NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
DIVISION OF SIGNALS, CONTROL AND ROBOTICS

Singing Voice Separation using Waveform-Level Deep Neural Networks

DIPLOMA THESIS

Panagiotis Papantonakis

Supervisor: Petros Maragos
Professor NTUA

COMPUTER VISION, SPEECH COMMUNICATION AND SIGNAL PROCESSING GROUP
Athens, November 2021

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

Διαχωρισμός Φωνητικών χρησιμοποιώντας Βαθιά Νευρωνικά Δίκτυα σε Επίπεδο Κυματομορφών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Παναγιώτης Παπαντωνάκης

Επιβλέπων: Πέτρος Μαραγκός
Καθηγητής Ε.Μ.Π.

ΕΡΓΑΣΤΗΡΙΟ ΟΡΑΣΗΣ ΥΠΟΛΟΓΙΣΤΩΝ, ΕΠΙΚΟΙΝΩΝΙΑΣ ΛΟΓΟΥ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ
Αθήνα, Νοέμβριος 2021

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

Διαχωρισμός Φωνητικών χρησιμοποιώντας Βαθιά Νευρωνικά Δίκτυα σε Επίπεδο Κυματομορφών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Παναγιώτης Παπαντωνάκης

Επιβλέπων: Πέτρος Μαραγκός
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 5η Νοεμβρίου, 2021.

....	....	....
Πέτρος Μαραγκός	Κωνσταντίνος Τζαφέστας	Γεράσιμος Ποταμιάνος
Καθηγητής Ε.Μ.Π.	Αναπληρωτής Καθηγητής Ε.Μ.Π.	Αναπληρωτής Καθηγητής Π.Θ.

Αθήνα, Νοέμβριος 2021

....
Ονομα
Διπλωματούχος Ηλεκτρολόγος Μηχανικός
και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © – All rights reserved Παναγιώτης Παπαντωνάκης, 2021.
Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Ο Διαχωρισμός Φωνητικών (Singing Voice Separation - SVS) είναι ένα σημαντικό πρόβλημα στην Ακρόαση Υπολογιστών το οποίο ερευνάται έντονα εδώ και πάρα πολλά χρόνια. Το πρόβλημα μπορεί να περιγραφεί ως η αυτόματη απομόνωση του φωνητικού μέρους από ένα δεδομένο μουσικό μείγμα, χωρίς πρότερη γνώση στις ιδιότητες των σημάτων που το απαρτίζουν. Πρόσφατα, έχει υπάρξει μια αύξηση τόσο στην ποσότητα όσο και στην ποιότητα των τεχνικών που εκτελούν τον διαχωρισμό στο πεδίο των κυματομορφών, με κάποια μοντέλα να πετυχαίνουν πολύ καλά αποτελέσματα. Σε αυτήν την εργασία πειραματιζόμαστε με δύο από τις καλύτερες βαθιές αρχιτεκτονικές στο πεδίο των κυματομορφών, χρησιμοποιώντας το σύνολο δεδομένων MUSDB18. Στο πρώτο μέρος επανυλοποιήσαμε το Wave-U-Net, μια βαθιά αρχιτεκτονική κωδικοποιητή-αποκωδικοποιητή με skip συνδέσεις, μαζί με μερικές επεκτάσεις που είχαν προταθεί από άλλες εργασίες. Στη συνέχεια, πραγματοποιήσαμε μια συγκεντρωτική έρευνα σε διαφορετικές εκδοχές του μοντέλου, ελέγχοντας τις επεκτάσεις μεμονωμένα ή συνδυαζόμενες μεταξύ τους, προκειμένου να ελεγχθεί η επίδρασή τους στην επίδοση του μοντέλου. Στο δεύτερο μέρος πειραματιστήκαμε με το Conv-TasNet, μια αρχιτεκτονική που μετασχηματίζει την κυματομορφή εισόδου σε έναν λανθάνοντα χώρο, κατάλληλο για τον διαχωρισμό, κατασκευάζει και εφαρμόζει μια πολλαπλασιαστική μάσκα για κάθε πηγή και έπειτα μετασχηματίζει το σήμα πίσω στο πεδίο του χρόνου, προτείνοντας πολλαπλές, νέες επεκτάσεις. Τα αρχικά, διερευνητικά πειράματα έδειξαν ότι μια παράλληλη, πολυζωνική τεχνική που χωρίζει το κωδικοποιημένο σήμα σε ζώνες του λανθάνοντα χώρου κι έπειτα επεξεργάζεται κάθε ζώνη ξεχωριστά, χρησιμοποιώντας πολλαπλούς διαχωριστές, μπορεί να είναι ευεργετική στο μοντέλο, αφού παρέχει μια αξιοσημείωτη αύξηση της επίδοσής του. Ως αποτέλεσμα, στη συνέχεια προχωρήσαμε σε μια εις βάθος ανάλυση της τεχνικής, σε σχέση με την εφαρμοσιμότητα και την κλιμακοσιμότητά της. Τα αποτελέσματα έδειξαν ότι η προτεινόμενη μέθοδος επιτυγχάνει ανταγωνιστικά αποτελέσματα, αξιοποιώντας τα διαχωριστικά χαρακτηριστικά της κάθε ζώνης και παράγοντας εξειδικευμένους διαχωριστές, ενώ διατηρεί τον αριθμό των εκπαιδευούμενων παραμέτρων σταθερό. Στο τελευταίο μέρος της εργασίας συνδυάζουμε την προτεινόμενη πολυζωνική επέκταση με δύο διαφορετικούς κωδικοποιητές που έχουν προταθεί σε άλλες έρευνες· έναν εκπαιδεύσιμο που συνδυάζει χαρακτηριστικά που προέρχονται τόσο από κυματομορφές όσο και από το χρονο-συχνοτικό πεδίο κι έναν σταθερό που μοντελοποιεί το ακουστικό σύστημα του ανθρώπου χρησιμοποιοώντας μια gammatone συστοιχία φίλτρων. Παρόλο που τα αποτελέσματα για τον πρώτο κωδικοποιητή δεν έδειξαν να υπάρχει κάποια βελτίωση, για τον δεύτερο, τα αποτελέσματα δείχνουν ότι υπάρχει αύξηση της επίδοσης όταν ο πολυζωνικός διαχωρισμός συνδυάζεται με ένα γραμμικό επίπεδο για επιλογή ζωνών.

Keywords — Διαχωρισμός Πηγών, Διαχωρισμός Φωνητικών, Conv-TasNet, Wave-U-Net, Συνελικτικά Νευρωνικά Δίκτυα

Abstract

Singing Voice Separation (SVS) is an important task of Computer Audition, that has been studied intensively for many years. The problem can be described as the automatic isolation of the vocal component from a given musical mixture, without prior knowledge on the properties of the participating signals. Recently, there has been an increase in both the quantity and quality of SVS techniques in the waveform domain, with some models achieving state-of-the-art results. In this thesis we experiment with two of the top performing deep architectures in the waveform domain, using the MUSDB18 dataset. In the first part we reimplement Wave-U-Net, a deep autoencoder architecture with skip connections, along with several modifications, already proposed by other studies. We then perform an ablation study on different model configurations, by enabling individual or multiple modifications each time, in order to examine their effect on the model's performance. In the second part we experiment with Conv-TasNet, an architecture that transforms the waveform input to a latent space, suitable for separation, constructs and applies a multiplicative mask for each source and then transforms the signal back to the time domain, proposing multiple novel modifications. Preliminary, exploratory experiments indicated that a parallel multi-band separation technique that splits the encoded signal in latent space bands and then processes each band individually, using multiple separators, could be beneficial to the model, as it provided a significant performance boost. As a result, we subsequently proceeded with an in-depth analysis of it, regarding its efficacy and scalability. The results show that the proposed method achieves competitive performance by taking advantage of the discriminative characteristics of each band and generating specialised separators, while keeping the amount of trainable parameters the same. In the last part of the thesis, we combine the proposed multi-band modification with two different encoders proposed in other studies, a trainable one that combines features derived from both waveform and time-frequency domains and a fixed one that models the human auditory system using a gammatone filterbank. Although the results for the former encoder do not display some kind of improvement, the results for the latter point towards performance improvements, with the assistance of a linear layer for band selection.

Keywords — Source Separation, Singing Voice Separation, Conv-TasNet, Wave-U-Net, Convolutional Neural Networks

Ευχαριστίες

Στην παρούσα διπλωματική εργασία συγκεντρώνεται και παρουσιάζεται η έρευνα που πραγματοποίησα στο Εργαστήριο Όρασης Υπολογιστών, Επικοινωνίας Λόγου και Επεξεργασίας Σήματος τον τελευταίο ενάμιση χρόνο. Ταυτόχρονα, με την εργασία αυτή ολοκληρώνεται και η φοίτησή μου στη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, ένα εξαετές ταξίδι που αν και περιλάμβανε αρκετή πίεση, κούραση και άγχος, με αντάμειψε με καινούργιες γνώσεις και πολλές ευχάριστες εμπειρίες που θα με συνοδεύουν στη μετέπειτα ζωή μου. Με την αφορμή που μου δίνεται, θα ήθελα να ευχαριστήσω:

Τον Καθηγητή κ. Πέτρο Μαραγκό για τη διδασκαλία των τριών μαθημάτων σημάτων (Ψηφιακή Επεξεργασία Σήματος, Όραση Υπολογιστών, Αναγνώριση Προτύπων) που έπαιξαν καθοριστικό ρόλο στην ανάπτυξη των ενδιαφερόντων μου και στην απόφαση να ασχοληθώ με το συγκεκριμένο αντικείμενο, καθώς και για την ευκαιρία που μου έδωσε να εκπονήσω τη διπλωματική μου εργασία στο εργαστήριό του.

Τον διδακτορικό φοιτητή Χρήστο Γαρούφη, για την πολύ καλή συνεργασία που είχαμε όλη αυτήν την περίοδο. Η καθοδήγηση, οι συμβουλές, οι προτάσεις και οι διορθώσεις του ήταν πολύτιμες τόσο στην διεκπεραίωση της έρευνας και στη διαμόρφωση της εργασίας όσο και στην αντιμετώπιση των όποιων προβλημάτων, αποριών ή εμποδίων δημιουργούνταν.

Τους γονείς μου, για την αμέριστη στήριξη που μου προσφέρουν, όλα αυτά τα χρόνια, την αδελφή και συγκάτοικό μου που είναι σταθερά δίπλα μου, στις ευχάριστες και δυσάρεστες στιγμές μου, την κοπέλα μου για την υποστήριξη και ουσιαστική κατανόηση που μου έδειξε και τους φίλους μου, και ειδικά «Το Παραδοσιακό», για τις αμέτρητες αξέχαστες εμπειρίες που μοιραστήκαμε.

Παπαντωνάκης Παναγιώτης
Νοέμβριος 2021

Contents

List of Figures

List of Tables

Chapter 1

Εκτεταμένη περίληψη

Contents

1.1 Εισαγωγή

1.1.1 Ορισμός Προβλήματος

Ο Διαχωρισμός Πηγών ορίζεται ως η διαδικασία αποσύνδεσης των σημάτων των πηγών που συνιστούν ένα δεδομένο μείγμα σημάτων, με σκοπό την αφαίρεση ανεπιθύμητων παρεμβολών από κάποιο σήμα ενδιαφέροντος ή την απομόνωση των επιμέρους πηγαίων σημάτων για περαιτέρω επεξεργασία.

Αυτή η εργασία ασχολείται με μια ειδική περίπτωση του διαχωρισμού πηγών ηχητικών σημάτων, συγκεκριμένα του Διαχωρισμού Φωνητικών, στην οποία το ηχητικό μείγμα είναι ένα τραγούδι. Στόχος του προβλήματος είναι, δεδομένου του μείγματος και χωρίς άλλες πληροφορίες για τις πηγές, όπως ο τύπος της μουσικής, ο αριθμός των τραγουδιστών κ.ά., να επιτευχθεί διαχωρισμός του φωνητικού μέρους από το ορχηστρικό, που αποτελείται από ένα συνδυασμό μουσικών οργάνων.

Το πρόβλημα του Διαχωρισμού Φωνητικών συνοδεύεται από μερικές προκλήσεις που δυσκολεύουν τη λύση του από έναν υπολογιστή. Πρώτον, λόγω της τεράστιας ποικιλομορφίας της μουσικής, είναι πολύ δύσκολο έως αδύνατο ο διαχωρισμός να πραγματοποιηθεί με τεχνικές που στηρίζονται σε ένα σύνολο κανόνων, όπως συμβαίνει σε προβλήματα γραμματικών και συντακτικού γλωσσών. Δεύτερον, στο συγκεκριμένο πρόβλημα το ηχητικό μείγμα είναι είτε μονοφωνικό, που σημαίνει ότι έχει μόνο ένα κανάλι ήχου, ή πολυφωνικό, χωρίς όμως να υπάρχει πληροφορία για τα μικρόφωνα που χρησιμοποιήθηκαν. Επομένως, δεν υπάρχουν τα απαιτούμενα στοιχεία για τον χωρικό εντοπισμό των πηγών, αποκλείοντας τη δυνατότητα εφαρμογής τεχνικών χωρικού φιλτραρίσματος, όπως το "beamforming". Τρίτον, στη μουσική τα πηγαία σήματα είναι πολύ συσχετισμένα. Ως αποτέλεσμα, δεν υπάρχει ένα απλό, αξιόπιστο χαρακτηριστικό, όπως η συχνότητα ή το πλάτος του σήματος που μπορεί να χρησιμοποιηθεί ως διαχωριστικό στοιχείο, για την αποσύνδεση των πηγαίων σημάτων. Κάτι τέτοιο δεν ισχύει για μια πληθώρα προβλημάτων αποθορυβοποίησης σημάτων, στα οποία ο θόρυβος μπορεί να μοντελοποιηθεί με βάση διάφορα στατιστκά χαρακτηριστικά, δίνοντας τη δυνατότητα χρήσης διάφορων τεχνικών από την Ψηφιακή Επεξεργασία Σήματος (ΨΕΣ).

1.1.2 Στόχοι και Συνεισφορές αυτής της Διπλωματικής

Σε αυτή τη διπλωματική εργασία πραγματοποιείται μια ενδελεχής έρευνα στις υπάρχουσες τεχνικές στον τομέα του διαχωρισμού φωνητικών, εστιάζοντας στις νεότερες τεχνικές που βασίζονται σε βαθιά νευρωνικά δίκτυα. Οι συνεισφορές της διπλωματικής χωρίζονται σε δύο μέρη, αντίστοιχα με την εκάστοτε βασική αρχιτεκτονική: Όσον αφορά στην πρώτη, το Wave-U-Net,

- Υλοποιήθηκαν διάφορες υπάρχουσες τροποποιήσεις και εξετάστηκε το κατά πόσο αυτές συνεργάζονται και η επίδρασή τους στην επίδοση του μοντέλου.

Σχετικά με τη δεύτερη, το Conv-TasNet,

- Υλοποιήθηκε και εκπαιδεύτηκε μία εκδοχή του Conv-TasNet, μιας αρχιτεκτονικής που πετυχαίνει πάρα πολύ καλά αποτελέσματα στο πρόβλημα του διαχωρισμού μουσικών πηγών.

- Προτάθηκαν διάφορες καινούργιες επεκτάσεις στο μοντέλο, εστιάζοντας σε μια πολυζωνική επέκταση, η οποία χωρίζει την λανθάνουσα αναπαράσταση σε ομάδες και επεξεργάζεται την κάθε ομάδα χωριστά, χρησιμοποιώντας πολλούς διαχωριστές.

- Εξετάστηκε κατά πόσον η προτεινόμενη τεχνική κλιμακώνει, χρησιμοποιώντας διαφορετικό αριθμό ομάδων και δύο διαφορετικές εκδοχές της τεχνικής.

- Συνδυάστηκε η προτεινόμενη τεχνική με έναν καλύτερο κωδικοποιητή και μια προκαθορισμένη συστοιχία φίλτρων για να ελεγχθεί κατά πόσον η τεχνική μπορεί να γενικευτεί σε άλλες αρχιτεκτονικές.

1.2 Wave-U-Net

Σε αυτήν την ενότητα α) θα παρουσιαστεί το Wave-U-Net, ένα νευρωνικό δίκτυο που προτάθηκε στο [61] ως μια λύση για το πρόβλημα του διαχωρισμού μουσικής και β) θα εξεταστούν η λειτουργία και η διασυνδεσιμότητα κάποιων επεκτάσεων της βασικής αρχιτεκτονικής που προτάθηκαν από τις εργασίες [44] και [29]. Οι στόχοι μας σχετικά με αυτή την αρχιτεκτονική ήταν κυρίως η εξοικείωση με το πρόβλημα, τα χρησιμοποιούμενα εργαλεία, τις ρυθμίσεις υπερπαραμέτρων και η διεκπεραίωση μιας έρευνας απάνω στις υπάρχουσες επεκτάσεις της, αντί της εισαγωγής πρωτότυπων βελτιώσεων σε αυτήν.

1.2.1 Βασική Αρχιτεκτονική

Το Wave-U-Net [61] είναι ένα μοντέλο που επεξεργάζεται ηχητικά σήματα στο πεδίο του χρόνου. Αυτό σημαίνει ότι, αφ'ενός, το μοντέλο κάνει μια απευθείας εκτίμηση των δειγμάτων των πηγαίων σημάτων αντί να εφαρμόζει μια μάσκα στην είσοδο ή σε μια λανθάνουσα αναπαράσταση και αφ'ετέρου, οι λανθάνουσες αναπαραστάσεις και οι χάρτες χαρακτηριστικών, παρόλο που μπορεί να είναι πολυκαναλικοί, είναι μονοδιάστατοι και υφίστανται επεξεργασία από 1Δ επίπεδα.

Το Wave-U-Net ακολουθεί μια αρχιτεκτονική αυτοκωδικοποιητή (autoencoder), αποτελούμενη από τα εξής τέσσερα μέρη:

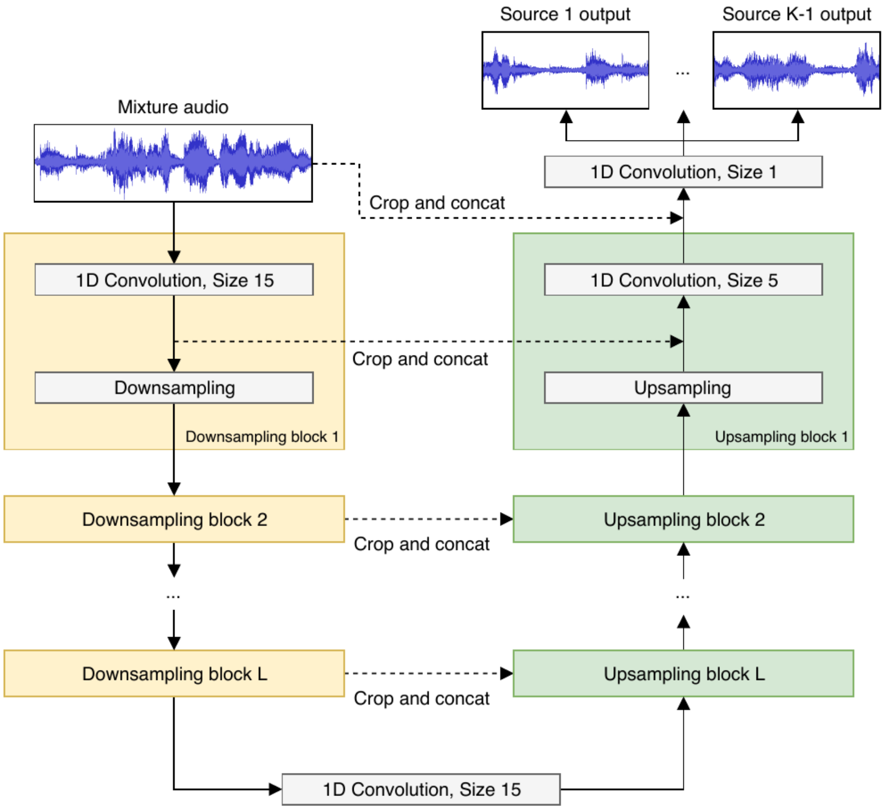
- Ένα μονοπάτι κωδικοποίησης ή υποδειγματοληψίας, το οποίο παίρνει το αρχικό σήμα σαν είσοδο και το επεξεργάζεται κατ'επανάληψη, μειώνοντας τα δείγματά του, μέχρι να παραχθεί μια πυκνή λανθάνουσα αναπαράσταση.

- Το σημείο συμφόρησης (bottleneck), στο οποίο και εκτελείται η όποια επεξεργασία επί της πυκνής αναπαράστασης.

- Ένα μονοπάτι αποκωδικοποίησης ή υπερδειγματοληψίας, το οποίο λειτουργεί αντίθετα από τον κωδικοποιητή, αφού δέχεται την επεξεργασμένη αναπαράσταση και κατ'επανάληψη συνδυάζει χαρακτηριστικά από προηγούμενα επίπεδα και αυξάνει τον αριθμό των δειγμάτων μέχρι να αποκατασταθεί η αναπαράσταση στην αρχική της ανάλυση.

- Ένα επίπεδο εξόδου που εκτελεί το επιθυμητό έργο, δηλαδή ο διαχωρισμός των πηγών.

Πιο αναλυτικά, το Wave-U-Net έχει βάθος L επιπέδων, που σημαίνει ότι κάθε ένα από τα μονοπάτια κωδικοποίησης και αποκωδικοποίησης περιλαμβάνει L μπλοκ επεξεργασίας. Κάθε ένα μπλοκ περιέχει ένα 1Δ συνελικτικό επίπεδο με LeakyReLU ενεργοποίηση και διαδικασία επαναδειγματοληψίας. Στο μονοπάτι κωδικοποίησης το συνελικτικό επίπεδο προηγείται της υποδειγματοληψίας, ενώ στο μονοπάτι αποκωδικοποίησης συμβαίνει το αντίθετο· η εξαγωγή χαρακτηριστικών ακολουθεί την υπερδειγματοληψία. Το σημείο συμφόρησης και το επίπεδο εξόδου αποτελεούνται από ένα μοναδικό 1Δ συνελικτικό επίπεδο, χωρίς να εμπλέκεται διαδικασία δειγματοληψίας και με LeakyReLU και tanh ενεργοποιήσεις αντίστοιχα.



