- **PIT's Effectiveness:** The implementation of Permutation Invariant Training (PIT) has been shown to further enhance the performance of separation algorithms, affirming its value in improving source separation outcomes.

More specifically in this work, we employed the Demucs architecture, recognized for its state-of-the-art performance in music source separation and we tried to evaluate its performance on the task of monotimbral music source separation. The aim of this experiment is to assess the applicability of a high-performing architecture to monotimbral source separation, specifically, to determine its efficacy in discerning subtle differences in sound timbres and identify its constraints. Additionally, we created two distinct datasets GuitarDuets and NIDataset comprising real and synthetic data to assess the Demucs architecture and examine the model's generalization across these domains. Our findings indicate that monotimbral source separation poses significant challenges in generalization between real and synthetic domains. However, when data from both domains are combined, the separation performance improves significantly for tests on real-domain data. The challenges observed in generalizing between real and synthetic domains in monotimbral music source separation underline the importance of diverse training datasets. This insight is equally applicable to speech

and singing voice separation, where models might benefit from exposure to both carefully annotated synthetic data and the unpredictability of real-world recordings. Overall, while the Demucs architecture demonstrates notable success in separating sources with distinct timbres, its performance on monotimbral data falls short. This underscores the importance of meticulous model tuning and careful consideration of model size and architecture for effective monotimbral source separation. Furthermore, we introduced a pipeline architecture that leverages a music transcription framework to identify the notes played by each guitar. A key rationale behind opting for a transcription-focused architecture as a supportive mechanism to the separation model, over utilizing a traditional separation only model lies in the inherent advantages of transcription systems in handling distinct notes. Transcription architectures have already demonstrated commendable performance in delineating individual notes on various instruments. By accurately transcribing all notes present in the audio mix, the transcription model can learn and recognize orchestration patterns specific to guitar duets. By understanding which notes are likely to belong to each guitar based on the arrangement of notes at any given time, it can make informed decisions about note separation. Recognizing that a single musical piece can be orchestrated in myriad ways, it is crucial not to make assumptions about consistent orchestration patterns. Instead, we maintain the separation model as the final output stage in our architecture, trusting it to consider the proposed orchestration from the transcription model. This model will then refine and deliver the ultimate separated signals, relying predominantly on the timbral characteristics, which are the most crucial feature for differentiating the two classical guitars. This approach ensures that the separation is informed yet not constrained by the transcription, allowing for a dynamic response to the complexity of the audio mix. Another reason for using the transcription model is to learn and understand the natural correlations and mutual exclusivities among guitar notes. The notes predicted by the transcription algorithm then serve as soft labels to guide the separation process. This approach resulted in notable improvements in the Scale-Invariant Signal-to-Distortion Ratio (SI-SDR) metric. The incorporation of a Permutation Invariant Training (PIT) approach, combining both audio separation and transcription separation, could greatly enhance the model's performance. Finally, we conducted a comparative analysis to evaluate the effectiveness of common metrics used in general source separation for assessing monotimbral source separation. Our analysis revealed that these metrics might not fully capture the nuances of monotimbral music source separation, suggesting the need for more tailored assessment criteria in this specific context. While a satisfactory SDR might suggest an overall competent separation, the SI-SDR accentuates more granular inaccuracies when amplitude variations are neutralized. Such a pronounced metric gap serves as a diagnostic indicator: consistent discrepancies across datasets could signal the model's challenges with specific separation nuances or heightened sensitivity to certain i nterferences. Conversely, a minimal difference between SDR and SI-SDR may be indicative of the model's robustness in source separation. Expanding on the potential applications of our findings, the methodologies and models developed in this thesis could significantly contribute to the fields of speech separation and voice singing separation. Speech and singing voice, while inherently different from musical instruments in their timbral qualities, share the common challenge of separating homogeneous sources in complex auditory scenes. The success of the Demucs architecture and our pipeline in guitar duet separation hints at promising outcomes for these related tasks.