Σχήμα 1.2.1: Wave-U-Net με K πηγές, βάθος L και με την προσθήκη a) επιπέδου εξόδου με υπολογισμό διαφοράς και b) μεγαλύτερου πλαισίου εισόδου [61].

Το μοντέλο δέχεται κατακερματισμένα μείγματα σημάτων $M \in [-1,1]^{L_m \times C}$, όπου L_m είναι το μήκος του τμήματος σε δείγματα και C ο αριθμός των καναλιών ήχου. Το πρώτο block αυξάνει τον αριθμό των καναλιών από τον αρχικό C σε ένα σταθερό αριθμό F_c. Τα υπόλοιπα συνελικτικά επίπεδα του μονοπατιού κωδικοποίησης, καθώς και του σημείου συμφόρησης αυξάνουν τον αριθμό των καναλιών κατά F_c, στοχεύοντας στην εξαγωγή ολοένα και περισσότερο πλούσιων σε πληροφορία χαρακτηριστικών για να σχηματίσουν και να επεξεργαστούν την πυκνή αναπαράσταση. Στο βαθύτερο σημείο, μετά το σημείο συμφόρησης, η λανθάνουσα αναπαράσταση έχει $(L+1) \cdot F_c$ κανάλια συνολικά. Το μονοπάτι υπερδειγματοληψίας δουλεύει με τον αντίθετο τρόπο· τα συνελικτικά επίπεδα μειώνουν τον αριθμό των καναλιών έτσι ώστε αυτός να ταιριάζει με τα αντίστοιχα μπλοκ του μονοπατιού υποδειγματοληψίας και σταδιακά να αποκαθίσταται ο αριθμός των καναλιών της αναπαράστασης. Με συνολικά L μπλοκ υπερδειγματοληψίας, τα κανάλια μειώνονται σε F_c. Στην αρχική εργασία οι συνελίξεις των δύο πρώτων μερών της αρχιτεκτονικής έχουν μέγεθος πυρήνα ίσο με 15 και του τρίτου ίσο με 5.

Μεταξύ των μπλοκ του ίδιου βάθους υπάρχουν skip συνδέσεις οι οποίες επιτρέπουν στην πληροφορία από το μονοπάτι υποδειγματοληψίας να φτάνει στο μονοπάτι υπερδειγματοληψίας ανεμπόδιστη, παραλείποντας ενδιάμεση επεξεργασία. Τα χαρακτηριστικά που έρχονται από τον κωδικοποιητή συνενώνονται με αυτά του αποκωδικοποιητή προτού υποστούν επεξεργασία από το συνελικτικό επίπεδο. Αυτό γίνεται για δύο λόγους: Πρώτον, με τον τρόπο αυτό λεπτομέρειες υψηλού επιπέδου, που μπορεί να είχαν χαθεί εξαιτίας της μείωσης δειγματοληψίας, φτάνουν κατευθείαν στο σημείο αποκατάστασης του σήματος, συμβάλοντας ευεργετικά στη σωστή ανακατασκευή των σημάτων. Δεύτερον, διευκολύνεται η εκπαίδευση των πρώτων επιπέδων, αντιμετωπίζοντας μερικώς το πρόβλημα των μειούμενων παραγώγων (vanishing gradient problem). Ουσιαστικά, το πρόβλημα αμβλύνεται αφού με τη χρήση αυτών των συνδέσεων τα πρότερα στάδια δέχονται παραγώγους από δύο μονοπάτια (το κανονικό και το skip).

Τέλος, το επίπεδο εξόδου δέχεται την πολυκαναλική αναπαράσταση με ανάλυση ίδια με του μείγματος εισόδου και εκτελεί τον διαχωρισμό με ένα συνελικτικό επίπεδο που αλλάζει τον αριθμό των καναλιών σε $K \cdot C$, όπου K ο αριθμός των πηγών. Το σχήμα 1.2.1 απεικονίζει την παραπάνω αρχιτεκτονική.

Σχετικά με τη διαδικασία αλλαγής δειγματοληψίας, τόσο η υποδειγματοληψία όσο και η υπερδειγματοληψία αλλάζουν την ανάλυση του χάρτη χαρακτηριστικών με παράγοντα 2, με τον μονοπάτι μείωσης να υποδιπλασιάζει την ανάλυση και το μονοπάτι αύξησης να τη διπλασιάζει. Αυτό επιτρέπει στα συνελικτικά επίπεδα να εξάγουν χαρακτηριστικά σε πολλαπλές κλίμακες, χωρίς την ανάγκη χρήσης πυρήνων διαφορετικού μεγέθους, που θα επέφερε μια αύξηση του υπολογιστικού κόστους. Επίσης, το γεγονός ότι τα πιο βαθιά επίπεδα επεξεργάζονται χάρτες χαρακτηριστικών μειωμένης ανάλυσης σε σχέση με πιο ρηχά επίπεδα, βοηθά στη διατήρηση των υπολογιστικών απαιτήσεων σε λογικά επίπεδα, παρόλο που τα πρώτα εφαρμόζουν στα δεδομένα σημαντικά περισσότερα φίλτρα από τα δεύτερα. Έτσι, για ένα τμήμα εισόδου με μήκος L_m, η αναπαράσταση στο σημείο συμφόρησης θα έχει μήκος $L_m/2^L$ δείγματα. Η αλλαγή δειγματοληψίας είναι υλοποιημένη ως μια απλή αποδεκάτιση (decimation) για τη μείωση και γραμμική παρεμβολή για την αύξηση.

1.2.2 Επεκτάσεις

Μεγαλύτερο Πλαίσιο Εισόδου

Η πρώτη επέκταση, που προτάθηκε από την αρχική εργασία [61], αντιμετωπίζει τη μείωση του μεγέθους των χαρτών χαρακτηριστικών, ένα γνωστό πρόβλημα των συνελικτικών επιπέδων.

Πιο συγκεκριμένα, επειδή για την πραγματοποίηση της πράξης της διακριτής συνέλιξης πρέπει οι πυρήνες να χωράνε εξ ολοκλήρου μέσα στο σήμα, οι χάρτες χαρακτηριστικών εξόδου είναι μικρότεροι από αυτούς της εισόδου. Για την αποφυγή αυτής της μείωσης που μπορεί να έχει δυσμενή αποτελέσματα για την εκπαίδευση και την επίδοση του μοντέλου, χρησιμοποιείται ένα επίπδο padding. Ομως, ανεξάρτητα από την τεχνική padding που ακολουθείται, η προστιθέμενη πληροφορία είναι εσφαλμένη, κάτι που μπορεί να φθείρει τις συνελίξεις κοντά στα άκρα του σήματος. Παρόλο που υπάρχουν περιπτώσεις που η επίδραση του padding είναι αμελητέα, στην επεξεργασία μουσικής, στην οποία τα σήματα λόγω του μεγάλου μεγέθους τους κατακερματίζονται σε τμήματα λίγων δευτερολέπτων, η χρήση padding ουσιαστικά προσθέτει λάθος ήχους πριν και μετά από κάθε τμήμα.

Αυτό το πρόβλημα είναι πολύ έντονο στο Wave-U-Net, επειδή ο λόγος του μεγέθους συνελικτικών πυρήνων προς το μεγέθος χάρτη χαρακτηριστικών, κι άρα η ποσότητα διεφθαρμένης πληροφορίας αυξάνεται όσο μεταβαίνουμε σε βαθύτερα επίπεδα. Μάλιστα, ανάλογα με τις υπερπαραμέτρους του μοντέλου, αυτός ο λόγος μπορεί να γίνει και μεγαλύτερος της μονάδας, για πολλαπλά επίπεδα, που σημαίνει ότι κάθε στοιχείο του χάρτη χαρακτηριστικών εξόδου είναι επηρεασμένο από το padding.

Οπότε, εφόσον η χρήση padding για την αποφυγή μείωσης των χαρτών χαρακτηριστικών δημιουργεί περισσότερα προβλήματα στο μοντέλο, προτείνεται απλά η χρήση μεγαλύτερων σημάτων εισόδου. Πρακτικά, για ένα δεδομένο τμήμα εισόδου με μήκος L_m, το πηγαίο σήμα που παράγει το μοντέλο θα έχει μικρότερο μήκος $L_s < L_m$, που σημαίνει ότι για να πετύχει την ίδια έξοδο ένα μοντέλο με μεγαλύτερο πλαίσιο εισόδου θα χρειαστεί μεγαλύτερη είσοδο, αυξάνοντας έτσι τις ανάγκες σε μνήμη του μοντέλου.

Για την ενσωμάτωση μεγαλύτερων παραθύρων στο Wave-U-Net, η συνένωση απαιτεί οι χάρτες χαρακτηριστικών του μονοπατιού κωδικοποίησης να περικόπτονται στο μέγεθος των αντίστοιχων χαρτών του μονοπατιού αποκωδικοποίησης.

Επίπεδο Εξόδου με Υπολογισμό Διαφοράς

Αυτή η επέκταση, που επίσης προτάθηκε στην αρχική εργασία [61], χρησιμοποιεί μια υπόθεση για τη φύση των δεδομένων προκειμένου να απλοποιηθεί το επίπεδο εξόδου. Συγκεκριμένα, εικάζεται ότι τα πηγαία σήματα συνδυάζονται προσθετικά, κάτι που ισχύει για το σύνολο δεδομένων MUSDB18 [50], αφού το μείγμα προκύπτει ως το άθροισμα των επιμέρους πηγαίων σημάτων, αλλά δεν είναι απαραίτητο ότι ισχύει για κάθε είδος δεδομένων.

Σχετικά με την υλοποίηση της επέκτασης, θεωρώντας ένα μείγμα $\mathbf{M}$ αποτελούμενο από K πηγαία σήματα $\mathbf{S}_j, j = 1 \dots K$, για το οποίο ισχύει $\mathbf{M} = \sum_{j=1}^{K} \mathbf{S}_j$, το επίπεδο εξόδου προβλέπει μόνο $K - 1$ πηγαία σήματα και υπολογίζει το τελευταίο $\widehat{\mathbf{S}}_K = \mathbf{M} - \sum_{j=1}^{K-1} \widehat{\mathbf{S}}_j$. Περιορίζοντάς το με αυτόν τον τρόπο, το μοντέλο δεν χρειάζεται να μάθει αυτόν τον κανόνα μέσω εκπαίδευσης, επιταχύνοντας τη διαδικασία και βελτιώνοντας έτσι την επίδοσή του.

Σημείο Συμφόρησης με Αναδρομικά Νευρωνικά Δίκτυα (RNN)

Στο [29] προτείνεται η προσθήκη ενός αναδρομικού επιπέδου, όπως LSTM ή BiLSTM, στο σημείο συμφόρησης, πριν από το συνελικτικό επίπεδο.

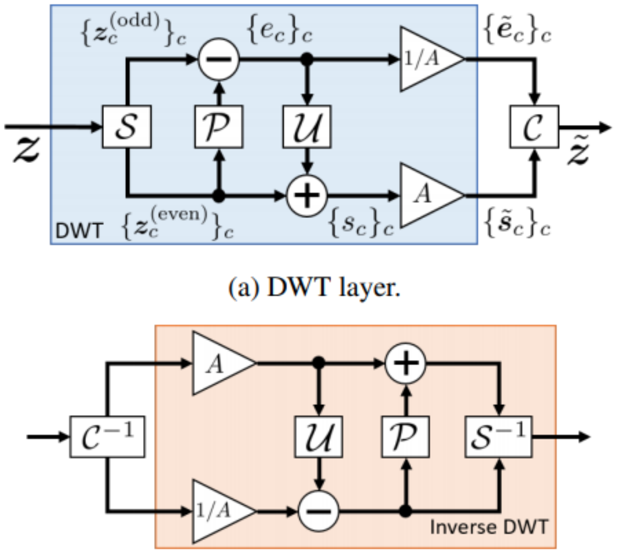
Το κίνητρο πίσω από αυτήν την επέκταση είναι ότι τα συνελικτικά επίπεδα, λόγω της δομής τους, έχουν ένα μικρό και πεπερασμένο δεκτικό πεδίο (receptive field). Επομένως, μπορούν να επεξεργάζονται μόνο τοπικές συσχετίσεις του σήματος και να ανακαλύπτουν τοπικά μοτίβα. Η υπάρχουσα λύση αξιοποίησης της μείωσης δειγματοληψίας για να αυξηθεί το δεκτικό πεδίο των συνελίξεων, αν και λειτουργική, απαιτεί πολύ βαθιές και σύνθετες αρχιτεκτονικές που είναι αργές και δύσκολες να εκπαιδευτούν και δημιουργεί πολύ αφαιρετικές λανθάνουσες αναπαραστάσεις, που μπορεί να μειώσουν τη συνολική επίδοση εξαιτίας της απώλειας πληροφορίας. Αντιθέτως, τα αναδρομικά επίπεδα διαθέτουν ένα απεριόριστο δεκτικό πεδίο κι άρα μπορούν να ενσωματωθούν στο σημείο συμφόρησης για να λύσουν το παραπάνω πρόβλημα. Αναμένεται ότι η χρήση αυτών των επιπέδων θα επιτρέψει τη χρήση πιο ρηχών μοντέλων, μειώνοντας τις παραμέτρους και την πολυπλοκότητα του μοντέλου και θα βελτιώσουν την επίδοσή του, αφού ειδικεύονται στην επεξεργασία ακολουθιακών δεδομένων, όπως τα σήματα ήχου.

Ενσωμάτωση Διακριτού Μετασχηματισμού Κυματιδίου

Η τελευταία βελτίωση για την αρχιτεκτονική είναι η χρήση Διακριτού Μετασχηματισμού Κυματιδίου (Discrete Wavelet Transform, DWT) στα μπλοκ αλλαγής δειγματοληψίας των μονοπατιών κωδικοποίησης και αποκωδικοποίησης [44].

Η ιδέα πίσω από αυτήν την επέκταση είναι ότι η διαδικασία αποδεκάτισης δημιουργεί επικάλυψη (aliasing) και δεν είναι πλήρως αναστρέψιμη, που σημαίνει ότι κομμάτια των χαρτών χαρακτηριστικών χάνονται. Αυτά τα δύο προβλήματα μπορούν να μειώσουν τη συνολική επίδοση του μοντέλου, καθώς η επικάλυψη φθείρει την πληροφορία, δημιουργώντας ηχητικά σφάλματα που δεν μπορούν να απομακρυνθούν από το υπόλοιπο μοντέλο και η απορριφθείσα πληροφορία μπορεί να είναι σημαντική για το διαχωρισμό των σημάτων.

Η χρήση του μετασχηματισμού κυματιδίου σαν διαδικασία αλλαγής δειγματοληψίας μπορεί να λύσει και



(a) DWT layer.

(b) Inverse DWT layer.

Σχήμα 1.2.2: Μπλοκ διάγραμμα των προτεινόμενων επιπέδων. Το C και το S επισημαίνουν τις πράξεις συνένωσης και χωρισμούς, αντίστοιχα, ενώ τα C^{-1} και S^{-1} είναι οι αντίστροφες πράξεις [44].

τα δύο αυτά θέματα, καθώς διαθέτει φίλτρο αντι-επικάλυψης (anti-aliasing) και μπορεί να αντιστραφεί πλήρως. Η ενσωμάτωση του μετασχηματισμού στην αρχιτεκτονική γίνεται δημιουργώντας δύο μονάδες αλλαγής δειγματοληψίας, ένα για τον ευθύ και έναν για τον αντίστροφο μετασχηματισμό, που αντικαθιστούν τους αρχικούς. Διαγράμματα των δύο μονάδων φαίνονται στο Σχήμα 1.2.2.

1.2.3 Πειράματα

Πειραματική Διάταξη

Για τα πειράματα χρησιμοποιήθηκε το σύνολο δεδομένων MUSDB18 [50], που είναι το πιο δημοφιλές στις εργασίες για διαχωρισμό μουσικής. Περιλαμβάνει τραγούδια από τα σύνολα δεδομένων MedleyDB [8] και DSD100 [45] καθώς και από άλλες πηγές. Συγκεκριμένα, έχει 150 τραγούδια διαφόρων μουσικών ειδών, με συνολική διάρκεια 10 ωρών. Τα τραγούδια είναι ολόκληρα, σε μορφή στέρεο (2 κανάλια ήχου), αποθηκευμένα σε υψηλή ποιότητα και με ρυθμό δειγματοληψίας 44.1kHz. Εκτός του μουσικού μείγματος, το σύνολο δεδομένων παρέχει τα 4 επιμέρους πηγαία σήματα κάθε τραγουδιού, που αντιστοιχούν σε 4 προκαθορισμένες κατηγορίες μουσικών οργάνων (φωνητικά, μπάσο, ντραμς, υπόλοιπα όργανα).

Σε αυτά τα πειράματα, το MUSDB18 χρησιμοποιήθηκε με 75-25 χωρισμό σε δεδομένα εκπαίδευσης-επαλήθευσης, με τα τραγούδια σε μορφή stereo (2 κανάλια) και με ρυθμό δειγματοληψίας 22.05kHz. Ολα τα μοντέλα εξάγουν ένα ηχητικό τμήμα μήκους 16384 δειγμάτων, δηλαδή περίπου 0.74 δευτερόλεπτα. Τα μοντέλα εκπαιδεύτηκαν με τη συνάρτηση κόστους L2 για 50 εποχές και χρησιμοποιήθηκε. και επιπρόσθετα η εκπαίδευση τερματιζόταν σε περίπτωση που το σφάλμα επαλήθευσης δεν βελτιωνόταν για 20 συνεχόμενες εποχές. Το σφάλμα υπολογιζόταν ως ο μέσος όρος των σφαλμάτων της φωνητικής και ορχηστρικής συνιστώσας. Τα μοντέλα με υπολογισμό διαφοράς στο επίπεδο εξόδου προέβλεπαν το ορχηστρικό μέρος και υπολόγιζαν το φωνητικό, ως τη διαφορά του ορχηστρικού μέρους από το μείγμα.

	L	F_c	#params	Input Context	Difference Layer	LSTM	DWT
M1	12	24	6.07M	✓	×	×	×
M2	6	48	3.63M	✓	×	×	×
M3	12	24	13.45M	✓	×	✓	×
M4	9	32	10.13M	×	×	✓	×
M5	12	24	6.07M	✓	✓	×	×
M6	12	24	13.45M	✓	✓	✓	×
M7	12	24	7.12M	✓	✓	×	✓
M8	9	32	10.98M	✓	✓	✓	✓

Πίνακας 1.1: Παραμετροποίηση των Wave-U-Net μοντέλων που εκπαιδεύτηκαν.

Για την αξιολόγηση των αποτελεσμάτων χρησιμοποιήθηκαν οι μετρικές Λόγος Σήματος προς Παραμόρφωση (Signal to Distortion Ratio, SDR), Λόγος Σήματος προς Σφάλματα (Signal to Artifact Ratio, SAR) και Λόγος Σήματος προς Παρεμβολές (Signal to Interference Ratio, SIR), αξιοποιώντας το πρωτόκολλο διάμεσος-των-διαμέσων που παρουσιάστηκε στο [62]. Σύμφωνα με αυτό το πρωτόκολλο οι μετρικές υπολογίζονται για τις εκτιμώμενες φωνητικές και ορχηστρικές συνιστώσες κάθε ηχητικού τμήματος. Στη συνέχεια, οι τμηματικές μετρικές συγκεντρώνονται για κάθε τραγούδι υπολογίζοντας τη διάμεσό τους και τέλος υπολογίζεται η διάμεσος όλων των τραγουδιών του συνόλου δεδομένων, δίνοντας μια και μόνο τιμή.

Στον Πίνακα 1.1 παρουσιάζονται τα μοντέλα που εκπαιδεύτηκαν.

		M1	M2	M3	M4	M5	M6	M7	M8
Voc.	SDR	4.48	4.43	4.77	4.60	4.78	4.52	**5.30**	5.09
	SIR	**12.35**	10.53	12.07	11.62	10.93	10.87	11.81	11.90
	SAR	5.28	5.29	5.61	5.74	5.70	5.57	5.96	**6.01**
Acc.	SDR	9.99	10.11	10.10	10.19	10.15	10.15	**10.84**	10.81
	SIR	13.80	13.77	14.37	13.78	13.95	13.95	**15.53**	14.83
	SAR	13.15	13.02	12.91	13.30	12.98	13.10	13.18	**13.50**

Πίνακας 1.2: Αποτελέσματα για τα μοντέλα M1-M8. Οι έντονες τιμές υποδεικνύουν την καλύτερη επίδοση μεταξύ των μοντέλων.

Αποτελέσματα και Σχολιασμός

Ο Πίνακας 1.2 περιλαμβάνει τα αποτελέσματα. Σχετικά με αυτά, η πρώτη παρατήρηση μεταξύ του βασικού μοντέλου M1 και του M2 είναι ότι πιο ρηχά μοντέλα λειτουργούν το ίδιο ή χειρότερα σύμφωνα με όλες τις μετρικές. Αυτό ήταν αναμενόμενο, αφού λιγότερες παράμετροι ισοδυναμούν με μικρότερη εκφραστικότητα. Παρόλ'αυτά, η μείωση στις παραμέτρους κι επομένως του μεγέθους του μοντέλου είναι πολύ μεγάλυτερη σε σχέση με τη μείωση στην επίδοση, που οδηγεί στο συμπέρασμα ότι η επίδοση επηρεάζεται περισσότερο από τα κανάλια και τη διαστασικότητα, παρά το βάθος των μοντέλων. Σε αυτήν την περίπτωση το πιο ρηχό μοντέλο έχει ίδιο αριθμό καναλιών με το πιο βαθύ στο σημείο συμφόρησης, εξαιτίας του αυξημένου αριθμού καναλιών ανά επίπεδο F_c, ενώ έχει περισσότερα δείγματα να επεξεργαστεί, εξαιτίας του μικρότερου αριθμού λειτουργιών υποδειγματοληψίας.

Η προσθήκη του αναδρομικού επιπέδου επιφέρει μια μεγάλη αύξηση στο χρόνο εκπαίδευσης. Σχετικά με την επίδοση, κρίνοντας από τα μοντέλα M1 και M3, το LSTM προσφέρει μια αύξηση σε όλες τις μετρικές εκτός από τον SIR των φωνητικών και τον SAR του ορχηστρικού μέρους. Το καλύτερο αποτέλεσμα στο SDR μπορεί να σχετίζεται με την καλύτερη μοντελοποίηση και επεξεργασία των ακολουθιακών δεδομένων από το LSTM. Για το M4, θέλαμε να ελέγξουμε αν το LSTM μπορεί να διαχειριστεί μακρύτερες ακολουθίες προερχόμενες από ένα πιο ρηχό μοντέλο χωρίς να χρησιμοποιεί το μεγαλύτερο πλαίσιο εισόδου. Το γεγονός ότι 3 μετρικές είναι καλύτερες από το M3 και 5 καλύτερες ή ίσες από το M1, υποδεικνύει ότι το LSTM ενδέχεται όντως να ωφελείται από μακρύτερες ακολουθίες και λιγότερο αφαιρετικά χαρακτηριστικά. Παρόλ'αυτά, υποθέτουμε ότι υπάρχει ένα σημείο καμπής, στο οποίο η περαιτέρω αύξηση του μήκους της ακολουθίας θα μειώσει την επίδοση του LSTM, αφού δεν είναι φτιαγμένο να διαχειρίζεται πολύ μεγάλες ακολουθίες και, σε συνδυασμό με ένα πιο ρηχό μοντέλο, θα χειροτερέψει τη συνολική ικανότητα διαχωρισμού του μοντέλου.

Σχετικά με τον DWT, το M7 πετυχαίνει την καλύτερη επίδοση μεταξύ όλων των εκπαιδευμένων μοντέλων σε 3 από τις μετρικές, συμπεριλαμβανομένου και του φωνητικού SDR. Την ίδια στιγμή, ξεπερνά σε επίδοση το M5 σε όλους τους τομείς, με μόνο μια μικρή αύξηση στις παραμέτρους. Αυτό συμφωνεί με τα αποτελέσματα της αρχικής εργασίας [61] και δείχνει ξεκάθαρα ότι ο διαχωρισμός των χαρακτηριστικών σε υψηλής και χαμηλής συχνότητας, συνδυαζόμενος με την αποφυγή απώλειας πληροφορίας, χάρις στη συνιστώσα υψηλής συχνότητας του DWT, τροφοδοτεί στο υπόλοιπο μοντέλο ουσιώδη πληροφορία για τη διαδικασία του διαχωρισμού πηγών. Η προσθήκη του LSTM με ταυτόχρονη μείωση του βάθους που συμβαίνει στο M8 χειροτερεύει την επίδοση σε 3 μετρικές, συμπεριλαμβανομένου του φωνητικού SDR, αλλά τη βελτιώνει στις υπόλοιπες, υποδεικνύοντας ότι και το LSTM μπορεί να επωφεληθεί από τα καλύτερα χαρακτηριστικά που προσφέρει ο DWT.

Τέλος, το M5 με το επίπεδο εξόδου με υπολογισμό διαφοράς έχει αισθητά καλύτερη επίδοση από το απλό μοντέλο M1, επισημαίνοντας ότι είναι ευεργετική η επιβολή της προσθετικής ιδιότητας. Την ίδια στιγμή όμως, το M6 εχει χειρότερη επίδοση συνολικά σε σχέση με το M5 και το M3, διαψεύδοντας το προηγούμενο συμπέρασμα. Δεν μπορούμε να το εξηγήσουμε αυτό επακριβώς, αλλά υποθέτουμε ότι είτε τα αποτελέσματα της επέκτασης δεν εμφανίζουν συνέπεια μεταξύ διαφορετικών πειραμάτων, ή ότι υπάρχει μια δυσκολία στην επεξεργασία των μουσικών οργάνων από το LSTM, καθώς το M6 εκπαιδεύεται με βάση μόνο το σφάλμα των ορχηστρικού μέρους.

Συνοψίζοντας, εάν δεν υπάρχουν περιορισμοί στη μνήμη, δεν υπάρχει λόγος να μην χρησιμοποιηθεί το μεγαλύτερο πλαίσιο εισόδου στο Wave-U-Net, ειδικά αφού η επίδραση του padding είναι τόσο εμφανής. Ο DWT είναι μια τεχνική που φαίνεται πολλά υποσχόμενη για αρχιτεκτονικές που περιλαμβάνουν ανάλυση σε πολλές κλίμακες σήματος. Ο μετασχηματισμός αυτός ταιριάζει απόλυτα στο Wave-U-Net, αφού βελτιώνει την επίδοσή του και είναι αρκετά ευέλικτος για να συνδυάζεται και με άλλες επεκτάσεις. Επομένως, πιστεύουμε ότι θα πρέπει να συμπεριλαμβάνεται σε κάθε παρόμοια εκδοχή της αρχιτεκτονικής. Σχετικά με το επίπεδο εξόδου με υπολογισμό διαφοράς. αν και ενδεχομένως είναι λειτουργικό, το γεγονός ότι βασίζεται στην ανεπιβεβαίωτη υπόθεση ότι οι πηγές έχουν αναμιχθεί προσθετικά, μαζί με το ότι τα αποτελέσματα ήταν ασυνεπή, μας κάνει να είμαστε πολύ επιφυλακτικοί για την αποδοτικότητά του. Τέλος, όσον αφορά στα αναδρομικά επίπεδα, ενδέχεται να αποτελεί ένα χρήσιμο κομμάτι της αρχιτεκτονικής Wave-U-Net, παρόλο που τα αποτελέσματα δεν έδειξαν μια ξεκάθαρη βελτίωση της επίδοσης. Σε κάθε περίπτωση, πιστεύουμε ότι ο επιπλέον χρόνος εκπαίδευσης αποτελεί ζημία κι επομένως οποιαδήποτε δυνητική βελτίωση της επίδοσης είναι αναντίστοιχη του παραπάνω κόστους.

1.3 Conv-TasNet

1.3.1 Βασική Αρχιτεκτονική

Το Conv-TasNet είναι ένα μοντέλο που επεξεργάζεται ηχητικά μείγματα στο πεδίο του χρόνου. Το μοντέλο διαχωρίζει τα πηγαία σήματα εκτιμώντας και εφαρμόζοντας μάσκες σε λανθάνουσες αναπαραστάσεις, που παράγονται από το δίκτυο. Η αρχιτεκτονική αποτελείται από 3 στάδια επεξεργασίας: έναν κωδικοποιητή, έναν διαχωριστή κι έναν αποκωδικοποιητή. Σε υψηλό επίπεδο, το μοντέλο λειτουργεί ως εξής:

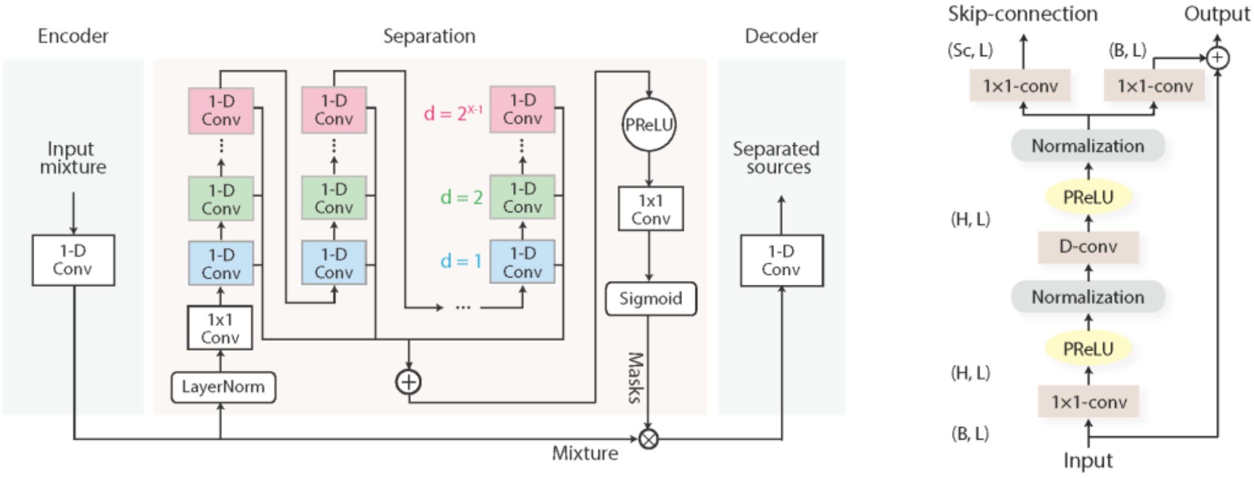
- Ο κωδικοποιητής μετατρέπει το σήμα εισόδου σε μια $N \times T$ χρονο-συχνοτική αναπαράσταση, κατάλληλη για τη διαδικασία διαχωρισμού
- Ο διαχωριστής επεξεργάζεται την είσοδο και προσπαθεί να εξάγει πληροφορία για μια συνάρτηση βάρους για κάθε πηγή. Αυτή η συνάρτηση εφαρμόζεται πολλαπλασιαστικά στην κωδικοποιημένη αναπαράσταση, αποτελώντας ουσιαστικά μια μάσκα.
- Ο αποκωδικοποιητής ανασκευάζει τις κυματομορφές των πηγών μετατρέποντας κάθε σήμα πίσω στην αρχική αναπαράσταση.

Πιο συγκεκριμένα, ο κωδικοποιητής μετατρέπει επικαλυπτόμενα, διαδοχικά τμήματα του μείγματος εισόδου σε μια λανθάνουσα αναπαράσταση μεγάλης διάστασης, εφαρμόζοντας 1Δ βηματική συνέλιξη με ένα σχετικά μεγάλο πυρήνα, μεγέθους P. Το μέγεθος του βήματος είναι το μισό του πυρήνα, δημιουργώντας 50% επικάλυψη μεταξύ διαδοχικών τμημάτων. Η συνέλιξη εφαρμόζει πολλαπλά φίλτρα, αυξάνοντας τον αριθμό των καναλιών από τον αρχικό A σε N, παράγοντας έναν πολυκαναλικό χάρτη χαρακτηριστικών. Αυτός δομικά μοιάζει με ένα φασματογράφημα, αλλά επειδή προκύπτει από εκπαίδευση, είναι πιθανό να δημιουργηθούν αναπαραστάσεις του σήματος που είναι πολύ πιο κατάλληλες για τη διαδικασία του διαχωρισμού από τον Μετασχηματισμό Φουριέ Βραχέως Χρόνου (Short-Time Fourier Transform, STFT) ή κάποιον άλλο προκαθορισμένο μετασχηματισμό. Επίσης, η πληροφορία για τη φάση του σήματος, η οποία παραλείπεται από πολλές τεχνικές που χρησιμοποιούν τον STFT, συμπεριλαμβάνεται στη λανθάνουσα αναπαράσταση,

Η έξοδος του κωδικοποιητή διέρχεται από μια ενεργοποίηση ReLU για να διασφαλιστεί ότι η αναπαράσταση είναι μη αρνητική, δημιουργώντας έτσι ουσιώδεις αναπαραστάσεις. Πριν τον διαχωριστή ο χάρτης χαρακτηριστικών κανονικοποιείται στις διαστάσεις καναλιού και χρόνου για να επιταχυνθεί η εκπαίδευση και περνάει από ένα συνελικτικό επίπεδο 1x1 το οποίο αλλάζει τον αριθμό των καναλιών από N σε B.

Ο διαχωριστής χρησιμοποιεί ένα Χρονικό Συνελικτικό Δίκτυο (TCN) [33] που αποτελείται από R υπομονάδες, συνδεδεμένες σειριακά. Κάθε υπομονάδα αποτελείται από X συνελικτικά μπλοκ με αυξανόμενο δείκτη διαστολής, $d_i = \{1, 2, \ldots, 2^{X-1}\}$. Οι πολλαπλοί δείκτες διαστολής επιτρέπουν στις υπομονάδες να ανακαλύπτουν μοτίβα σε πολλαπλές κλίμακες, επειδή τα επιμέρους μπλοκ λειτουργούν ως φίλτρα με διαφορετικό δεκτικό πεδίο. Κάθε συνελικτικό μπλοκ μετατρέπει τη λανθάνουσα αναπαράσταση κατά μήκος της διάστασης καναλιών από τη διάσταση του σημείου συμφόρησης, B σε μια κρυφή διάσταση H, προκειμένου να εκτελεστεί η διαδικασία της συνέλιξης. Πριν και μετά αυτής, οι εκάστοτε χάρτες χαρακτηριστικών κανονικοποιούνται.

Από την παραπάνω διαδικασία προκύπτει μια μάσκα, η οποία χρησιμοποιείται για να παραχθούν δύο έξοδοι. Η μάσκα ακολουθεί δύο διαφορετικά μονοπάτια, και στα δύο εκ των οποίων υπάρχει ένα επίπεδο συνέλιξης 1x1 που αλλάζει τον αριθμό των καναλιών πίσω σε B, οδηγώντας σε δύο αναπαραστάσεις.



(a) (b)

Σχήμα 1.3.1: (a) Διάγραμμα της αρχιτεκτονικής Conv-TasNet. (b) Σχεδιασμός του μονοδιάστατου συνελικτικού μπλοκ που χρησιμοποιείται στο TCN. [37]

Η πρώτη αναπαράσταση προωθείται με μια skip σύνδεση στο τέλος του διαχωριστή, ενώ η δεύτερη προστίθεται με την είσοδο του μπλοκ και προωθείται στο επόμενο μπλοκ ως είσοδος. Ουσιαστικά, με αυτόν τον τρόπο οι δείκτες διαστολής κάθε μπλοκ και κάθε υπομονάδας ενώνονται, κάνοντας όλο τον διαχωριστή να λειτουργεί σαν ένα φίλτρο με πολύ μεγάλο δεκτικό πεδίο, ικανό να ανακαλύψει εκτός από τοπικές και καθολικές εξαρτήσεις του σήματος. Οι επιμέρους μάσκες που έρχονται από τις skip συνδέσεις αθροίζονται προκειμένου να παραγάγουν τη συνολική μάσκα, η οποία περνάει με τη σειρά από μια PReLU ενεργοποίηση. ένα συνελικτικό επίπεδο 1x1 που αλλάζει τον αριθμό των καναλιών από B σε $C \cdot N$, όπου C ο αριθμός των πηγών και μια σιγμοειδή ενεργοποίηση, πριν προκύψει ο τελικός πίνακας μασκών.

Ο πίνακας μασκών που προκύπτει από τον διαχωριστή χωρίζεται σε C μάσκες, μία για κάθε πηγή, που μετέπειτα εφαρμόζονται στην κωδικοποιημένη αναπαράσταση, με αποτέλεσμα πολλαπλά σήματα στον λανθάνοντα χώρο. Τέλος, τα πηγαία σήματα ανακατασκευάζονται από τον αποκωδικοποιητή χρησιμοποιώντας βηματική, ανάστροφη, 1Δ συνελίξη, με το ίδιο βήμα και μέγεθος πυρήνα με αυτήν του κωδικοποιητή.

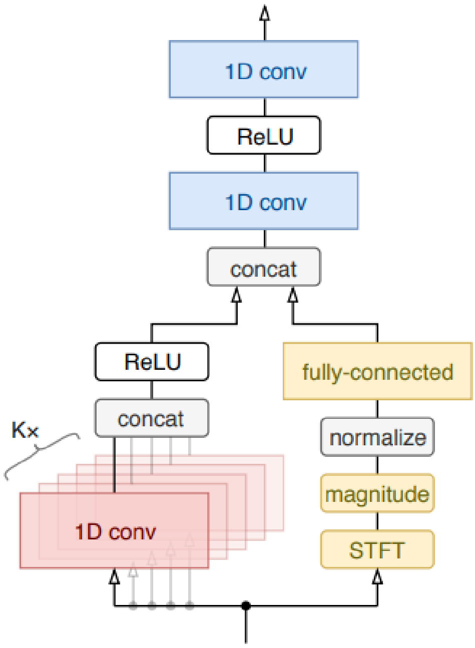
Στην εκδοχή της αρχιτεκτονικής που παρουσιάστηκε στο [14], η οποία είναι και αυτή που χρησιμοποιήθηκε στα πειράματά μας, δεν υπάρχουν skip συνδέσεις. Αντιθέτως, ο πίνακας μασκών προκύπτει από την αναπαράσταση του τελευταίου συνελικτικού μπλοκ του διαχωριστή. Επιπροσθέτως, ο αποκωδικοποιητής αντί να χρησιμοποιεί ένα μονοδιάστατο ανάστροφο συνελικτικό επίπεδο για να αλλάξει τις διαστάσεις καναλιών και χρόνου, χρησιμοποιεί έναν γραμμικό μετασχηματισμό που αλλάζει τον αριθμό των καναλιών από N σε $A \times P$ και μετά ανασκευάζει το σήμα χρησιμοποιώντας μέθοδο επικάλυψης-προσθήκης (overlap-add), που επαναφέρει την αρχική χρονική ανάλυση του σήματος.

1.3.2 Επεκτάσεις

Ισχυρότερος Κωδικοποιητής

Αυτή η επέκταση προτάθηκε στο [56] και ουσιαστικά αποτελεί έναν συνθετότερο κωδικοποιητή, ικανό να εντοπίσει πιο πολλά χαρακτηριστικά από το σήμα εισόδου. Αυτό το πετυχαίνει συνδυάζοντας δύο ομάδες χαρακτηριστικών, ερχόμενες από μια συστοιχία συνελικτικών επιπέδων και από ένα φασματογράφημα πλάτους.

Όσον αφορά στην υλοποίηση, για την πρώτη ομάδα χαρακτηριστικών, αντί να χρησιμοποιεί ένα συνελικτικό επίπεδο, ο κωδικοποιητής ενσωματώνει K τέτοια επίπεδα συνδεδεμένα παράλληλα με διαφορετικά μεγέθη πυρήνα, προκειμένου να ανακαλύψουν χαρακτηριστικά σε ένα ευρύ φάσμα συχνοτήτων, τα οποία στη συνέχεια συνενώνονται και περνάνε από μια ενεργοποίηση ReLU. Το k-οστό επίπεδο έχει πυρήνα με μέγεθος ίσο με $\frac{1}{2^k}$ του αρχικού και κανάλια ίσα με $\frac{2^k}{2K}$ του αρχικού. Για τη δεύτερη ομάδα χαρακτηριστικών, ο κωδικοποιητής παίρνει χαρακτηριστικά από ένα φασματογράφημα STFT, τα οποία κανονικοποιούνται και περνάνε από έναν γραμμικό μετασχηματισμό. Τα δύο είδη χαρακτηριστικών συνενώνονται και διέρχονται από δύο μονοδιάστατα συνελικτικά επίπεδα, χωρισμένα από μια ενεργοποίηση ReLU, τα οποία διορθώνουν τον αριθμό των καναλιών να ταιριάζει με αυτόν του αρχικού κωδικοποιητή, δηλαδή N. Επομένως, ο ισχυρότερος κωδικοποιητής μπορεί να αντικαταστήσει πλήρως τον αρχικό χωρίς καμία επιπλέον αλλαγή στο υπόλοιπο δίκτυο.



Σχήμα 1.3.2: Αρχιτεκτονική του ισχυρότερου κωδικοποιητή [56].

Ο αποκωδικοποιητής λειτουργεί αντίστοιχα, μετατρέποντας τη λανθάνουσα αναπαράσταση με ένα συνελικτικό επίπδο και μια συνάρτηση ReLU και μετά χωρίζοντας την σε K μέρη. Αυτά περνάνε από συνελικτικά επίπεδα, ίδια στον αριθμό και στα χαρακτηριστικά με αυτά του κωδικοποιητή, προτού αθροιστούν για να προκύψουν τα πηγαία σήματα εξόδου.

Κωδικοποιητής με Gammatone Συστοιχία Φίλτρων (Gammatone Filterbank)

Στην εργασία [15] μια προκαθορισμένη συστοιχία φίλτρων, η πολυφασική gammatone συστοιχία φίλτρων (MP-GTF) προτείνεται ως αντικαταστάτης του αρχικού κωδικοποιητή, για το πρόβλημα του διαχωρισμού ομιλίας.