Future Work

- An extensive analysis is imperative to further understand the efficacy of common metrics in the assessment of monotimbral music source separation. This deep dive should aim to uncover potential limitations these metrics present when applied to monotimbral contexts, thereby paving the way for the development of more tailored and representative evaluation methods specific to monotimbral separation tasks.
- Conduct a formal listening test to further assess the efficacy of our separation algorithm and gain deeper insights into the representativeness of the evaluation metrics.
- Future endeavors should concentrate on expanding the real data dataset, coupled with strategies for annotating this data to enable the effective training of transcription architectures with authentic notes.

Appendix A

Bibliography

[1] *Source Separation Tutorial Landing Page Image*. [Online]. Available: https://source-separation.github.io/tutorial/landing.html.

[2] HRISTOKARAGIOZOV. *Lecture 3 Part 2 – Digital Represenatation of the Sound*. [Online]. Available: https://karagyozovblog.wordpress.com/2011/09/13/.

[3] Rafael, B. and Oertl, S. "MTSSM -A Framework for Multi-Track Segmentation of Symbolic Music". In: *World Academy of Science, Engineering and Technology* 61 (2010).

[4] Stoller, D., Ewert, S., and Dixon, S. "Wave-u-net: A multi-scale neural network for end-to-end audio source separation". In: *arXiv preprint arXiv:1806.03185* (2018).

[5] Szeliga, D. et al. "Musical instrument recognition with a convolutional neural network and staged training". In: *Procedia Computer Science* 207 (2022), pp. 2493–2502.

[6] Cheng, Y.-H., Chang, P.-C., and Kuo, C.-N. "Convolutional Neural Networks Approach for Music Genre Classification". In: *International Symposium on Computer, Consumer and Control (IS3C)*. 2020.

[7] Liu, G. et al. "Passenger flow estimation based on convolutional neural network in public transportation system". In: *Knowledge-Based Systems* 123 (2017), pp. 102–115.

[8] Husein, M. and Chung, I.-Y. "Day-Ahead Solar Irradiance Forecasting for Microgrids Using a Long Short-Term Memory Recurrent Neural Network: A Deep Learning Approach". In: *Energies* 12 (May 2019), p. 1856.

[9] Hochreiter, S. and Schmidhuber, J. "Long Short-term Memory". In: *Neural computation* 9 (1997), pp. 1735–80.

[10] Cho, K. et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014.

[11] Aust, S. and Richter, H. "Real-time processor interconnection network for fpga-based multiprocessor system-on-chip (mpsoc)". In: *Proceedings of the 4th International Conference on Advanced Engineering Computing and Applications in Sciences*. 2010.

[12] Draguns, A. et al. "Residual Shuffle-Exchange Networks for Fast Processing of Long Sequences". In: *CoRR* abs/2004.04662 (2020).

[13] Ethan Manilow Prem Seetharaman, J. S. *TF Representations and Masking*. [Online]. Available: https://source-separation.github.io/tutorial/basics/tf_and_masking.html.

[14] Huang, P.-S. et al. "Singing-Voice Separation from Monaural Recordings using Deep Recurrent Neural Networks". In: *International Society for Music Information Retrieval ISMIR*. 2014.

[15] Jansson, A. et al. "Singing Voice Separation with Deep U-Net Convolutional Networks". In: *International Society for Music Information Retrieval Conference ISMIR*. 2017.

[16] Ronneberger, O., Fischer, P., and Brox, T. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention MICCAI*. 2015.

[17] Luo, Y. and Mesgarani, N. "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation". In: *IEEE/ACM transactions on audio, speech, and language processing* 27 (2019), pp. 1256–1266.

[18] Schwabe, M. and Heizmann, M. "Improved Separation of Polyphonic Chamber Music Signals by Integrating Instrument Activity Labels". In: *IEEE Access* 11 (2023), pp. 42999–43007.