Η MP-GTF βασίζεται στην ακουστική gammatone συστοιχία φίλτρων (A-GTF) [46], η οποία μοντελοποιεί την κίνηση της μεμβράνης του αφτιού στο ακουστικό σύστημα του ανθρώπου. Αυτή η συστοιχία φίλτρων αποτελείται από μια σειρά από φίλτρα στενής ζώνης, τοποθετημένα με μη γραμμικό τρόπο στη συχνότητα και με αυξανόμενο εύρος ζώνης. Η κρουστική απόκριση του κάθε φίλτρου ορίζεται ως

$$\gamma(t) = \alpha t^{(p-1)} e^{-2\pi b t} \cos(2\pi f_c t + \phi)$$

όπου f_c είναι η κεντρική συχνότητα, ϕ η διαφορά φάσης, α το πλάτος, $t > 0$ ο χρόνος σε δευτερόλεπτα, p η τάξη του φίλτρου και b η παράμετρος εύρους ζώνης. Οι κεντρικές συχνότητες f_c είναι τοποθετημένες σε ίσα διαστήματα στην κλίμακα ERB

$$\mathrm{ERB}_{\mathrm{scale}}(f_{Hz}) = 9.265 \log\left(1 + \frac{f_{Hz}}{24.7 \times 9.265}\right)$$

Η κλίμακα αυτή προκύπτει ολοκληρώνοντας την ποσότητα $\mathrm{ERB}(f_c)^{-1}$ ως προς τη συχνότητα, όπου ERB είναι το ισοδύναμο τετραγωνικό εύρος ζώνης (Equivalent Rectangular Bandwidth). Για ένα δοθέν φίλτρο, αυτό πρόκειται για το εύρος ζώνης των τετραγωνικών φίλτρων με το ίδιο μέγιστο κέρδος και συνολική ενέργεια, και δίνεται από τον ακόλουθο τύπο

$$\mathrm{ERB}(f_c) = 24.7 + \frac{f_c}{9.265}$$

Η MP-GTF χρησιμοποιεί φίλτρα τάξης $p = 2$, με κρουστική απόκριση που κόβεται στα 2ms και των οποίων το εύρος ζώνης υπολογίζεται ως $b = \mathrm{ERB}(f_c)/1.57$. Οι αποστάσεις μεταξύ των κεντρικών συχνοτήτων των φίλτρων είναι 1 στην κλίμακα ERB, που σημαίνει ότι η κεντρική συχνότητα του επόμενου φίλτρου καθορίζεται ως

$$f_{i+1} = \mathrm{ERB}_{\mathrm{scale}}^{-1}(\mathrm{ERB}_{\mathrm{scale}}(f_i) + 1)$$

όπου i ο δείκτης του φίλτρου.

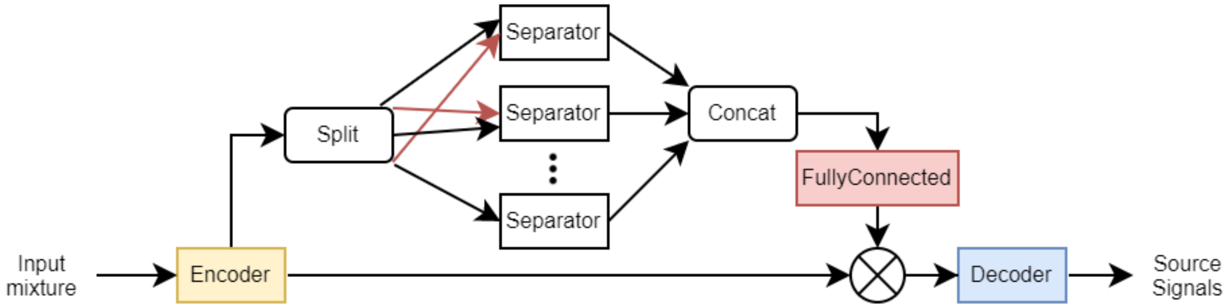
Επίσης, για να ικανοποιείται ο περιορισμός μη αρνητικότητας της εισόδου του διαχωριστή, τα φίλτρα στην συστοιχία ορίζονται σε ζεύγη, περιλαμβάνοντας για κάθε φίλτρο και το αρνητικό του. Επομένως, ο αριθμός των φίλτρων είναι

$$\#\mathrm{filters} = 2 \cdot \#\mathrm{center\_frequencies} = 2 \cdot (\lfloor \mathrm{ERB}_{\mathrm{scale}}(f_{\max}) \rfloor - \lfloor \mathrm{ERB}_{\mathrm{scale}}(f_{\min}) \rfloor)$$

όπου οι $f_{\max}$ και $f_{\min}$ αντιστοιχούν στη μεγαλύτερη και μικρότερη επιθυμητή κεντρική συχνότητα. Τυπικά αυτές οι τιμές είναι $f_max = f_{\mathrm{Nyquist}}/2$ και $f_min = 100Hz$. Καθώς αυτός ο αριθμός για τα φίλτρα είναι λίγο περιορισμένος (48 φίλτρα για ένα διάστημα συχνοτήτων 100-4000Hz), μπορούν να τοποθετηθούν επιπλέον φίλτρα με την ίδια κεντρική συχνότητα και διαφορά φάσης στο διάστημα $[0, \pi)$, αφού οι φάσεις από $[\pi, 2\pi)$ επιλέγονται αυτόματα στα αρνητικά φίλτρα, ή να αρθεί ο περιορισμός απόστασης 1 στην κλίμακα ERB μεταξύ των φίλτρων.

Πολυζωνικός Διαχωρισμός

Ο πολυζωνικός διαχωρισμός για το Conv-TasNet είναι εμπνευσμένος από το MMDenseLSTM [63], ένα μοντέλο που πέτυχε πολύ καλά αποτελέσματα στο πρόβλημα του διαχωρισμού φωνής. Το μοντέλο αυτό λειτουργεί στο χρονοσυχνοτικό πεδίο, χωρίζοντας το φασματογράφημα σε πολλαπλές συχνοτικές ζώνες και επεξεργαζόμενο την κάθε ζώνη ξεχωριστά. Προκειμένου να δημιουργηθεί μια παρόμοια αρχιτεκτονική στο Conv-TasNet αντιμετωπίζουμε την αναπαράσταση που παράγει ο κωδικοποιητής σαν ένα λανθάνον φασματογράφημα, αφού πρόκειται για μια δισδιάστατη αναπαράσταση, αποτελούμενη από πολλά κανάλια μονοδιάστατων χαρτών χαρακτηριστικών. Αυτήν, λοιπόν, η αναπαράσταση χωρίζεται κατά μήκος της διάστασης καναλιών προκειμένου να δημιουργηθούν Q ζώνες $w_i \in \mathbb{R}^{B_i \times L'}, i = 1 \dots Q$, όπου B_i ο αριθμός των καναλιών που ανατίθενται σε κάθε ζώνη. Σημειώνεται ότι κάθε κανάλι δεν αντιστοιχίζεται κατ'αποκλειστικότητα σε μια ζώνη, αλλά αντιθέτως μπορεί να συμμετέχει σε πολλές ζώνες. Αυτές οι ζώνες προωθούνται σε Q διαχωριστές, ο καθένας εκ των οποίων παράγει τον αντίστοιχο πίνακα μασκών. Σκοπός είναι οι διαχωριστές να εξειδικευτούν στα συγκεκριμένα κανάλια που δέχονται και να βελτιώσουν την ικανότητα διαχωρισμού κι άρα τη συνολική επίδοση του μοντέλου. Στη συνέχεια, οι πίνακες συνενώνονται κατά μήκος της διάστασης καναλιών προκειμένου να αποκατασταθεί η διάσταση και μετά επεξεργάζονται από ένα πλήρως συνδεδεμένο (fully-connected) επίπεδο. Το υπόλοιπο δίκτυο μένει απαράλλακτο. Στην περίπτωση που κάθε κανάλι ανατίθεται αποκλειστικά σε μια ζώνη αυτό το τελευταίο επίπεδο παραλείπεται.



Σχήμα 1.3.3: Μπλοκ διάγραμμα του πολυζωνικού διαχωρισμού. Τα κόκκινα στοιχεία χρησιμοποιούνται μόνο όταν η κατανομή καναλιών σε ζώνες δεν είναι 1-1. Τα κόκκινα βέλη δείχνουν ότι ένα κανάλι μπορεί να δοθεί ως είσοδος σε 2 διαχωριστές.

1.3.3 Πειραματική Διάταξη

Για τα πειράματα χρησιμοποιήθηκε το σύνολο δεδομένων MUSDB18 [50] με τις ίδιες ιδιότητες με πριν, αλλά με 86-14 διαχωρισμό σε δεδομένα εκπαίδευσης και επαλήθευσης. Ως επαύξηση δεδομένων χρησιμοποιήθηκε η διαδικασία από το [67]. Συγκεκριμένα, τα τραγούδια, με μια πιθανότητα, υφίστανται μεταβολή φάσης κατά π και αλλαγή των δύο καναλιών. Επίσης, χρησιμοποιήθηκε χρονική μετατόπιση έως 2 δευτερολέπτων για κάθε συνιστώσα του μείγματος ξεχωριστά και εκ νέου σύνθεση του μείγματος με συνιστώσες από διαφορετικές παρτίδες (batches). Όλοι αυτοί οι μετασχηματισμοί γίνονται κατά τη διάρκεια της εκπαίδευσης, με αποτέλεσμα ένα ξεχωριστό σύνολο εκπαίδευσης κάθε εποχή, το οποίο βελτιώνει τη διαδικασία της εκπαίδευσης, αποφεύγοντας φαινόμενα υπερεκμάθησης (overfitting).

Πειράματα με την Πολυζωνική Αρχιτεκτονική

Σε αυτό το σετ πειραμάτων εκπαιδεύτηκαν 5 μοντέλα (B1-B5) που περιλαμβάνουν τον πολυζωνικό διαχωρισμό για 150 εποχές. Οι ιδιότητες των μοντέλων περιγράφονται στον Πίνακα 1.3, τα αποτελέσματα φαίνονται στον Πίνακα 1.4, ενώ το Σχήμα 1.3.4 παρουσιάζει την απόκριση συχνότητας των φίλτρων του κωδικοποιητή για κάθε μοντέλο. Τα φίλτρα του κωδικοποιητή είναι χωρισμένα με βάση τον διαχωριστή στον οποίο κατανέμονται.

Model	#params	#bands (Q)	Bottleneck Coefficient	Full Band
B1	6.55M	1	×1	×
B2	6.47M	2	×1	×
B3	12.87M	2	×2	×
B4	6.51M	4	×1	×
B5	6.61M	3	×1	✓

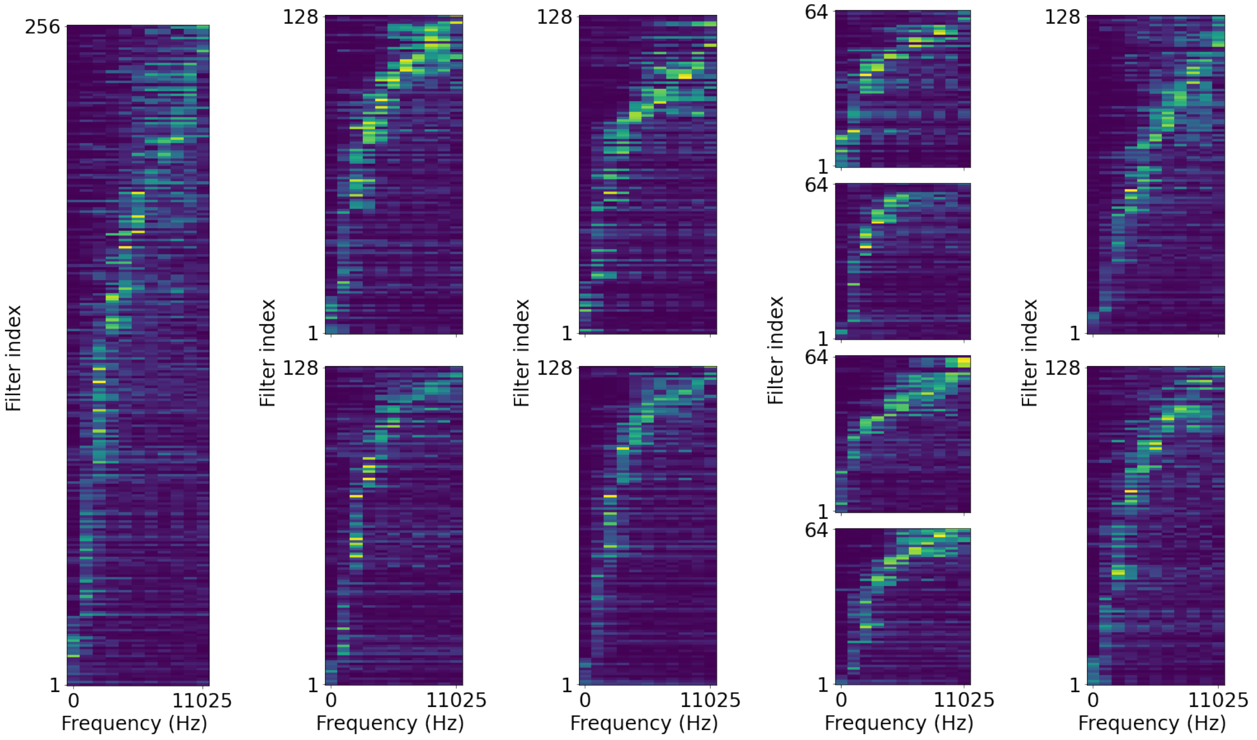
Πίνακας 1.3: Αριθμός παραμέτρων και αρχιτεκτονικές λεπτομέρειες για τα B μοντέλα. Ο συντελεστής σημείου συμφόρησης χρησιμοποιείται για να αλλάξει το μέγεθος του σημείου συμφόρησης κάθε διαχωριστή, που κανονικά ταιριάζει με τον αριθμό των καναλιών της ζώνης που εισέρχεται σ'αυτόν.

		B1 [14]	B2	B3	B4	B5
Voc.	SDR	5.81	**6.37**	6.11	6.05	5.94
	SIR	14.13	14.25	**14.69**	14.61	14.23
	SAR	6.59	**7.12**	6.59	6.98	6.78
Acc.	SDR	11.78	12.21	**12.47**	11.66	11.76
	SIR	16.01	16.69	**16.76**	16.04	16.01
	SAR	14.24	**14.52**	14.25	14.10	14.37

Πίνακας 1.4: Αποτελέσματα των πολυζωνικών πειραμάτων. Οι έντονες τιμές δείχνουν την καλύτερη επίδοση. Οι υπογραμμισμένες τιμές δείχνουν μια στατιστικά σημαντική βελτίωση (p=0.01) έναντι του βασικού μοντέλου B1.

Τα δύο διζωνικά μοντέλα B2 και B3 πετυχαίνουν την καλύτερη επίδοση μεταξύ όλων των μοντέλων, με το B2 να είναι λίγο καλύτερο στον φωνητικό SDR και το B3 στον ορχηστρικό SDR. Οι αποκρίσεις συχνότητας των φίλτρων των διαχωριστών μοιάζουν πάρα πολύ κι επομένως συμπεραίνεται ότι ο στόχος για δημιουργία εξειδικευμένων διαχωριστών, με βάση τη συχνότητα, επετεύχθη, αφού ο κωδικοποιητής επεξεργάζεται τις εισόδους του κάθε διαχωριστή με μια ομάδα φίλτρων με διαφορετικά χαρακτηριστικά. Πιο συγκεκριμένα, στα Σχήματα 1.3.4b)-c), ο ένας διαχωριστής λαμβάνει περισσότερα φίλτρα με υψηλότερη κεντρική συχνότητα, ενώ ο άλλο περισσότερα με χαμηλή κεντρική συχνότητα, μικρότερο εύρος ζώνης και μικρότερη ενέργεια. Βέβαια, σημειώνεται ότι το B3 έχει διπλάσιες παραμέτρους από το B2, επομένως η αυξημένη εκφραστικότητα εξαιτίας της διπλάσιας διάστασης B δεν οδηγεί σε καλύτερο διαχωρισμό.

Το μοντέλο με τις 4 ζώνες, B4, ξεπερνά σε επίδοση το βασικό, B1, αλλά όχι τα μοντέλα με τις 2 ζώνες B2 και B3. Αυτό υπονοεί ότι η χρήση περισσότερων διαχωριστών προκειμένου αυτοί να διαχειριστούν στενότερες συχνοτικές ζώνες δεν κλιμακώνει καλά. Η χαμηλή επίδοση θα μπορούσε να αποδοθεί και στον μικρότερο αριθμό καναλιών που αποδίδονται σε κάθε διαχωριστή, καθώς και στο μικρότερο μέγεθος της διάστασης B, που χρησιμοποιήθηκε για να κρατηθεί ο αριθμός των παραμέτρων σταθερός.



(a) B1 (b) B2 (c) B3 (d) B4 (e) B5

Σχήμα 1.3.4: Αναπαράσταση των φίλτρων του κωδικοποιητή στο πεδίο της συχνότητας για τα πολυζωνικά μοντέλα, σύμφωνα με τα φίλτρα που δέχεται κάθε διαχωριστής. Τα φίλτρα κάθε ζώνης έχουν ταξινομηθεί με αύξουσα σειρά ως προς την κεντρική τους συχνότητα..

Ουσιαστικά, θεωρούμε ότι καθώς η πληροφορία που είναι διαθέσιμη για κάθε διαχωριστή μειώνεται, αυτοί δυσκολεύονται να συγκλίνουν ανεξάρτητα σε καλές λύσεις που βασίζονται στην αποκλειστική πληροφορία που διαθέτουν, αφού δεν έχουν επίγνωση του τι μαθαίνουν οι άλλοι διαχωριστές. Έτσι, καταλήγουν να εξάγουν επικαλυπτόμενες πληροφορίες, πιθανώς περιορίζοντας τη συνολική ικανότητα διαχωρισμού. Αυτό φαίνεται και από τις αποκρίσεις συχνότητας των φίλτρων του κωδικοποιητή, αφού, παρόλο που οι 2 μεσαίες ζώνες δέχονται φίλτρα με ξεχωριστές ιδιότητες, οι 2 ακραίες ζώνες δέχονται φίλτρα που καλύπτουν περίπου τις ίδιες συχνότητες.

Το μοντέλο που επιπλέον επεξεργάζεται όλη τη λανθάνουσα αναπαράσταση, B5, κατέγραψε αισθητά μικρότερα αποτελέσματα από τα 3 προηγηθούμενα μοντέλα, σε όλες τις μετρικές εκτός τον SAR. Αυτό ίσως οφείλεται στο ότι η επεξεργασία όλης της αναπαράστασης, πέρα από την ξεχωριστή επεξεργασία των ζωνών, ακυρώνει την ικανότητα εξειδίκευσης των διαχωριστών. Αυτό φαίνεται και από το Σχήμα 1.3.4-e), όπου, αν και υπάρχουν κάποιες διαφορές μεταξύ των φίλτρων που ανατίθενται σε κάθε διαχωριστή, το φαινόμενο δεν είναι τόσο εμφανές όσο στις περιπτώσεις που δεν υπήρχε η επεξεργασία όλης της αναπαράστασης, δηλαδή στα μοντέλα B2 και B3.

Πειράματα με Πολυζωνικού Διαχωρισμού με Διαφορετικούς Κωδικοποιητές

Προκειμένου να ελεγχθεί η ευελιξία της τεχνικής του πολυζωνικού διαχωρισμού, συνδυάζεται με τους δύο διαφορετικούς κωδικοποιητές που αναφέρθηκαν παραπάνω. Ο Πίνακας 1.5 δείχνει την περιγραφή των εκπαιδευμένων μοντέλων και ο Πίνακας 1.6 τα αποτελέσματα. Τα C μοντέλα χρησιμοποιούν τον ισχυρότερη κωδικοποιητή και εκπαιδεύτηκαν για 150 εποχές, ενώ τα D χρησιμοποιούν τον κωδικοποιητή με συστοιχία φίλτρων gammatone και εκπαιδεύτηκαν για 250 εποχές.

Όσον αφορά στα C μοντέλα, παρατηρούμε ότι τα αποτελέσματα του C2 είναι κοντά σε αυτά του βασικού, C1, ενώ το C3 έφερε χειρότερα αποτελέσματα. Απ'αυτό εξάγεται το συμπέρασμα ότι τα χαρακτηριστικά που παρέχονται από τον ισχυρότερο κωδικοποιητή δεν είναι συμβατά με την πολυζωνική τεχνική. Αυτό μπορεί να οφείλεται στην ετερογένεια των χαρακτηριστικών, δηλαδή αφ'ενός στο ότι προέρχονται από 2 διαφορετικά μονοπάτια επεξεργασίας με εντελώς διαφορετικά χαρακτηριστικά και αφ'ετέρου στο ότι τα δύο μονοπάτια δεν δημιουργούν μια ενιαία συστοιχία φίλτρων, όπως στον αρχικό ή στον MP-GTF κωδικοποιητή.

Η διαφορά στην επίδοση των μοντέλων C2 και C3 αποδίδεται στο ότι η επιτυχία του ισχυρότερου κωδικοποιητή βασίζεται στο συνδυασμό των χρονικών με τα χρονοσυχνοτικά χαρακτηριστικά. Έτσι, ο χωρισμός των δύο μονοπατιών χαρακτηριστικών για την επεξεργασία τους από διαφορετικούς διαχωριστές, στο μοντέλο C3, περιορίζει την επίδραση και το όφελος του κωδικοποιητή, σε αντίθεση με το C2 που τα χαρακτηριστικά συνενώνονται και υφίστανται επεξεργασία όλα μαζί, πριν χωριστούν σε ζώνες για τους διαφορετικούς διαχωριστές.

Όσον αφορά στα D μοντέλα, η κύρια παρατήρηση είναι ότι ο ορχηστρικός SDR είναι σημαντικά χαμηλότερος από κάθε άλλο μοντέλο. Αυτό πιθανότατα σχετίζεται με το ότι η συστοιχία φίλτρων gammatone είναι σχεδιασμένη να μοντελοποιεί την ανθρώπινη ομιλία, αντί των ηχών των μουσικών οργάνων. Κατά τ'άλλα, το μοντέλο D4 είναι το καλύτερο μεταξύ των D μοντέλων, με 5 από τις 6 μετρικές να είναι αρκετά καλύτερες. Το D3 έχει τον μεγαλύτερο ορχηστρικό SIR, αλλά εν γένει μέτρια επίδοση, ενώ το D2 έχει σχετικά μέτρια προς κακή επίδοση.

Model	Description	#params
C1	Stronger Encoder Baseline	7.28M
C2	Stronger Encoder + Multi-Band, $Q = 2$	7.22M
C3	Stronger Encoder + Multi-Band, $Q = 2$ + Split Feature Paths	7.07M
D1	MP-GTF Baseline	6.52M
D2	MP-GTF + Multi-Band, $Q = 2$	6.44M
D3	MP-GTF + Multi-Band, $Q = 2$ + Channel distribution based on phase	6.44M
D4	MP-GTF + Multi-Band, $Q = 2$ + Channel distribution based on linear layer	6.44M

Πίνακας 1.5: Αριθμός παραμέτρων και αρχιτεκτονικές λεπτομέρειες για τα C και D μοντέλα.

Η βασική διαφορά των τριών μοντέλων είναι η κατανομή των συχνοτικών διαστημάτων που η συστοιχία καλύπτει σε ζώνες για τους διαχωριστές. Πιο συγκεκριμένα, στο μοντέλο D3 οι διαχωριστές δέχονται σήματα που περιέχουν όλο το συχνοτικό περιεχόμενο του αρχικού σήματος, απλά στην κάθε περίπτωση τα σήματα έχουν υποστεί επεξεργασία με φίλτρα θετικής ή αρνητικής φάσης. Κρίνοντας εκ των αποτελεσμάτων, αυτό δεν φαίνεται να δημιουργεί κάποια εξειδίκευση στους διαχωριστές, γεγονός που οδηγεί στη σκέψη ότι προκειμένου να υπάρξει εξειδίκευση θα πρέπει να υπάρχει μια περιορισμένη επιλογή φάσματος σε κάθε ζώνη. Στην εντελώς αντίθετη πλευρά, το μοντέλο D2 κατανέμει τα κανάλια με βάση καθαρά τη συχνότητα των φίλτρων, χωρίζοντας την αναπαράσταση σε χαμηλή και ψηλή συχνότητα. Αυτό επίσης φαίνεται να αποτελεί τροχοπέδη για την διαχωριστική ικανότητα του δικτύου, πιο έντονη από την περίπτωση του D3, υπονοώντας ότι παρόλο που οι διαχωριστές χρειάζον-

	C1 [56]	C2	C3	D1 [55]	D2	D3	D4
SDR	**6.39**	6.36	6.24	5.55	5.31	5.49	**5.69**
Voc. SIR	14.39	**14.92**	13.84	14.96	14.92	14.63	**15.06**
SAR	6.82	7.09	**7.25**	7.28	7.09	7.23	**7.39**
SDR	**12.23**	12.03	11.78	8.06	8.03	7.99	**8.09**
Acc. SIR	**17.57**	17.51	17.08	18.40	18.14	**18.56**	17.77
SAR	14.20	14.07	**14.25**	14.65	14.57	14.64	**14.94**

Πίνακας 1.6: Αποτελέσματα των πειραμάτων με διαφορετικούς κωδικοποιητές. Οι έντονες τιμές υποδεικνύουν τις καλύτερες επιδόσεις σε κάθε ομάδα μοντέλων. Οι υπογραμμισμένες τιμές δείχνουν σαττιστικά σημαντική βελτίωση (p=0.01) ως προς τα βασικά μοντέλα της κάθε ομάδας μοντέλων (C1 για τα C μοντέλα, D1 για τα D μοντέλα).

ται ένα περιορισμένο εύρος ζώνης προκειμένου να εξειδικευτούν, ο περιορισμός αυτός δεν πρέπει να αποκλείει κάποια συχνοτικά διαστήματα, καθώς μπορεί να είναι χρήσιμα για τη διαδικασία διαχωρισμού άλλων διαστημάτων. Τέλος, το D4 με το ενδιάμεσο γραμμικό επίπεδο φαίνεται ότι επωφελείται από τα προτερήματα των δύο άλλων μοντέλων, αφού λόγω του γραμμικού μετασχηματισμού, κάθε διαχωριστής έχει τη δυνατότητα να επιλέγει και να συνδυάζει τις συχνότητες κατά το δοκούν, πετυχαίνοντας έτσι πολύ καλά αποτελέσματα.

Όσον αφορά και στις δύο ομάδες πειραμάτων, συγκρίνοντας τα «καλύτερα» μοντέλα από κάθε ομάδα, τα μοντέλα C1 και B2 είχαν την καλύτερη επίδοση στον SDR και για τις δύο πηγές, ξεπερνώντας το αρχικό μοντέλο B1, σε αντίθεση με το D4 που έφερε χειρότερα αποτελέσματα σε αυτή τη μετρική, ειδικά για το ορχηστρικό κομμάτι. Παρόλ'αυτά, όσον αφορά τις υπόλοιπες μετρικές, το D4 πετυχαίνει από λίγο έως πολύ καλύτερες τιμές από όλα τα μοντέλα.

Συνοψίζοντας, η πολυζωνική τεχνική αποδείχθηκε ευεργετική για τη βασική αρχιτεκτονική, οπότε πιστεύουμε ότι πρέπει να περιλαμβάνεται σε εκδοχές του Conv-TasNet. Σχετικά με τους δύο κωδικοποιητές, ο ισχυρότερος κωδικοποιητής προσέφερε μια μεγάλη βελτίωση στην επίδοση όλων των μετρικών και εφόσον ξεπέρασε και τον κωδικοποιητή με την MP-GTF στην πιο σημαντική μετρική, τον SDR, είναι πολύ πιθανό να μπορεί να συμβάλει στη δημιουργία ενός Conv-TasNet μοντέλου με κορυφαίες επιδόσεις. Ακόμα κι έτσι, πιστεύουμε ότι ο MP-GTF κωδικοποιητής μπορεί να είναι μια ελκυστική επιλογή, χάρις στις πολύ υψηλές επιδόσεις στις μετρικές εκτός του SDR, στο ότι έχει προκαθορισμένη δομή και δεν επηρεάζεται από την εκπαίδευση του μοντέλου καθώς και στο ότι φαίνεται να είναι αρκετά ευέλικτος και να μπορεί να συνδυαστεί με άλλες τεχνικές, όπως με τον πολυζωνικό διαχωρισμό.

1.4 Σύνοψη και Μελλοντικές Επεκτάσεις

Σε αυτήν την εργασία πραγματοποιήθηκε μια λεπτομερειακή έρευνα στο πρόβλημα του Διαχωρισμού Φωνητικών και έγιναν πειράματα με δύο δημοφιλείς και επιτυχημένες αρχιτεκτονικές Βαθέων Νευρωνικών Δικτύων, το Wave-U-Net και το Conv-TasNet.

Αναφορικά στην πρώτη αρχιτεκτονική, εξετάστηκε το βασικό μοντέλο και ελέγχθηκε η επίδραση διάφορων αλλαγών και επεκτάσεων που είχαν προταθεί στην αρχική και σε άλλες εργασίες, στην επίδοσή του. Πιο συγκεκριμένα, μεταξύ των αλλαγών που προτάθηκαν στο [61], πραγματοποιήθηκαν πειράματα με το μεγαλύτερο πλαίσιο εισόδου για τα συνελικτικά επίπεδα, μια τεχνική που χρησιμοποιείται για να ελαχιστοποιηθούν οι αρνητικές συνέπειες της χρήσης padding στην αρχιτεκτονική, και το επίπεδο εξόδου με υπολογισμό διαφοράς, το οποίο επιβάλει έναν περιορισμό στο επίπεδο εξόδου για να απλοποιήσει την εκπαίδευση. Σχετικά με επεκτάσεις από άλλες έρευνες [29, 56], ενσωματώθηκε η χρήση αναδρομικού δικτύου στο σημείο συμφόρησης, για να ελεγχθεί κατά πόσο θα μπορούσε να βοηθήσει στην επεξεργασία ακολουθιακών δεδομένων, και αντικαταστάθηκαν τα υπάρχοντα μπλοκ επεξεργασίας με αντίστοιχα που εφαρμόζουν DWT στους χάρτες χαρακτηριστικών, με σκοπό να αντιμετωπιστεί το πρόβλημα της επικάλυψης και της απώλειας πληροφορίας. Τα αποτελέσματα έδειξαν ότι η χρήση μεγαλύτερου πλαισίου εισόδου, σε συνδυασμό με τα μπλοκ επεξεργασίας με DWT βελτιώνουν λίγο, αλλά αισθητά την επίδοση του αρχικού μοντέλου. Αντιθέτως, για τις άλλες δύο αλλαγές τα αποτελέσματα δεν είναι αρκετά ξεκάθαρα κι επομένως δεν μπορεί να εξαχθεί κάποιο συμπέρασμα.

Αναφορικά στη δεύτερη αρχιτεκτονική, το Conv-TasNet, έγιναν τρεις ομάδες πειραμάτων. Στην πρώτη ομάδα, προτάθηκαν μερικές πρωτότυπες αλλαγές στο αρχικό μοντέλο, για να βρεθεί μια αποδοτική ώστε να αναλυθεί εις βάθος. Πιο συγκεκριμένα, προτάθηκαν δύο αρχιτεκτονικές που χρησιμοποιούν πολλαπλά Conv-TasNet, συνδεδεμένα παράλληλα και εν σειρά, μια αρχιτεκτονική που άλλαζε τον διαχωριστή, χωρίζοντάς τον σε πολλούς διαχωριστές με βάση τους δείκτες διαστολής και σε μια αρχιτεκτονική που χρησιμοποιεί πολλαπλούς διαχωριστές που επεξεργάζονται κομμάτια της λανθάνουσας αναπαράστασης ξεχωριστά. Όλες οι αλλαγές εκτός της τελευταίας απέδωσαν το ίδιο ή χειρότερα από το αρχικό μοντέλο, που σημαίνει ότι είτε είναι μη λειτουργικές ή πρέπει να ενσωματωθούν με διαφορετικό τρόπο.

Η τελευταία αλλαγή, ο πολυζωνικός διαχωρισμός, έφερε μέτρια έως καλά αποτελέσματα κι επομένως, στη δεύτερη ομάδα πειραμάτων εκτελέστηκε περαιτέρω ανάλυση. Τα πειράματα περιλάμβαναν αλλαγή των αριθμό των ζωνών και αλλαγή στις υπερπαραμέτρους, για να ελεχθεί η αποτελεσματικότητα και η κλιμακοσιμότητα της τεχνικής Τα αποτελέσματα έδειξαν ότι, ανάλογα με τις ρυθμίσεις υπερπαραμέτρων, επιτυγχάνεται αξιοσημείωτη εξειδίκευση σε κάθε διαχωριστή, όπως ήταν και ο αρχικός στόχος, καθιστώντας την τεχνική επιτυχημένη.

Στο τελευταίο μέρος των πειραμάτων συνδυάστηκε ο πολυζωνικός διαχωρισμός με δύο διαφορετικούς κωδικοποιητές, έναν ισχυρότερο κωδικοποιητή από το [56] κι έναν προκαθορισμένο κωδικοποιητή gammatone από το [46], προκειμένου να ελεχθεί κατά πόσο η τεχνική μπορεί να γενικευτεί με άλλες εκδοχές της βασικής αρχιτεκτονικής. Τα αποτελέσματα ήταν συγκρουόμενα, καθώς η επίδραση της αλλαγής ήταν ευεργετική για τον δεύτερο κωδικοποιητή και ασήμαντη για τον πρώτο. Η εξήγηση που δόθηκε έγκειται στο ότι η δομή του πρώτου κωδικοποιητή, σε αντίθεση με αυτή του δεύτερου, διαφέρει αρκετά από τη δομή του αρχικού, σε σχέση με τη φύση των χρησιμοποιούμενων χαρακτηριστικών, εφόσον σε αυτήν την περίπτωση προέρχονται τόσο από το χρονικό όσο και το χρονοσυχνοτικό πεδίο.

Σχετικά με τον πολυζωνικό διαχωρισμό, μια μελλοντική έρευνα θα μπορούσε να εστιάσει στη φύση της τεχνικής, ερευνώντας τις ιδιότητες του λανθάνοντα χώρου που δημιουργείται από τον κωδικοποιητή και τη σχέση του με τις διαχωρισμένες ζώνες. Αυτό θα βοηθήσει στην καλύτερη κατανόηση της επίδρασης της τεχνικής και θα μπορούσε να οδηγήσει σε μια καλύτερη επιλογή των υπερπαραμέτρων της λειτουργώντας ευεργετικά για τη συνολική επίδραση του μοντέλου.

Ακόμη, μια ενδιαφέρουσα κατεύθυνση έρευνας είναι η δημιουργία των ζωνών των διαχωριστών χειροκίνητα, αντί της αυτόματης εύρεσης ζωνών μέσω της εκπαίδευσης, όπως τώρα. Με αυτόν τον τρόπο. οι ζώνες θα μπορούσαν να κατασκευαστούν ώστε να έχουν συγκεκριμένες ιδιότητες, που θα μπορούσαν να διευκούνουν τη διαδικασία διαχωρισμού γενικά ή να την προσαρμόσουν στις ανάγκες μια συγκεκριμένης εφαρμογής.

Τέλος, θα μπορούσε να ερευνηθεί σε ποιο βαθμό η προτεινόμενη τεχνική μπορεί να χρησιμοποιηθεί με άλλες αρχιτεκτονικές για διαχωρισμό σημάτων που έχουν τη δομή κωδικοποιητή-διαχωριστή-αποκωδικοποιητή. Αυτό θα μπορούσε να οδηγήσει στην ενσωμάτωση της τεχνικής σε άλλες εφαρμογές ηχητικού διαχωρισμού, όπως ο διαχωρισμός μουσικών σημάτων ή ο διαχωρισμός ομιλίας.

Chapter 2

Introduction

Contents

Artificial Intelligence (AI) is an exciting field of computer science, which is receiving increasing research attention in recent years. As a whole, AI consists of the research, development and implementation of algorithms and systems that enable computation machines to think and act like humans, by receiving information, processing it and acting based on it, in order to solve a certain task. Nowadays, AI is used seemingly everywhere, with applications ranging from recommendation engines in online platforms and chatbots for customer support to speech synthesis software and self-driving cars. As many applications require the processing of sensory input, respective subfields of research have been created, like computer vision, which deals with the modelling of human vision and the understanding of visual stimuli, and computer audition, which is related to the understanding and analyzing of sound.

2.1 Problem Definition

A common problem in digital signal processing (DSP), which is the focus of much AI research, is that of source separation.

Source Separation can be defined as the process of decoupling the various source signals that make up a given signal mixture, in order to either eliminate any unwanted interferences of a single signal of interest, or to isolate the source signals for further processing. The signals can be of any nature, and hence there is a source separation task for images, audio signals or even electric signals, e.g. during electroencephalography (EEG).

Source separation overlaps partly with the signal denoising problem, because we can model the noise as a signal of the same nature as the signal of interest, generated by a different source (noise generator). Nevertheless, source separation primarily focuses on separating "proper" signals, as there are existing techniques that work extremely well on the generic denoising problem.

This thesis deals with a specific case of audio source separation, namely blind singing voice separation, in which the mixture is a music song. The goal of the problem is, given the mixture and no prior information of the sources (such as music genre, number of singers, type of instruments, microphone information etc), to separate the vocal component from the accompaniment, which is comprised of a mix of instruments. The problem is closely related to music source separation, which performs further separation of the mix to the participating, individual instrument signals, and speech separation, which tries to isolate the speech of individual speakers from a multi-speaker environment.

The problem of singing voice and music separation has many applications, starting with the most apparent one, which is the isolation of the stems of a song, that is the individual source components that make it up. These stems can then be used to create new remixes, a process that is common for DJs and in the electronic dance music scene. Also, the isolation of the instrumental part can contribute to the automatic generation of instrumental versions of songs that interest a significant part of the music audience and are essential for the karaoke industry. Finally, the separation of the participating parts can be used as an intermediary step for many other applications, such as automatic lyrics transcription, singer and music genre identification, generation of spatial effects by source manipulation, music information retrieval, sound denoising for hearing aid devices and more.

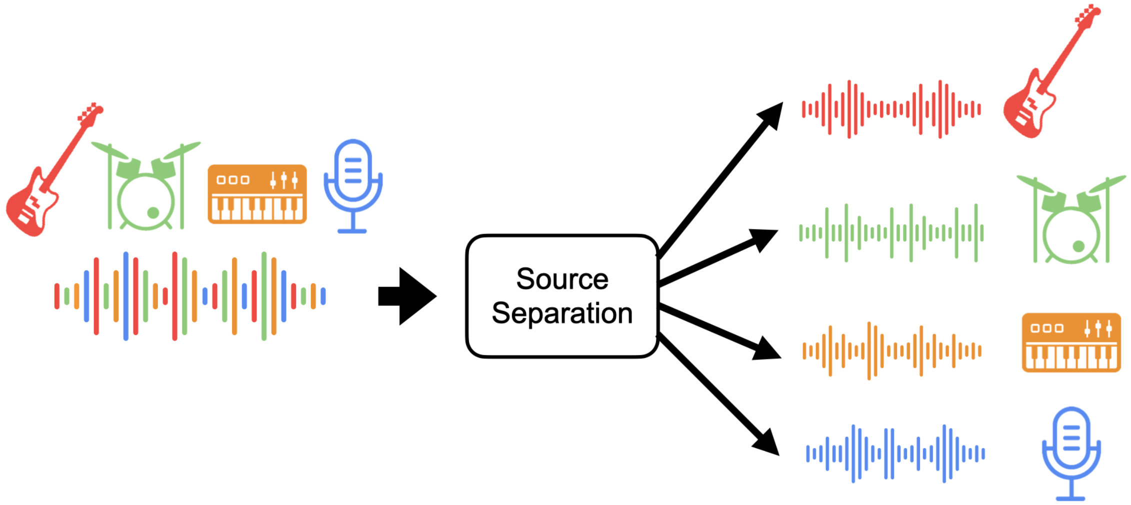


Figure 2.1.1: Overview of the music source separation problem [1].

2.2 Challenges of this Task

Singing voice separation is closely related to the *cocktail party effect*, which is the ability of the brain to focus its auditory attention on a single source, fully filtering out and thus ignoring the others [9]. Consequently, due to this "selective hearing", one can effortlessly take part in a conversation happening in noisy environments, with many distracting sounds, such as a cocktail party, hence the name of the effect. The main reasons that the brain is so successful at telling audio signals apart or distinguishing between different types of audio signals are, on the one hand, the structure of the human auditory system, which provides several cues that facilitate the sound localization and, on the other hand, the innate ability of the human brain to subconsciously model and understand many aural features. Regarding the first point, the source location can be used as a sound filtering criterion; by estimating the direction and the proximity of a sound source, one can ignore distant or ambient sounds, and amplify closer or spatially focused sounds. Regarding the second point, the human brain can distinguish a rich set of characteristics of audio signals, such as the loudness, the tone, repeated patterns etc, that can be used to filter information. Additionally, regarding speech, humans can use their communicational skills and language knowledge, like the body language, the vocabulary family etc, to further refine the distinguishability of sounds, by adapting the innate language model according to the occasion. As a result, the brain is able to make context-aware assumptions about the upcoming words in a sentence, thus making it easier to filter out non-appropriate sounds, match sounds to phonemes and even completely fill in missing words [34].

Unlike the brain, in the current problem, the computer lacks all this information, so many difficulties can be observed.

First and foremost, music has an irregular, unstructured nature. Depending on the instruments used, the singers' voices, the melody, the tempo and the combinations of them, based on the music style and the artists' preferences, there is an infinite number of audio mixtures, which differ vastly, that could legitimately be considered as "music". So, it is impossible to base the separation on rule-based techniques, although they have been proven very useful on tasks involving language [28], which is another infinite, but heavily structured, human construction.

Another difficulty lies behind the fact that either the sound mixture is monaural, meaning there is only one sound channel, or, in the multi-channel case, there is no information on the microphones used for recording. This raises a problem, because there is not enough information to localize the sound sources, cancelling the ability to apply spatial filtering by, for instance, beamforming techniques, as in [6].

Lastly, a key challenge arises from the fact that the source signals in music are highly correlated. Therefore, there does not exist a simple, reliable feature, like the signal frequency or the signal amplitude we can use as a discriminator factor, to separate the source signals. That is not the case in many signal denoising tasks, in which the noise can be modelled to have certain statistical characteristics, which enable the use of many DSP techniques, like Wiener filtering.

2.3 Goals and Contributions of this Thesis

In this thesis, a meticulous research on the existing techniques in the field of singing voice separation is carried out, focusing on the newer techniques, which employ deep neural networks and are currently the state of the art, as the provide the best results. The contributions of this research can be split into two major categories, matching two basic architectures and are the following: Regarding Wave-U-Net,

- We incorporate several existing modifications and test their synergy and their impact on the performance.

Regarding Conv-TasNet,

- We reproduce and train an implementation of Conv-TasNet, an architecture that yields state-of-the-art results in music source separation.
- We propose several extensions to the model, focusing on a multi-band extension, which splits the separator latent space into latent bands and dealing with each band individually, using multiple separators.
- We investigate the scalability of the proposed technique, using multiple numbers of bands and two different frameworks.
- We combine the proposed technique with a better encoder and a fixed filterbank to test its modularity.

2.4 Thesis Outline

The rest of the thesis is organized as follows.

- In Chapter 3, we provide the theoretical machine learning and signal processing background that is necessary for the full comprehension of the different techniques and the intuition behind them.
- In Chapter 4, we present an overview of the previous and related work in the field. We start off with some traditional techniques and then move to more recent work, that utilized deep neural networks (DNN). Also, we attempt to categorize the various DNN techniques based on

some fundamental criteria. In this chapter, we also present our database, MUSDB18, and our evaluation protocol.

- In Chapter 5, we provide an overview of the Wave-U-Net architecture and present the research and the experiments performed.
- In Chapter 6, we present the Conv-TasNet architecture and introduce our modifications towards improving its performance. We also display and discuss the results from three sets of experiments.
- In Chapter 7, we draw some conclusions regarding the contributions and results of the thesis and we discuss our thoughts on potential future extensions.