[19] Uhlich, S. et al. "Improving music source separation based on deep neural networks through data augmentation and network blending". In: *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. 2017.
[20] Yu, D. et al. "Permutation invariant training of deep models for speaker-independent multi-talker speech separation". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017.
[21] Rafii, Z. et al. *MUSDB18 - a corpus for music separation*. Zenodo, 2017. DOI: 10.5281/zenodo.1117372.
[22] Bittner, R. et al. *MedleyDB: A Multitrack Dataset for Annotation-Intensive MIR Research*. 2014.
[23] Manilow, E. et al. "Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity". In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2019.
[24] Li, B. et al. "Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications". In: *IEEE Transactions on Multimedia* 21 (2018), pp. 522–535.
[25] Hsu, C.-L. and Jang, J.-S. R. "On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset". In: *IEEE transactions on audio, speech, and language processing* 18 (2009), pp. 310–319.
[26] Xi, Q. et al. "GuitarSet: A Dataset for Guitar Transcription". In: *International Society for Music Information Retrieval ISMIR*. 2018.
[27] Sarkar, S., Benetos, E., and Sandler, M. "EnsembleSet: A new high-quality synthesised dataset for chamber ensemble separation". In: *International Society for Music Information Retrieval ISMIR*. 2022.
[28] Couturier, J.-M. "A scanned synthesis virtual instrument". In: *Conference on New Instruments for Musical Expression (NIME-02), Dublin, Ireland*. 2002.
[29] Company, N. I. *Session Guitarist - Picked Nylon*. [Online]. Available: https://www.native-instruments.com/en/products/komplete/guitar/session-guitarist-picked-nylon.
[30] MuseScore. *MuseScore*. [Online]. Available: https://musescore.com.
[31] Inc., A. *Logic Pro*. [Online]. Available: https://www.apple.com/logic-pro.
[32] Rouard, S., Massa, F., and Défossez, A. "Hybrid Transformers for Music Source Separation". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023.
[33] D'efossez, A. "Hybrid Spectrogram and Waveform Source Separation". In: *ArXiv* abs/2111.03600 (2021).
[34] Cano, E. et al. "Musical source separation: An introduction". In: *IEEE Signal Processing Magazine* 36 (2018), pp. 31–40.
[35] Roma, G. et al. "Music remixing and upmixing using source separation". In: *Proceedings of the 2nd AES Workshop on Intelligent Music Production*. 2016.
[36] Mart'inez-Ram'irez, M. A. et al. "Automatic music mixing with deep learning and out-of-domain data". In: *International Society for Music Information Retrieval Conference ISMIR*. 2022.
[37] Reiss, J. D. "Intelligent systems for mixing multichannel audio". In: *17th International Conference on Digital Signal Processing (DSP)*. 2011.
[38] Analyst-Prep. *Overfitting and Methods of Addressing it*. [Online]. Available: https://analystprep.com/study-notes/cfa-level-2/quantitative-method/overfitting-methods-addressing.
[39] Davis, S. and Mermelstein, P. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28 (1980), pp. 357–366.
[40] Deruty, E. *Intuitive understanding of MFCCs*. [Online]. Available: https://medium.com/derutycsl/intuitive-understanding-of-mfccs-836d36a1f779.
[41] Bartsch, M. and Wakefield, G. "Audio thumbnailing of popular music using chroma-based representations". In: *IEEE Transactions on Multimedia* 7 (2005), pp. 96–104.
[42] project, R. T. *Bringing Parallelism to the Web with River Trail*. [Online]. Available: https://intellabs.github.io/RiverTrail/tutorial.
[43] Cui, X. et al. "Multiscale Spatial-Spectral Convolutional Network with Image-Based Framework for Hyperspectral Imagery Classification". In: *Remote Sensing* 11 (2019), p. 2220.
[44] NeuralNet, M. *Make your Own Neural Network: Calculating the Output Size of Convolutions and Transpose Convolutions*. [Online]. Available:

https://makeyourownneuralnetwork.blogspot.com/2020/02/calculating-output-size-of-convolutions.html.
[45] FirelordPhoenix. *Computer Science Wiki MaxPoolSample2.png.* [Online]. Available: https://computersciencewiki.org/index.php?title=File:MaxpoolSample2.png.
[46] AIWiki. *Activation Function.* [Online]. Available: https://machine-learning.paperspace.com/wiki/activation-function.
[47] Su, Y. et al. "Environment Sound Classification Using a Two-Stream CNN Based on Decision-Level Fusion". In: *Sensors* 19 (2019).
[48] Phi, M. *Illustrated Guide to LSTM's and GRU's: A step by step explanation.* [Online]. Available: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.
[49] Roberts, A., Jesse H. Engel Colin Raffel, C. H., and Eck, D. "A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music". In: *CoRR* abs/1803.05428 (2018).
[50] Vaswani, A. et al. "Attention is All you Need". In: *Neural Information Processing Systems.* 2017.
[51] Alammar, J. *The Illustrated Transformer.* [Online]. Available: https://jalammar.github.io/illustrated-transformer.
[52] Gardner, J. et al. "MT3: Multi-Task Multitrack Music Transcription". In: *International Conference on Learning Representations ICLR.* 2022.
[53] Oja, E. and Hyvarinen, A. "Independent component analysis: algorithms and applications". In: *Neural networks* 13 (2000), pp. 411–430.
[54] Smaragdis, P. "Blind separation of convolved mixtures in the frequency domain". In: *Neurocomputing* 22 (1998), pp. 21–34.
[55] Casey, M. A. and Westner, A. "Separation of mixed audio sources by independent subspace analysis". In: *International Conference on Mathematics and Computing.* 2000.
[56] Vembu, S. and Baumann, S. "Separation of Vocals from Polyphonic Audio Recordings". In: *International Society for Music Information Retrieval ISMIR.* 2005.
[57] Virtanen, T. "Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria". In: *IEEE Transactions on Audio, Speech, and Language Processing* 15 (2007), pp. 1066–1074.
[58] Carabias-Orti, J. J. et al. "Nonnegative signal factorization with learnt instrument models for sound source separation in close-microphone recordings". In: *EURASIP Journal on Advances in Signal Processing* (2013), pp. 1–16.
[59] Virtanen, T. and Klapuri, A. "Separation of harmonic sound sources using sinusoidal modeling". In: *IEEE International Conference on Acoustics, Speech, and Signal Processing.(ICASSP).* 2000.
[60] Virtanen, T. "Algorithm for the separation of harmonic sounds with time-frequency smoothness constraint". In: *Digital Audio Effects Conference (DAFx).* 2003.
[61] Roweis, S. "One microphone source separation". In: *Advances in Neural Information Processing Systems Conference.* 2000.
[62] Wang, Y., Narayanan, A., and Wang, D. "On training targets for supervised speech separation". In: *IEEE/ACM transactions on Audio, Speech, and Language Processing* 22 (2014), pp. 1849–1858.
[63] Chandna, P. et al. "Monoaural Audio Source Separation Using Deep Convolutional Neural Networks". In: *International Conference Latent Variable Analysis and Signal Separation.* 2017.
[64] Perraudin, N., Balazs, P., and Søndergaard, P. L. "A fast Griffin-Lim algorithm". In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.* 2013.
[65] Gerkmann, T., Krawczyk-Becker, M., and Le Roux, J. "Phase Processing for Single-Channel Speech Enhancement: History and recent advances". In: *IEEE Signal Processing Magazine* 32 (2015), pp. 55–66.
[66] Takahashi, N. et al. "PhaseNet: Discretized Phase Modeling with Deep Neural Networks for Audio Source Separation". In: *Interspeech.* 2018.
[67] Grais, E. M., Sen, M. U., and Erdogan, H. "Deep neural networks for single channel source separation". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 2014.
[68] Uhlich, S., Giron, F., and Mitsufuji, Y. "Deep neural network based instrument extraction from music". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 2015.
[69] Ritter, S. et al. "Been There, Done That: Meta-Learning with Episodic Recall". In: *International Conference on Machine Learning.* 2018.