Chapter 3

Theoretical Background

Contents

3.1 Machine Learning

Machine Learning (ML) is the field of research that focuses on teaching computers to learn from data and improve with experience in order to perform certain tasks, instead of explicitly programming them to do so. Of course, it must be noted that the motivation behind the development of ML is not that humans lack the willingness to solve certain tasks. On the contrary, the constantly evolving world has raised needs and applications that involve tasks whose mathematical modeling is overly complex for humans to solve analytically. Nowadays, ML is present in a wide variety of applications, such as computer vision, image classification, sentiment analysis on texts and images, face recognition and many more.

In a high-level approach, a ML system can be seen as a single parametric function $\mathbf{Y} = f(\mathbf{X}, \theta)$, where $\mathbf{X}$ is the input matrix and θ denotes the model's parameters. The training process includes calculating the system's output when using samples from a given dataset as input, measuring its performance and adapting its parameters in order to improve it. From this high-level rundown of the way that ML systems work, we can see that each ML approach has four fundamental parts:

- The model itself, which is a parametric system containing a number of learnable parameters.
- The dataset, which is used for training and evaluation. In order to further check the generalization ability of the model, the dataset is actually split in two disjoint sets, the train and test set. This might be the most important part of an ML system, as it heavily influences the performance of the model. A poor dataset that has either too few or not general enough samples, can render a model useless for real world applications, where generalizability is of top importance.
- A "performance" index, which is called "cost function" and numerically interprets how well or badly the system perfomed its task. This cost function might be easily constructible or even trivial for some problems (e.g. accuracy on a classification task), but may be extremely difficult to find in other problems, due to the high level of abstractness of the output (e.g. quality of generated human speech)
- An optimization algorithm that tunes the model's parameters based on the output of the cost function.

ML approaches can be broadly divided in three categories:

Supervised Learning

In **Supervised Learning**, the input is accompanied by the true output. Using formal notation, for a given dataset D, the samples come in input-output pairs, forming the set $D = \{(\mathbf{x_i}, \mathbf{y_i}), i = 1, \dots, N\}$, where N is the total number of samples. The task is to learn a function that maps the input $\mathbf{X}$ to the output $\mathbf{Y}$. In order to quantify how well the model fits the training data, a loss function $L : Y \times Y \to \mathbb{R}^{\geq 0}$ is used as follows: for a sample $(\mathbf{x_i}, \mathbf{y_i})$, the model predicts a value $\mathbf{\hat{y}_i} = f(\mathbf{x_i})$, and the respective loss is estimated as $L(\mathbf{\hat{y}_i}, \mathbf{y_i})$.

This approach has the advantages of making the training humanly interpretable and the system's performance easy to evaluate, by calculating several metrics/loss functions, since the correct output is known for the dataset's samples. On the other hand, supervised methods rely heavily on the used dataset, making them more susceptible to dataset mistakes and cannot extract hidden or unknown information or assume an output that is missing from the dataset.

Supervised learning tasks can be further classified into two parts, **classification** and **regression**.

In classification tasks, the goal is to label the input data in two (binary classification) or more classes. The classification can either assign to each sample one label (single-label classification), or more (multi-label classification). A common example of binary classification is email filtering in "spam" and "no spam".

Contrarily, in regression tasks, the system processes the input variables, also called "features", to predict the output variable, which takes a value within a specific range. An example of regression task is the prediction of a house's price based on a set of features, such as house area, year of construction, location etc.

One central problem in supervised learning is the bias-variance tradeoff that affects the two sources of errors that cause a supervised learning algorithm to fail to generalize on unseen data, beyond the training set.

Regarding the error, between an independent variable $\mathbf{X}$ and a dependent variable $\mathbf{Y}$, we assume that there is a function with noise $\mathbf{Y} = f(\mathbf{X}) + \epsilon$, where the noise, ϵ has zero mean and variance σ^2. As our system doesn't know the true relation f, it can only find an estimate $\widehat{f}(\mathbf{X})$, by optimizing its parameters. Therefore, we get the equation $\mathbf{Y} = \widehat{f}(\mathbf{X}) + \epsilon$. Using a learning algorithm for the model estimation, the expected mean square error can be written as:

$$\begin{aligned} Err((f(\mathbf{X}) - \widehat{f}(\mathbf{X}))^2) &= Bias(\widehat{f}(\mathbf{X}))^2 + Variance(\widehat{f}(\mathbf{X})) + \sigma^2 \text{ where} \\ Bias(\widehat{f}(\mathbf{X})) &= \mathbb{E}[\widehat{f}(\mathbf{X})] - f(\mathbf{X}) \text{ and} \\ Variance(\widehat{f}(\mathbf{X})) &= \mathbb{E}[\widehat{f}(\mathbf{X})^2] - \mathbb{E}[\widehat{f}(\mathbf{X})]^2 \end{aligned}$$

The three error terms are separated in two categories; the irreducible error ϵ which comes from the problem itself and the reducible errors, bias and variance error, which are a matter of model selection, based on the mentioned tradeoff.

Bias is the initial assumptions about the form that our model has to fit. In the context of training data, bias can be thought of as how much we ignore them in favor of our initial assumptions. A model with high bias can oversimplify our model, reduce its flexibility and prevent it from discovering the underlying relation between inputs and outputs during training. Therefore, high bias models have both high training and test set errors. On the other hand, variance is the variability of our model to training data. In other words, it shows the dependency of our model to the training data. A model with high variance can fit the training set very well to perfectly, but fails to generalise to unseen data, leading to high test set error.

An ideal model would have low bias, so as to be flexible enough to fit well the training data, learning the relevant input-output relations and low variance, so that it can avoid being too dependent to training data and generalize well. Unfortunately, as it has been implied by the word "tradeoff", these two parameters are not independent. Instead, they are the two sides of the same coin, that of model's complexity, and hence, minimising both bias and variance is not possible. On the one hand,

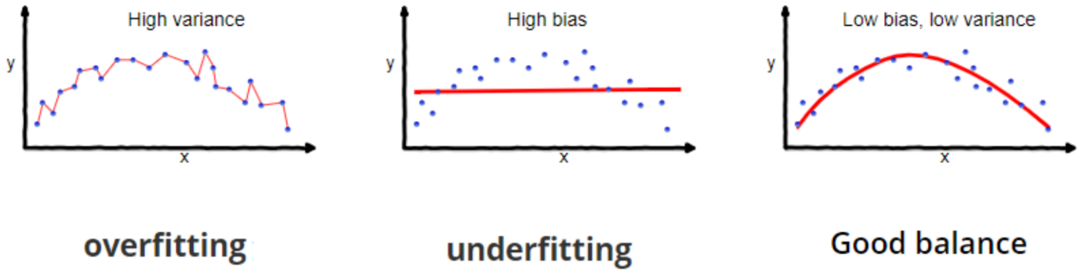


Figure 3.1.1: Depiction of the bias-variance tradeoff [2]

a complex model has many learnable parameters and high expressibility, that is it can estimate a wider range of functions. As such, it can overfit the training data, by accurately modeling both the input and the noise, which hinders its ability to generalize. On the other hand, a simple model has a few parameters, meaning that it can underfit the training data, that is to fail to model the underlying function at all.

In order to check whether the model overfits or underfits and tune its hyperparameters accordingly, it is common to remove a few samples from the training set to form the validation or dev (from development) set. In a sense, this dataset functions as a hybrid; it contains samples that are used to evaluate the model during training, but doesn't participate in neither the training of the model nor in the final testing.

Unsupervised Learning

In **Unsupervised Learning** the dataset is not accompanied by a label or a desired output value. Instead, the goal of the system is to find patterns and useful correlations in the data by its own. In contrast to supervised techniques, the unsupervised ones require simpler datasets, since they don't need any kind of data annotation, which constitutes the most time- and energy-consuming part of dataset construction. A couple of typical categories of tasks that are solved with unsupervised learning is clustering, where we want to find hidden groupings in the data, and dimensionality reduction, where we want to create smaller and denser data representations from a high-dimensional dataset.

Reinforcement Learning

Reinforcement Learning is very different from the other two methods, because it doesn't involve a fixed dataset. Instead, the system is given a set of allowable actions, rules and tools on how to act in and interact with an environment. Also, it is provided feedback on its actions in terms of reward and punishment, based on a set goal, score or potential end state(s). Then, the system is left alone in the environment to learn on its own, by trying to perform the task repeatedly, building experience in the process. This method is similar to trial and error that humans use in many tasks, mainly those containing a fine use of motor skills. An example of reinforcement learning is teaching a machine to play chess. Since providing the machine with a dataset containing all possible actions would be inefficient, the program is equipped with the set of allowable actions, the rules and the finish condition and is left to practice. In this case, the rewards could be winning a game or capturing an opponent's piece.

3.2 Neural Networks

3.2.1 Introduction

Historically, **Artificial Neural Networks** or simply **Neural Networks** were created in an attempt to mathematically model information processing of biological systems, by mimicking the structure of biological neurons. Nowadays, neural networks are a class of models used in machine learning that have proven to be very effective for multiple tasks.

3.2.2 Fully-Connected Networks

The Perceptron

A neural network, as its name suggests, is composed of multiple artificial neurons, often called nodes, connected in a variety of layouts.

The perceptron is the simplest type of neural network as it contains just a single node. It can be used to solve the task of binary classification of linearly separable classes. Although its applications are very specific, it is worth mentioning, as it is the building block of a very valuable class of neural networks, namely the multilayer perceptron, and it can help in understanding fundamental parts of machine learning.

Using mathematical notation, given a vector of inputs $\mathbf{x} \in \mathbb{R}^n$ and an internal weight vector $\mathbf{w} \in \mathbb{R}^n$, the perceptron calculates their dot product, $\mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^{n} w_i x_i$. This value is summed with an additional bias term that does not depend on the input, and then is passed through a "threshold" function h, giving the result $a = h(\mathbf{w} \cdot \mathbf{x} + b)$, which constitutes the unit's activation, and at the same time, since this is the only unit in the network, its output. The threshold function is a very simple non-linear activation function, defined as:

$$h(\mathbf{x}) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

In classification tasks, where we want the model to predict the class of the input, the output of the network isn't perceptually meaningful, unless we first assign a class to each output value in an one-one manner.

Multilayer Perceptron

Having seen how the perceptron works, we proceed to more complex networks, that can be used in a wider range of applications.

Multilayer perceptrons (MLP), as their name suggests, consist of multiple layers of nodes, More specifically, an MLP contains at least three layers: an input layer, an output layer and one or more hidden layers. The input layer receives the signal to be processed and passes it on the hidden

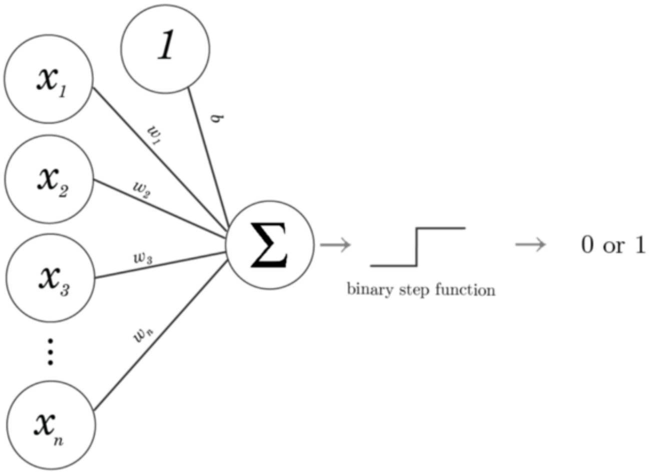


Figure 3.2.1: Graphic representation of the way a perceptron operates. The bias term is modeled as an additional input with the value of 1, weighted by a weight of b [3].

layer. No processing takes place in this layer. The hidden layers perform the main computation and processing on the input data, yielding some intermediate representations and transfer the information to the output nodes. Finally, the output layer transforms the extracted representations in a proper way depending on the task in hand. For instance, in binary classification tasks the output layer uses the extracted information to provide a single binary value, while in regression tasks the output is not necessarily bounded. Each layer, apart from the first one which is bound by the input features, can have an arbitrary amount of nodes, receiving the outputs of the previous layer as inputs and feeding their outputs to the nodes of the next layer. Each node is similar to the perceptron with the difference that its activation function needs not to exclusively be the threshold function. In fact, there is a great variety of activation functions, each one having its own advantages and drawbacks, which will be covered in a next section.

In an MLP, the mathematical notation requires some additions in order to cover the multiple nodes and layers. Let us consider a network with $K + 1$ layers, each with M_k nodes, $k = 0, \dots, K$ and an input vector $\mathbf{x} \in \mathbb{R}^D$. The input layer nodes receive one input value each and forward it without any processing or activation function application. The nodes of the first hidden layer are characterized by the following equation

$$a_j^{(1)} = \sum_{i=1}^{D} W_{ji}^{(1)} x_i + b_j^{(1)}$$

where the subscript $j = 1, \dots, M_k$ corresponds to the node index in the layer and (1) indicates the current layer (the input layer is considered to be the 0th layer). The parameters $W_{ji}^{(k)}$ are the weights and the $b_j^{(k)}$ are the biases, composing the weight matrix and bias vector of the layer, respectively. The quantities a are called activations and are given as input to the activation function to produce the layer's outputs.

$$z_j^{(1)} = h(a_j^{(1)})$$

For the following layers, the equation is slightly altered, as the inputs are the outputs of the previous layer:

$$a_j^{(k)} = \sum_{i=1}^{M_{k-1}} W_{ji}^{(k)} z_i^{(k-1)} + b_j^{(k)}$$

where M is the number of nodes of the previous layer. Finally, the activations of the last layer are transformed using an appropriate activation function to give the network's outputs

$$y_j = h(a_j^{(K)})$$

in a network with $K+1$ layers.

Training Algorithms

The process of training is fundamental in neural networks, as it is the one that enables them to improve themselves by accumulating experience. Because the network's output is affected by its parameters, the goal of training is to find and assign correct weights to the various nodes, in order to perform as well as possible.

In training we start off by calculating the performance of the network. As it has already been mentioned, we can quantify its performance by calculating a cost (loss) function that, given the true output and the predicted output of our network, calculates an error that shows how "close" the two values are. This error is dependent on the weights of the nodes, so, the process of learning can be thought of as an optimization problem.

As it is too complex to find an analytical solution to this problem, approximated solutions are offered by iterative processes. These processes are split in two parts:

- A distribution of the total error to the individual nodes. We can think that this distribution helps to see "in what way the weights of a node contribute to the total error".
- An algorithm that utilizes the above information to update the weights of each node in order to reduce the total error.

The first part is covered by the **Backward Propagation of Error** algorithm and the second by various iterative algorithms, such as gradient descent.

Backward Propagation of Errors or, shortly, backpropagation is a method for calculating the derivatives of the error with respect to each of the weights of the network. The calculation of the partial derivative is based on the chain rule of calculus. For example, suppose we have a network output $\hat{\mathbf{y}}$. Then, the cost will be $C(\hat{\mathbf{y}}, \mathbf{y})$, with $\mathbf{y}$ being the ground truth corresponding to the estimated value $\hat{\mathbf{y}}$. To find the partial derivative of the cost with respect to a weight w_{ij}, we use the chain rule as follows:

$$\frac{\partial C}{\partial w_{ij}} = \frac{\partial C}{\partial z_j} \frac{\partial z_j}{\partial a_j} \frac{\partial a_j}{\partial w_{ij}}$$

The above derivative can be calculated as the activation functions are partially differentiable.

Indeed, as we can infer from the above formula, the derivative with respect to a network weight requires the calculation of the derivative with respect to the output and the derivative of the output with regard to the activation, which in turn is dependent on all subsequent layer weights. The backpropagation algorithm takes advantage of this structure by calculating the derivatives efficiently, as its name suggests, calculating the derivatives from the last, output, layer through the first, moving backwards through the network.

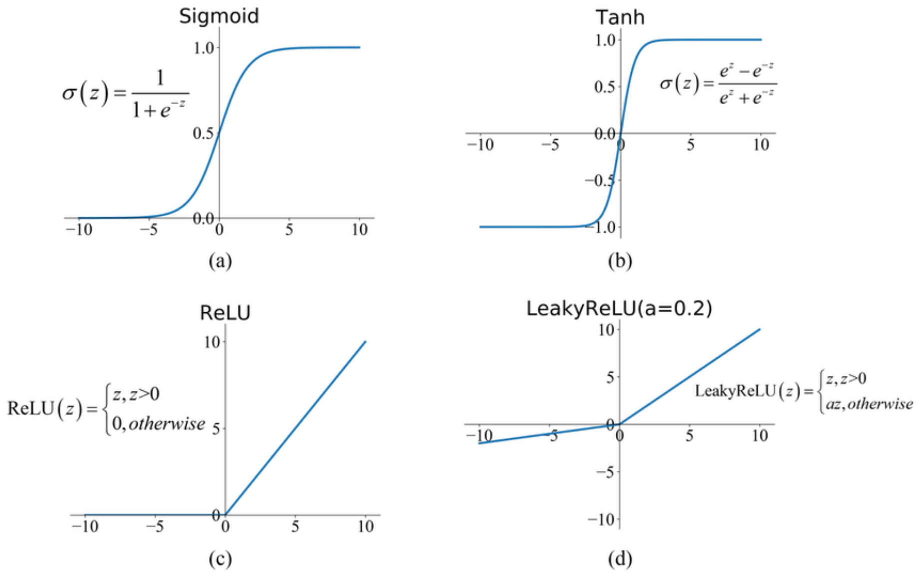
Regarding the second part, the most common algorithm of updating the weights of a neural network is gradient descent. Gradient descent is an iterative algorithm that is used to find local minima of a differentiable function. This method is based on the fact that a function $f(\mathbf{x})$ decreases the fastest if we change $\mathbf{x}$ to the direction of the negative gradient of f.

So, in the case of neural networks, we can update the network weight matrices, W, as such:

$$W = W - \gamma \cdot \nabla_W E(W)$$

where γ is the learning rate, a hyperparameter that controls how quickly or slowly we "descend" and $\nabla_W E(W)$ denotes the gradient of the error with respect to the network weights, as calculated using the backpropagation algorithm.

Typically, the above algorithm requires that we accumulate the loss from all training samples in the training set, before updating the weights. However, in practice and in the case of large datasets and deeper network structures, this is slow and close to impossible due to memory limitations. So, instead, a stochastic version of the algorithm is used, where we update the weights using a subset of the training set, which is called batch. Of course, this algorithm doesn't ensure that we always follow the fastest way downhill, but in practice it works efficiently and manages to converge quickly.



(a) (b) (c) (d)

Figure 3.2.2: Graphs of the most popular activation functions [19].

Activation Functions

Activation functions are non-linear functions that are partially differentiable and have a crucial role in neural networks. First of all, without non-linear activation functions decoupling the individual linear transformations that the layers apply to the input, the whole neural network's processing could be described by a single, linear transformation. Thus, neural networks could only deal with tasks that are solved linearly. Secondly, because most activation functions have a small, finite range of values, they act as a normalizer towards the activations, thus affecting the convergence and the convergence speed of the network.

Figure 3.2.2 shows graphs of 4 activation functions. The top two, sigmoid and tanh, have a finite range of $(0, 1)$ and $(-1, 1)$ respectively. Their characteristics are that they normalize the activations, by squashing the input values, and are continuously differentiable. The bottom two, Rectified Linear Unit (ReLU) and Leaky ReLU, have an infinite range of values, are only partially differentiable and have very simple formulas. For instance, ReLU can be computed as $\text{ReLU}(x) = \max(0, x)$. Thus they require less computation during training, eventually reducing the overall training time.

Loss Functions

A loss function has the crucial role of quantifying the performance of the model in a specific task, intuitively representing some "cost" associated with using the specific model. Thus, a loss function can be any mathematical function that can take as inputs the true and predicted output values and calculate their distance in a suitable space. Depending on the nature of the task (regression or classification task), different loss functions are used. For regression tasks, commonly used loss functions include the Mean Absolute Error (MAE) and the Mean Square Error (MSE), and for classification tasks, there is the cross-entropy loss. MAE and MSE of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ are defined as follows:

$$\text{MAE}(\mathbf{y}, \mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \mid y_i - x_i \mid$$

$$\text{MSE}(\mathbf{y}, \mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \mid y_i - x_i \mid^2$$

3.2.3 Convolutional Networks

Introduction

A convolutional neural network (CNN) is a class of deep neural networks designed for processing structured arrays of data such as images. In many Computer Vision/Computer Audition tasks, the state of the art architectures consist either entirely or partially of CNNs. What makes convolutional networks special is their ability in discovering and extracting features, by picking up patterns in the data. This property has opened up the path for end-to-end architectures, in which the data are processed in their raw form by the network, instead of being pre-processed to extract hand-crafted features.

Structure of a Convolutional Network

Typically, convolutional networks contain three types of layers: **convolutional**, **pooling** and **fully connected**.

The convolutional layer is the main computation block of a CNN. This layer, as its name suggests, performs the convolution operation between a number of kernels also known as filters, and the input data. The kernels can be thought of as "feature detectors", which, sliding through the input during the convolution, generate a feature or activation map each, resulting in a multi-channeled output, which is then passed to the next layer. The discrete convolution operation is as simple as calculating, for each output point, the dot product of the kernel and an equally-sized slice of the data, aligned with the output.

The convolutional operation is characterized by 3 hyperparameters, the tuning and the importance of which is specific to and influenced by the application: the **stride** of the convolution and the **size** and the **dilation** of the kernel.

The first parameter indicates the stride of the slide and determines how densely or sparsely the convolution is applied. For example, a stride of 1 means that the kernel is applied on every sample of the input, while a bigger stride means that the kernel skips some samples before being applied again. Strided convolutions can be useful in reducing the computation cost of the layer in cases where there is overlapping information on the data as they result in smaller feature maps.

The size and the dilation directly affect the kernel and determine the receptive field, that is the area of the input that is visible to the kernel of the filter. In more detail, with a kernel size of k and a dilation factor of n, the receptive field of the filter r is equal to $r = k \cdot n$. The size is simply the number of units that constitute the kernel. Whether these units are sequential or not is determined by the dilation, that shows how many data samples are skipped between the kernel units. For example, in the case of 1D convolution and data, a kernel of size 3 and dilation of 1 would process three consecutive samples, while a dilation of 2 would mean that in a series of 5 samples, the kernel would process the 1st, 3rd and 5th. Dilation can be thought of as searching for features at a different scale, with low dilation indicating scanning of local patterns and high dilation global patterns.

In contrary to what we know from traditional signal processing, in a neural network context, the convolutional operation takes place only for those signal values that the kernel fits entirely inside the aligned signal. As a result, the output feature map has a different (smaller) size than the input. To prevent this, padding can be used to the input by adding zeros on the edges (zero-padding) or repeating the edge values (same-padding).

Figure 3.2.3 showcases a variety of 1D convolution operations, with different settings.

Pooling layers are used to reduce the dimensionality of the feature map, by applying a downsampling operation upon it, and are usually positioned in between convolutional layers. Since CNNs are deep, containing multiple levels of layers, this allows the next convolutional layer to have a larger receptive field, that is it can discover patterns/features of larger scale, while keeping the kernel size the same. As with the convolutional layers, pooling layers have a kernel with a predefined size, which slides through the input, performing an operation. Usually, pooling layers are not trainable, meaning that the operation they perform is fixed and have no weights, although recent variants such as the auto-pool layer have appeared [39]. The two main types of pooling are max and average pooling, where the output in each step is the maximum and average, respectively, of the values of the receptive field.

Finally, fully-connected layers work in a similar fashion as the MLP; they connect each input to each output with a weight. These are typically used at the end of a deep CNN, transforming the processed feature maps in order to discover global patterns in the data or perform the required task.

Advantages of Convolutional Layers

The structure of convolutional networks provide them with some interesting and useful advantages over the standard fully-connected networks.

First of all, convolutional layers require less computation and train faster than their fully-connected counterparts. The transformations that fully-connected layers apply to the input data are implemented as matrix multiplications. Even though these operations can be computed in parallel and very quickly by modern hardware and software frameworks, in some cases where the data are of high dimensionality, such as images, matrix multiplications are extremely costly and their training is slow.

On the other hand, CNNs contain drastically fewer trainable parameters than fully-connected layers, creating less complex models that train faster, require less memory and are less prone to overfitting. More specifically, while the parameters p_f of a fully connected layer are the product of input i and output o dimensions, $p_f = i \cdot o$ the parameters of a convolutional layer p_c are affected by the number of input c_i and output c_o filters/channels and the size of the kernel k, as $p_c = c_i \cdot c_o \cdot k$.

The previous concept is known as "parameter sharing", because, at every training step, the kernel that slides through the data has parameters that are independent on the input, as opposed to the fully connected layers where each input value has a different weight depending on the node it is connected to. This contributes to the most important property of CNNs, that is equivariance to translation. This means that if we have a convolution operation g and a translation operation t, applying the translation and then the convolution is equivalent to applying the convolution first and then the translation, $g(t(\mathbf{x})) = t(g(\mathbf{x}))$. Additionally, since the weights of the kernel are independent of the input, convolutional networks are highly flexible to the input shape, as opposed to the the fully-connected layers that demand the input to have a fixed size and properties.

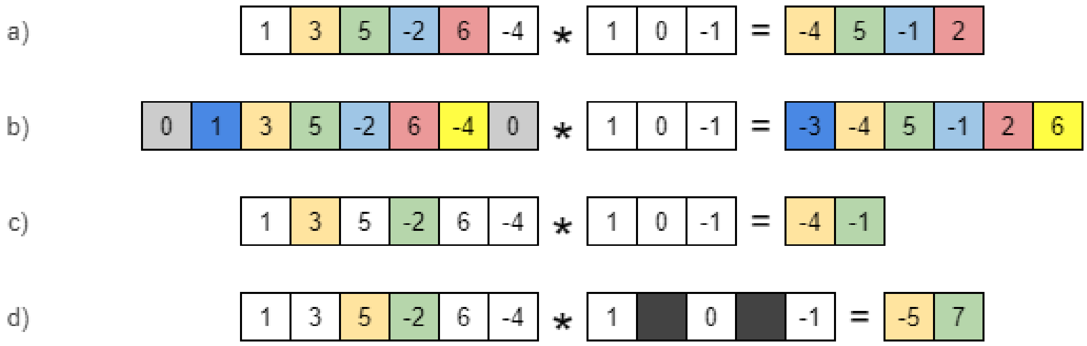


Figure 3.2.3: Plots of 1D convolutions with a kernel of size 3 and various configurations. a) Simple convolution with 1 stride and dilation. b) Convolution when zero-padding is used. Note that the output map is the same size as the input one, with the two edge elements containing possibly false information. c) Convolution with stride of two. d) Dilated convolution with a factor of 2.

Special Types of Convolutional Layers

Due to the wide use of convolutional layers in neural networks, we think that it is useful to present two common, special cases of them, the 1x1 and the transposed convolution.

1x1 Convolutional Layer: This is a normal convolutional layer with a kernel with a size of 1 (1x1 denotes the width and height of the kernel). Although this layer has the smallest possible receptive field and doesn't associate neighboring samples, it can be used to change the number of channels of a multi-channeled representation without heavy computational cost and increase in network's parameters. Additionally, since the filter is a single unit, the feature map size is left unaltered, removing the need of using padding layers.

Transposed Convolutional Layer: The basic convolutional layer typically decreases the feature map size. Transposed convolution is in a sense the opposite operation to the normal one, as it typically upsamples the feature map size. We can think transposed convolution as the operation that reconstructs an original input given the output of a convolutional layer.

As with the normal one, this operation includes a kernel, that can have a dilation factor and that slides on the input with a fixed stride. Also, it can change the number of channels of the representation by applying an arbitrary amount of filters to the input. In contrast to the normal operation, the kernel broadcasts the input elements, thereby creating an output that is larger than the input and then the overlapping elements are summed to create the output feature map. Figure 3.2.4 contains an example to better understand the operation.

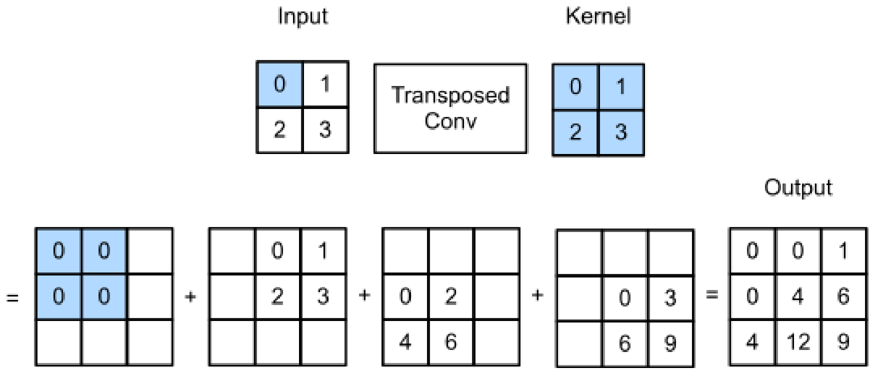


Figure 3.2.4: Example of tranposed convolution operation. Note that the output has a bigger size than the input [4].

3.2.4 Recurrent Neural Networks

A recurrent neural network (RNN) is a class of deep neural networks designed for processing of sequential data. More commonly, the sequential nature is temporal, as in the frames of a video, or the values in a time series data, but can also refer to spatial, or other kind of dependencies, depending on the way we model a given problem and task. RNNs have seen much success in tasks like language modeling [41], time series prediction [75] and speech synthesis [48].

Unlike the other neural network classes, the output of an RNN doesn't depend only on the input, but additionally on the inputs before it. Hence, the same input value, fed into an RNN after different sequences could produce a completely different output. This is made possible by the use of an internal state, that acts as the network's memory and accumulates information from prior inputs to influence the current output. Moreover, in order to handle sequences of multiple inputs, RNNs contain a feedback connection to themselves, creating a loop (thus the name), which passes information of the internal state to the start of the network. In that way, a single RNN can handle series of an arbitrary number of steps.

RNNs have some key differences from the other types of networks that offer a couple of advantages over them. Firstly, an RNN can discover and make use of sequential dependencies, solving problems that otherwise would either need extremely complex architectures or not be solvable at all. Secondly, RNNs are structured in such a way that they also benefit from the concept of parameter sharing, similar to CNNs. By using the internal state, they are able to adapt themselves to the new input, without the need of a great number of additional parameters. That's why RNN architectures typically have less parameters than their counterparts. Thirdly, as it has been already mentioned, RNNs can process sequences of any length, without increasing their nodes and their parameters.

Of course, there is a couple of shortcomings to this type of networks, which should always be taken into consideration when incorporating them in architectures. Although RNNs can handle sequences of any length, normal architectures tend to be affected significantly more by the most recent samples in the sequence, than by the older ones, essentially canceling the property of the infinite receptive field. Also, using extremely long sequences can lead to vanishing or exploding gradients, a problem that is common in very deep architectures and that prevents networks from converging fast enough, or entirely, respectively. Both of these cases are caused by the accumulation of very small or very large values during the backpropagation algorithm, which can ruin the update of the weights in the gradient descent algorithm. There are two architectures that try to solve both of these problems, namely the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), but still RNNs should be used with caution.

3.3 Audio Transformations and Representations

In signal processing, signals are basically functions that carry information about a phenomenon or an observation. Depending on the transformations applied on this information, it can be represented in a number of different ways, each one demonstrating and highlighting different parts of it and being useful to different tasks.

Audio signals belong to the family of the uni-dimensional signals, which means that they are a series of values spread across time. As such, their natural representation is the waveform, which is the graph of this time-series over time. Although this representation is simple, provides a visual interpretation of audio and doesn't require any transformation, it isn't that useful, concealing parts of the information, such as the frequency content of the signal, that are vital to many tasks. For that reason, other representations are used.

Short-Time Fourier Transform

The most commonly used time-frequency representation is the power or magnitude spectrogram of the Short-Time Fourier Transform (STFT) of a signal. Since the STFT is part of the very general and broad field of Fourier Analysis, we will only provide an intuitive explanation of it.

Fourier transform is a mathematical transformation that decomposes a complex-valued function into its frequency content. This transformation is based on Fourier Analysis which dictates that any function can be modeled as an infinite sum of weighted sinusoidal functions, according to the following formula:

$$f(x) = \alpha_0 + 2\sum_{n=1}^{\infty} A_n \cos(n\omega_0 t + \phi_n)$$

So, in this context, the frequency content refers to the magnitude A_n and the phase ϕ_n, that is the initial angle, of the participating components as functions of the frequency of the sinusoidal functions. In a sense, the Fourier Transform can be thought of projecting a signal from the time domain to the frequency domain, where much of the frequency related information is exposed.

In addition to the forward transformation, there is an inverse one, that transforms functions from the frequency domain back onto the time domain. The forward and inverse transform, for continuous functions, are defined as following:

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-i2\pi ft} dt$$

$$x(t) = \int_{-\infty}^{\infty} X(f) \cdot e^{i2\pi ft} df$$

Since we work with digital signals, the Continuous Fourier Transform defined above cannot be applied directly, as neither are the signals continuous, nor can we integrate them over an infinite range. For that reason, we work with the Discrete (forward and inverse) Fourier Transform (DFT) which are defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-i2\pi kn/N}$$

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] \cdot e^{i2\pi kn/N}$$

where n is the sample index of the original signal, k the sample index of the transformed signal and N the total length for both signals. In the case of the transformed $X[k]$, N can be thought of as the number of "frequency bins" that the transformation computes.

A problem of the Fourier Transform is that it assumes that the signals are stationary, meaning that their characteristics are constant through time. Of course, in the case of audio signal analysis and processing, this is not the case as the signals vary greatly through time and many characteristics have a finite, small time span. So, applying the Fourier Transform to the entire signal is not informative at all. A solution to this problem is to segment the signal by multiplying it element-wise with a window function and apply the transform on the individual segments. Typically, the segments are small and have an overlap to better capture the varying characteristics.

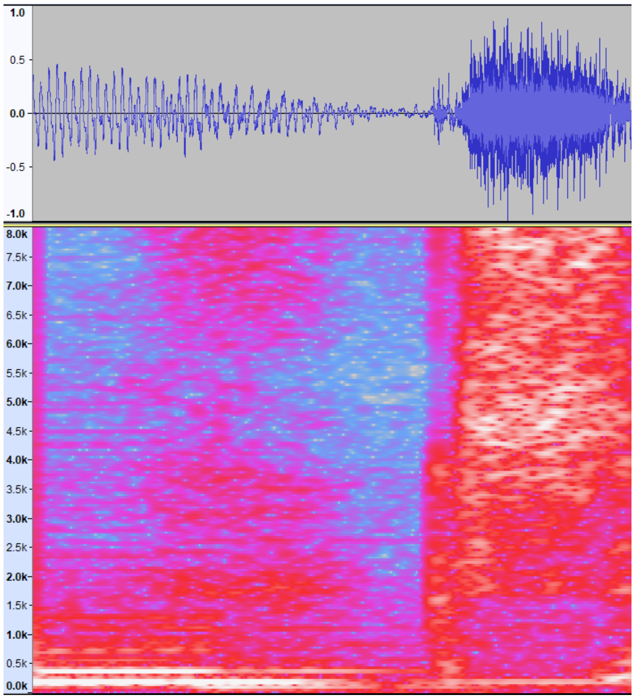


Figure 3.3.1: Waveform and spectrogram representation of a song segment. Notice that the periodic left part of the waveform, corresponds to the the part of the spectrogram with high values at two, low frequency bands, while the aperiodic, noisy right part includes a wide range of frequencies.

The above procedure constitutes the Short-Time Fourier Transform (STFT). This transformation basically creates a two dimensional representation, in which one dimension represents the frequency and the other corresponds to the time. The representation has as many columns as the number of segments T and as many rows as half the frequency bins estimated from the DFT, $N/2$. This is due to the fact that real signals have a DFT that has the property of conjugate symmetry. This means that the magnitude of the DFT has even symmetry and the phase has odd symmetry. This $N/2 \times T$ representation is called "spectrogram", Since, in practice, the complex-valued spectrogram has limited interpretability, we mostly use the magnitude, phase or power-of-magnitude spectrograms, the last of which is the most common and is often referred to as plain "spectrogram".

Filterbanks

An alternative way to access a two-dimensional representation of a signal is to use a filterbank. A filterbank is an array of bandpass filters, that is filters that process specific frequency bands, while they zero out or suppress others. By applying these filters on a signal, it is separated in multiple components, each one corresponding to a specific frequency band. Hence, the output of a filterbank is a two dimensional representation, with each row containing a band-limited, processed form of the original signal.

Depending on the number and the properties of the filters, like their bandwidth, central frequency and shape, many filterbank designs have been proposed for various tasks. For example, the mel filterbank tries to imitate the non-linear perception of sound of the human auditory system. To this end, it uses an array of triangular shaped filters with increasing bandwidth and with central frequencies placed non-linearly in frequency, according to the following formula, which transforms frequency values in Hz domain f to the corresponding values in mel scale m

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$

Figure 3.3.2 shows 40 mel filters in the frequency domain.

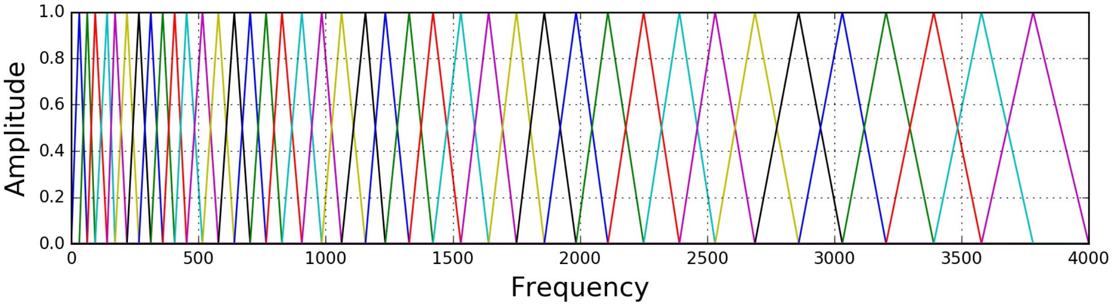


Figure 3.3.2: Depiction of mel-scale filterbank in the frequency domain [7].

Discrete Wavelet Transform

The Continuous Wavelet transformation (CWT), like the Fourier Transform, decomposes a signal in components using a set of mutually orthogonal basis functions $\overline{\psi}$, called "wavelets", which are scaled, translated and dilated versions of a base function ψ, called "mother wavelet". The original signal can be completely recovered by an inverse transformation (ICWT). The wavelets have two properties that differentiate them from the sinusoidals that the Fourier Transform uses. Firstly, they are not periodic as they are spatially localized, that is they are non-zero only for a finite time range. Secondly, the wavelets are functions of two variables, the translation which moves the non-zero part in time, and the scale which determines how dilated or compressed the function is.

The motivation behind the Discrete Wavelet Transform (DWT) is that we want to analyse a signal $x[n]$ in two components; one that constitutes an approximation of the signal, $a[n]$, which holds the general trend of the original signal, and one that describes local details, $d[n]$, which holds rapid fluctuations of it. To this end, a low-pass filter h (LPF) and a high-pass filter g (HPF) need to be used. The length and the coefficients of these filters are derived by wavelets, in order to satisfy similar properties to their continuous counterparts.

The coefficients of the two filters are connected by the following formula:

$$g_k = (-1)^k h_{n-k-1}, k \in \{0, \dots, n-1\}$$

where n is the length of the filter.

For example, for the Haar wavelet, which is one of the most common ones the h and g filter are defined as:

$$h = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]$$
$$g = \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right]$$

Due to the structure of the filters, their outputs contain half the frequency content of the original signal. Thus, a decimation by a factor of two can be safely applied afterwards, to reduce the temporal resolution and keep the total length of the output (approximation + detail signal) constant. The application of the filters along with the down-sampling operation constitutes one level of the DWT.

The above procedure can be applied multiple times for the approximation signal that is generated each time, leading to multiresolution analysis of a signal. A block diagram of this analysis is depicted in Figure 3.3.3. The end result is a set of $N + 1$ signals, $\{a_N[n], d_1[n], d_2[n], \ldots, d_N[n]\}$ where N is the number of levels of the analysis. With a sampling frequency of f_s, the approximation signal $a_N[n]$ contains frequencies in the range $[0, f_s/2^{N+1}]$ and the detail signals $d_k[n]$ contain frequencies in the ranges $[f_s/2^{i+1}, f_s/2^i], i = 1, \ldots, N$. This means that for low temporal resolution, that is low levels, the frequency resolution is high, while for high time resolution, the frequency resolution is low. This is a very important property of the DWT, that is absent from the discrete STFT, in which the time and frequency resolution remain constant, throughout the transformation.

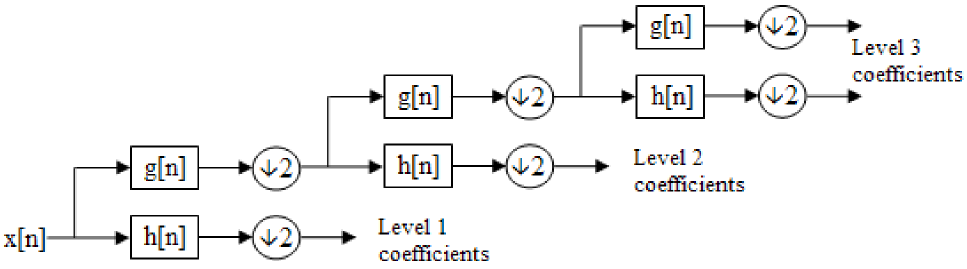


Figure 3.3.3: Block diagram of the three first stages of multiresolution analysis of a signal [5].

Chapter 4

Literature Review

Contents

The general problem of source separation, along with the special case of single channel audio separation, has been researched extensively for many years. Like many problems in the signal processing field, source separation research can be divided in a pre and post (deep) neural networks era. Although the DNN methods have dominated the field and currently are the state of the art, the traditional techniques are worth mentioning both for the sake of history and because they still provide ideas and intuition for neural network-based techniques. In fact, a state of the art DNN technique is inspired by a matrix decomposition.

4.1 DSP Methods

Source separation, as it has already been mentioned, is a multifaceted problem, and its aspects can be classified in many ways. A very basic classification can be made depending on the number of sources and sensors. In the cases where the number of sensors is greater or equal to that of the sources, the problem is classified as over-determined or determined, respectively, whereas in the cases where the sensors are fewer than sources, the problem is labelled as under-determined.

For the first two cases, matrix factorisation methods revolving around Independent Component Analysis (ICA) [25] [58] have yielded very good results.

ICA is a computational method for separating a signal into additive subcomponents, that are assumed to be statistically independent and non-Gaussian. It expresses an observed mixture from n sensors, x as the product of an $n \times p$ mixing matrix with linearly independent columns, A and p statistically independent and non-Gaussian vector signals, s:

$$x = As,$$

ICA tries to find an *unmixing* matrix, which approximates the pseudoinverse of A, $W \approx A^+$, so that the estimated components $\mathbf{u}$ are as statistically independent as possible.

$$u = Wx = WA^+s$$

Despite its elegancy and its success in the over-determined and determined cases, this method fails at monaural source separation, mainly due to the requirement that the sensors are more than the sources.