[70] Chen, F. et al. *Federated Meta-Learning with Fast Convergence and Efficient Communication*. 2019. arXiv: 1802.07876 [cs.LG].
[71] Giri, R., Isik, U., and Krishnaswamy, A. "Attention Wave-U-Net for Speech Enhancement". In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. 2019.
[72] Kaspersen, E. T., Kounalakis, T., and Erkut, C. "Hydranet: A Real-Time Waveform Separation Network". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020.
[73] Défossez, A. et al. "Music Source Separation in the Waveform Domain". In: *CoRR* abs/1911.13254 (2019).
[74] Miron, M., Janer, J., and Gómez, E. "Monaural Score-Informed Source Separation for Classical Music Using Convolutional Neural Networks". In: *International Society for Music Information Retrieval Conference ISMIR*. 2017.
[75] Cheuk, K. W. et al. *Jointist: Joint Learning for Multi-instrument Transcription and Its Applications*. 2022. arXiv: 2206.10805.
[76] Yang, M. et al. "Complex Transformer: A Framework for Modeling Complex-Valued Sequence". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020.
[77] Schlüter, J. and Grill, T. "Exploring data augmentation for improved singing voice detection with neural networks". In: *International Society for Music Information Retrieval ISMIR*. 2015.
[78] McFee, B., Humphrey, E. J., and Bello, J. P. "A software framework for musical data augmentation". In: *International Society for Music Information Retrieval ISMIR*. 2015.
[79] Weng, C. et al. "Deep neural networks for single-channel multi-talker speech recognition". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23 (2015), pp. 1670–1679.
[80] Hershey, J. R. et al. "Deep clustering: Discriminative embeddings for segmentation and separation". In: *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. 2016.
[81] Raffel, C. "Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching". PhD thesis. 2016.
[82] Goto, M. et al. "RWC Music Database: Popular, Classical and Jazz Music Databases". In: *International Society for Music Information Retrieval ISMIR*. 2002.
[83] Praetzel, E. *Mutopia project: Free sheet music for everyone*. [Online]. Available: https://www.mutopiaproject.org/index.html. 2000.
[84] Vincent, E., Gribonval, R., and Fevotte, C. "Performance measurement in blind audio source separation". In: *IEEE Transactions on Audio, Speech, and Language Processing* 14 (2006), pp. 1462–1469.
[85] Lambert, R. "Difficulty measures and figures of merit for source separation". In: *Proc. Int. Symp. ICA and BSS* (Nov. 1999).
[86] Févotte, C., Gribonval, R., and Vincent, E. "BSS EVAL Toolbox User Guide Revision 2.0". In: (Jan. 2005).
[87] Le Roux, J. et al. "SDR–half-baked or well done?" In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019.
[88] Mazzola, G. et al. "A Short History of MIDI: An Introduction". In: Jan. 2018, pp. 115–116.
[89] Producer, R. *Audio Synthesis 101 - An Introduction to Synth Programming for Electronic Music Producers*. [Online]. Available: https://www.renegadeproducer.com/audio-synthesis.html.
[90] Starostenko, O. and Lopez-Rincon, O. "A 3D Spatial Visualization of Measures in Music Compositions". In: *Technology, Science, and Culture: A Global Vision*. Mar. 2019, p. 127.
[91] Thickstun, J., Harchaoui, Z., and Kakade, S. M. "Learning Features of Music from Scratch". In: *ArXiv* abs/1611.09827 (2016). URL:
[92] Papers with Code. *Music Transcription on MusicNet*. [Online]. Available: https://paperswithcode.com/sota/music-transcription-on-musicnet. 2024.
[93] Kumar, K. et al. "MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis". In: *Neural Information Processing Systems*. 2019.
[94] Press, O., Smith, N. A., and Lewis, M. *Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation*. 2022.
[95] Su, W. et al. *VL-BERT: Pre-training of Generic Visual-Linguistic Representations*. 2020. arXiv: 1908.08530 [cs.CV].
[96] Devlin, J. et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *North American Chapter of the Association for Computational Linguistics*. 2019.

[97] Touvron, H. et al. "Going deeper with Image Transformers". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021.

[98] Benjamin Lefaudeux Francisco Massa, D. L. et al. *xformers: A modular and hackable transformer modelling library*. Available at https://github.com/facebookresearch/xformers. 2021.

[99] Hung, Y.-N., Wichern, G., and Le Roux, J. "Transcription is all you need: Learning to separate musical mixtures with score as supervision". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021.

[100] "Non-negative matrix factorization for speech/music separation using source dependent decomposition rank, temporal continuity term and filtering". In: *Biomedical Signal Processing and Control* 36 (2017), pp. 168–175.