Regarding the under-determined case, and especially for the single-channel case, which is the main problem of this research, the traditional methods can be separated into three broad categories: **spectral-decomposition-based**, **model-based** and **Computational Auditory Scene Analysis(CASA)-based**.

In spectral decomposition methods, a representation of the input mixture is decomposed in basis elements which are then grouped into disjoint sets, corresponding to the individual sources. The representation of the signal can be of any form, but the most common one is the magnitude or power spectrogram of the mixture, as estimated via the STFT. The constraints of the decomposition along with the grouping criterion are the factor of differentiation of the various techniques.

One such technique is the Independent Subspace Analysis (ISA), an extension of the ICA, in which the statistical independence assumption is relaxed for the basis elements of the same group; between elements of different groups, however, the constraint still stands. In [10], a method based on ISA is used to decompose the mixture spectrogram into independent source subspaces, which yield the separated sources, after they are inverted.

Another matrix decomposition technique is the Non-Negative Matrix Factorisation. As its name suggests, NMF requires that all the participating matrices are non-negative. So, in the decomposition

$$V = WH$$

where V is the known observation matrix, H is the matrix containing the basis vectors and W is the weight matrix, every matrix element is greater or equal to zero. The non-negativity constraint equips the technique with perceptual meaningfulness, something that is lacking from most matrix decompositions. More specifically, the non-negative weights ensure that the basis elements are combined in a purely additive manner, while the non-negative basis vectors prevent the possibility of having elements that cancel each other out. As the magnitude and power spectrograms are non-negative by definition, these representations are perfectly suitable for this technique.

An NMF based technique is used in [68] [70] with an additional term that enforces temporal continuity being utilised during the estimation of the weight and basis vector matrices.

CASA tries to mimic the way that the human auditory system decodes sounds into meaningful elements. CASA methods [65] and [72] achieve this by using psychoacoustical cues, such as harmonicity and onset-offset time, and then by building streams based on pitch proximity. Due to that, these methods fail at separating overlapping sources that play the same pitch. More advanced methods use spectral filtering to allocate energy at overlapping streams [18], or add a time-frequency smoothness constraint [71]. However, they require prior knowledge of the spectral content of the sources.

Regarding model-based techniques, generative models of the source signals are created in order to perform the separation process. Because these models learn their parameters from solo excerpts, they are very sensitive to the recording environment. The models can be based on Hidden Markov Models (HMM), as for instance in [54] [23].

4.2 DNN Methods

4.2.1 Classification

With the advancement of deep learning, fully supervised techniques have been on the rise. DNN methods can be roughly classified using the following major properties:

- Based on the domain in which the data are being processed, there is the **waveform-based** methods, which utilize the waveform representation, which is the "natural", 1D representation of audio data and the **spectrogram-based**, where a 2D time-frequency representation that is derived from a transformation of the data is utilized. This transformation can be either a predefined one, such as the the Short Time Fourier Transform (STFT) magnitude or an independently learned one.

- Especially for the techniques that process spectrograms, based on the estimation of the signal, there is the **direct** estimation, in which the spectrograms of the source signals are learned directly, and the **indirect** estimation, in which the model estimates a 2D mask for each source, which is applied to the input spectrogram in an element-wise multiplicative manner, to retrieve the source signal.

The method of signal estimation is not that influential of a choice to the model's performance, apart from the fact that masking techniques have an upper limit, set by Ideal Binary Mask (IBM) and Ideal Ratio Mask (IRM), while direct estimation methods can theoretically recover perfectly the source signals. In [73], IBM and IRM for time-frequency signals are defined as,

$$\mathrm{IBM}(t,f) = \begin{cases} 1, & \text{if } \mathrm{SNR}(t,f) > 0 \\ 0, & \text{otherwise} \end{cases}$$
$$\mathrm{IRM}(t,f) = \left(\frac{\mathrm{SNR}(t,f)}{\mathrm{SNR}(t,f)+1}\right)^{\beta}$$

where β is a tunable parameter to scale the mask.

Contrarily, each choice regarding the domain of computation has its advantages and disadvantages.

On the one hand, data in the time-frequency domain are represented in a more compact way compared to the waveform domain, which means that the model requires less processing, reducing training times and also crucial information, like temporal dependencies, can be taken advantage of by less complex models, which leads to fewer trainable parameters. Moreover, models that use 2D representations can borrow techniques and ideas developed for image-related tasks, as the particular field is heavily researched and the existing solutions are generally more advanced. On the other hand, using the STFT magnitude as the time-frequency domain signal representation has several shortcomings. Firstly, the STFT is a generic signal transformation, not necessarily optimised for the task of source separation. Secondly, the vast majority of relative techniques omit the phase of the signal from the estimation of the sources, thus limiting the overall performance, since part of total information is ignored. As a result, the phase of the separated signals is not estimated; in order to circumvent this problem, these methods either take for granted that the source phase is identical to that of the mixture [12], or find an approximation by applying the Griffin-Lim algorithm [47], which is slow and often unsuccessful [32]. Unlike magnitude, phase cannot be estimated easily by DNNs, due to its periodic nature, that creates discontinuities at the wrapping point (e.g. if the value range is $(-\pi, \pi]$, the wrapping point is π). Phase unwrapping could be used as a solution, but it solely transforms the problem, as it greatly increases the value range, again making it hard for DNNs to estimate the phase successfully. However, it has been shown that phase information is very vital in speech enhancement tasks [20] and in audio separation tasks. In fact, [64] achieves good results by approaching the problem of phase estimation in a novel manner, as it casts the regression problem to a classification problem.

4.2.2 Methods using Time-Frequency (2D) Representations

Representing audio signals in the time-frequency domain, mainly using STFT, has been very common in various audio tasks. Thus, it is only natural that initial research in source separation using DNNs was performed in this domain. This, along with the fact that the majority of the well-performing techniques utilise this domain, is the reason we begin our literature review with this family of methods, although this thesis proposes networks that operate in the waveform domain.

Initial work was mainly done using simple and relatively shallow neural networks, because the resources and the available machine learning tools and frameworks at the time were limited and not as established as they are today. In [22], one of the earliest works to tackle the problem of speech separation research using DNNs, a very simple network is used to combine single frame estimates provided by NMF in a nonlinear way, as the linearity of the combination of the basis vectors is considered to limit the separation capabilities, whilst the use of nonlinear activation functions between layers seems to improve the expressibility of the model. In [66], instead of a single spectrum frame, the network is fed with a set of neighbouring frames to provide temporal context and use the relative information.

In the following years, DNN became deeper and deeper, combining various layers in complex ways in order to improve separation performance. Because audio signals may have arbitrarily long temporal dependencies, recurrent layers have been incorporated into architectures so as to handle long sequences of frames efficiently. In [24], deep recurrent neural networks (DRNN) are used and different temporal connections are explored. In [43], an autoencoder architecture is implemented using a bidirectional GRU (BiGRU) as the encoder and a unidirectional GRU for the decoder. In [42] and [16], the idea of recurrence is adopted for mask prediction in a very interesting manner. Simply put, an iterative algorithm that runs until a convergence criterion is met, is used to employ a stochastic depth to that specific network part, thus giving great flexibility and expressibility to the model. In [26], an adaptation of *U-Net* [51] for spectrogram-based music separation is proposed. The architecture is a deep autoencoder, comprised of a series of 2D convolutional layers, applied on multiple scales through upsampling and downsampling, in order to capture both local and global patterns. The convolutional layers have been additionally used in one dimension to individually learn temporal and timbre information, as in [13], which uses "vertical" and "horizontal" convolutions to achieve that. *MMDenseLSTM*, an architecture that integrates both recurrent and convolutional layers in a unified pipeline is proposed in [63]. It first splits the input spectrogram in multiple frequency bands, and then proceeds in a similar way as before, by processing each band, as well as the full spectrogram on multiple scales using convolutional layers, and additionally using an LSTM module at the bottleneck. This architecture achieved competitive performance among architectures that operate in the time-frequency domain. After performing the source separation, some methods have an additional denoising step, influenced by speech enhancing tasks. In [42], [16], the authors use a fully connected autoencoder, that consists of an feedforward neural network (FNN) encoder and an FNN decoder. while [43] uses a highway network [60] and a generalized Wiener filtering process, heavily influenced by audio denoising tasks.

4.2.3 Methods using Time/Waveform (1D) Representations

Methods in the waveform domain have in common that they do not use predefined frontends for feature extraction, such as the STFT and that they process data as one-dimensional. They achieve that by either using a learned transformation that leads to a latent representation, or by processing the input data directly, in an end-to-end manner. Another motivation for using 1D techniques is that in the waveform domain there is no loss of phase information that occurs when using the STFT magnitude, as mentioned in 4.2.1.

The one-dimensional aspect of these methods does not necessarily mean that the latent representations are themselves one-dimensional. Instead, it means that the data is treated by the various processing layers as being multi-channelled one-dimensional. So, the respective layers have one-dimensional activation maps. For example, when a previous method would have used a 2D CNN to process the data, these methods would use 1D convolutions.

In this category of methods, there are two major architectures which constitute the basis of almost every technique; the *TasNet* [36], which was originally used for the speech enhancement task and the *Wave-U-Net* [61]. Since both of these architectures are explained thoroughly in the following chapters, below we will provide a concise overview of them and their variations.

TasNet

TasNet is a technique that learns a latent representation from the mixture waveform, in order to retrieve the source signals by performing masking upon it. Simply put, it is comprised of three parts: the **encoder**, the **decoder** and the **separator**. The architecture's pipeline is the following:

- The encoder performs a decomposition of the input signal into a set of basis vectors and weights.
- These weights are processed by the separator to produce a set of masks, one for each source.
- These masks are applied over the weights in an element-wise multiplicative manner and the final product is passed through the decoder to reconstruct the source signals.

In the original paper [36], the encoder and the decoder consist of a strided convolutional layer with a relatively large window size that alters the feature channels of their input. The separator is comprised of a deep LSTM network that models the temporal dependencies of successive segments, followed by a fully connected layer.

A number of variations of this base architecture alter the implementation of one of the modules, in order to achieve better performance. Typically, the module that gets changed is the separator, while the encoder and the decoder remain unaltered. In [35], the idea of the RNN separator used in the original model is further refined, by splitting the input into fixed-size blocks and incorporating RNNs that process the data on both feature and channel dimensions. Thus, the network extracts both inter and intra block information, separately modeling local and global dependencies. Although the above approach is very successful, the training of the RNNs is slow and processing of long sequences increases the model's processing and training time. *Conv-TasNet* in [37] tackles these problems by using the Temporal Convolutional Network (TCN) [33], which is a series of residual blocks using dilated, depthwise convolutions to capture information on multiple scales. Although, unlike RNNs, the TCN does not have an infinite receptive field, it provides a rich set of features, an immense speedup on training times and a reduction on trainable parameters, making the implementation of [14] one of the top-performing in music separation. In [57], *FurcaPorta* is introduced, further increasing the separator's performance by adding two gating mechanisms to the TCN; the first gate controls information inflow and the second controls information processing and outflow. A completely different approach on modifying the separator, inspired by meta-learning models, is proposed in [56]. Essentially, a generator network learns source specific information and generates the parameters of the separator, adapting it to the separation of a specific source and thus making the model able to separate multiple sources and not just a specific one.

Even though the separator is the major point of interest in the TasNet architecture, significant research regurding the encoder and the decoder used has also been carried out. In [27] a deep, non-linear encoder is proposed, by stacking a series of convolutional layers after the first strided one (respectively before the last one for the decoder), which improves performance by a small but significant amount. On a completely different approach, the authors of [15] propose to use a fixed, *Multi-Phase Gammatone Filterbank* (MP-GTF) as the model's encoder instead of a learned one. Although a learned encoder provides more degrees of freedom, hence increasing the expressibility of the model, it also adds more variance to it, due to the increased number of parameters, rendering it more vulnerable to overfitting and thus less capable of generalizing. The proposed MP-GTF is a variation of the Auditory Gammatone Filter (A-GTF), which is a filterbank that tries to model the non-linear perception of sounds of the human auditory system, adapted to be used within Conv-TasNet. It does so by using non-linearly spaced narrow-band filters with an increasing bandwidth over the filter's center frequency, whose impulse responses are given by a gamma probability distribution function multiplied by a sinusoidal tone. Using MP-GTF as the encoder and pairing it with a learnable decoder improves the network's performance probably thanks to supplying the rest of the network with representations of the input that are more meaningful and suitable for the separation task. In [56], a stronger encoder and its respective decoder are proposed as improvements to the original Conv-TasNet architecture. This encoder combines two sets of features to enrich the representation that is fed to the separator. The first set comes from an array of convolutional layers with different receptive fields, that extract features in multiple scales, while the second set comes from a spectrogram representation.

Wave-U-Net

The second architecture is simpler and more straightforward. *Wave-U-Net* is an 1D adaptation of *U-Net* [52]. The aim of the network is to create a dense representation of the input mixture signal, which is fitting for the separation task and then restore the source signals to the original representation. In simple words, the network consists of the following parts:

- The **downsampling path**, which is a series of downsampling blocks that extract information in multiple scales.
- The **bottleneck**, which processes the dense representation.
- The **upsampling path**, which is the opposite of the downsampling path, as it takes the dense representation and restores it to its original size by a series of upsampling blocks. Additionally, these blocks concatenate the input features from their previous block in the upsampling path and the features from the downsampling path that are provided through skip connections between blocks that correspond to the same depth.

In the original paper [61], the downsampling blocks contain convolutions that increase the channels of their input, in order to widen the representation and pooling layers in order to downsample it in the temporal dimension. The bottleneck is a convolutional layer that performs a transformation of the features without increasing or decreasing the feature channels. The upsampling blocks increase the feature map size using linear interpolation and concatenate their input and the data coming through the skip connections using a convolutional layer that also decreases the feature channels. The last upsampling block outputs as many channels as the number of sources.

Further research based on Wave-U-Net attempts to increase its performance by changing the resampling block structure, the bottleneck or the combination of the features from same level blocks. The model of [29] tries to make use of the long-range temporal dependencies of the data by using two BiLSTM layers followed by a fully connected layer as the bottleneck. In [21], instead of simply concatenating features in the upsampling path with their downsampling counterparts, an attention mask is introduced in order to identify relevant and meaningful features. The authors of [44] focus on the decimation procedure. More specifically, they support that since the downsampling layers do not apply any kind of anti-aliasing filter, they propagate information containing high-level artifacts thus destroying a part of the information. The retrievability of the lost information through skip connections, which are partially used for this exact reason, depends solely on training. To solve this problem, they suggest resampling the feature map sizes by performing a Discrete Wavelet Transformation (DWT) instead of the previously-used pooling layers. This transformation works as an anti-aliasing filter and satisfies the perfect reconstruction property. The used implementation fits the Wave-U-Net architecture perfectly and achieves a slight performance improvement, without introducing additional parameters and with only a slight, constant increase of the training time.

4.3 Datasets

As we saw above, the solutions of the problem of music separation have shifted from traditional DSP techniques to fully supervised end-to-end deep neural networks. This has made the existence of well-constructed datasets for training and evaluation an absolute necessity. In [26], the authors, having access to a music database, create a dataset suitable for singing voice separation by subtracting instumental version of songs from the original mixtures to get the vocal component. Although they created a huge dataset of 20,000 track pairs, this method is prone to errors, as it is not fully supervised.

For that reason, most researchers use datasets that are constructed especially for that use, like MedleyDB [8], iKala [11], DSD100 [45], slakh [38], DALI [40] and MUSDB18 [50].

MedleyDB [8] is a dataset that was created mainly to promote research on melody extraction. It contains 122 songs for a total length of 7.17 hours of audio. The songs are in stereo format, recorded with a sampling rate of 44.1kHz. For each song, the mixture signal is accompanied by the processed stems, raw audio and metadata, along with annotations about the melody f0, the instrument activations and the genre.

The iKala dataset [11] was especially created for the task of singing voice separation. It consists of 252 30-second excerpts, sampled at 44.1kHz. Each signal has two channels, with the one being the music and the other the voice component. The dataset also contains pitch contour annotations and lyrics with timestamps for every excerpt.

Synthesized Lakh (Slakh) [38] is a dataset of multi-track audio, designed for the tasks of music source separation and multi-instrument automatic transcription. Each track is synthesized from the Lakh MIDI Dataset v0.1 [49] with the use of professional-grade sample-based virtual instruments. The end result is mixed together to create musical mixtures. The dataset contains 2100 tracks for a total of 145 hours of mixture data, along with aligned MIDI files synthesized from 187 instrument segments that are split into 34 classes.

The DALI dataset [40], in its current, second version, contains synchronised (vocal/instrumental) audio 7756 songs. Each song is accompanied with time-aligned lyrics, which appear in four levels of granularity, and with time-aligned symbolic vocal melody notations. Additionally, for each song there is provided multimodal information such as genre, language, musician, album covers or links to the respective video clip.

The DSD100 dataset [45] consists of 100 songs of different musical genres, all of which are of stereo format and sampled at 44.1kHz. The songs mixtures are accompanied by the isolated drums, bass, vocals and other stems.

The MUSDB18 dataset [50] is the most popular in music separation related papers and it was used on all the conducted experiments. Its sources include both the other two aforementioned datasets (MedleyDB and DSD100) and other material. It contains 150 songs of various musical genres, for a total of 10 hours of audio. The songs are of full-length, in stereo format, recorded in high quality and sampling rate of 44.1kHz. Apart from the mixture, the dataset contains 4 individual signals (stems) for each song, which correspond to 4 predefined categories (vocals, bass, drums and the rest of the accompaniment, denoted as "other") in order to facilitate and promote multi-instrument separation. The song mixture is a linear combination of the individual sources, thus, the sum of the 4 stems returns the original signal. The dataset has a default train-test split of 100-50 songs.

4.4 Source Separation Evaluation Metrics

The general topic of Blind Audio Source Separation (BASS) has been an active research topic for many years, with many successful techniques yielding good results. However, because of the subjective nature and difficulty of the task, the quantification of the performance as well as the comparison between several techniques requires the use of a widely accepted and of high quality evaluation metric.

Historically, various metrics have been used, such as Inter-Symbol Inteference (ISI) [31], or the MSE between L_2-normalized versions of the sources. Both of these measures, although relevant to the problem, suffer from limitations, with the most important being that they consider the dsired signal $\widehat{s}_j$ to be recovered up to a permutation and a gain, and not any other distortion, which is restricting for a number of applications. Also, since these measures provide a single performance metric, they are not able to distinguish the various error terms, such as the sensor noise, the interferences between various sources, the spectral correctness of the extracted source and the introduction of unrelated artifacts. This separation is crucial to the assessment of the technique. In particular, not all error terms affect the perceived result in the same way, with artifacts being for instance more noticable and pervasive than the sensor noise, and some applications are more sensitive to one type of error over the others. In a separation task the errors can be classified in three categories; sensor noise, e_{noise}, interference from other sources, e_{interf}, and "burbling" artifacts, e_{artif}, with the last being considered the most annoying, while the first the less noticeable. Hence, a technique could have scored higher in the metrics, while having worse perceived performance, due to a different mix of error terms.

A toolkit that contains metrics that attempt to solve the above issues is BSS Eval. This toolkit was originally created for MATLAB, but is widely used in the python community through the museval package. The metrics provided by the BSS Eval toolkit are the Source to Distortion Ratio (SDR), Source to Interference Ratio (SIR) and Source to Artifacts Ratio (SAR), as defined in [69]. These metrics can be configured to allow the signals to be recovered up to a time-invariant filter and a time-invariant gain, in order to more closely adhere to the application's needs.

Regarding the computation of these metrics, it is assumed that the estimated source signal, $\widehat{s}_j$ is decomposed in 4 terms, as such: $\widehat{s}_j = s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}$. The proposed decomposition is based on orthogonal projections of the source signals onto subspaces spanned by the source signals and/or the sensor noise. Therefore, the metrics are defined as:

$$\begin{aligned}
\text{SDR} &:= 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}\|^2} \\
\text{SIR} &:= 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}}\|^2} \\
\text{SAR} &:= 10 \log_{10} \frac{\|s_{\text{target}} + e_{\text{interf}} + e_{\text{noise}}\|^2}{\|e_{\text{artif}}\|^2}
\end{aligned}$$

SDR is considered to measure the overall quality of the separated signals, while SIR and SAR quantify the clarity of the separated sources and the existence of auditory artifacts, respectively. Among these three metrics, usually SDR is considered the most important as the closest to human perception, although SAR and SIR can be equally or more important, depending on the application.

Although the BSS Eval toolkit has been widely accepted and is used in the literature for the evaluation of source separation algorithms, there are still a couple of issues that should be taken into consideration, regarding the allowed distortions and the correlation between the metrics and the human perception.

Regarding the first point, the distortions that are allowed by the BSS Eval toolkit can deform the reference in such an extent that it matches any estimated signal [53], thus hindering the objectivity and the credibility of the evaluation of diefferent algorithms. More specifically, the space of signals that is accessible by convolving the reference with a short FIR filter, the use of which is justified as a counterbalance to the room impulse response (RIR), is huge and can lead to signals that differ a lot from the original.

For that reason, the authors of [53] have proposed scale-invariant versions of the above metrics, that replace and improve/redefine the aforementioned ones. More specifically, for a mixture $x = s + n$ of a target signal s and an interference signal n, the proposed metrics use a scaling factor to rescale the target so that the residual signal $s - \hat{s}$ is orthogonal to it. The optimal such factor is obtained as $\alpha = \hat{s}^T s \|s\|^2$, the scaled reference is defined as $e_{\text{target}} = \alpha s$ and the estimate is decomposed as $\hat{s} = e_{\text{target}} + e_{\text{res}}$. A further decomposition of the e_{res} as $e_{\text{res}} = e_{\text{interf}} + e_{\text{artif}}$, where e_{interf} is defined as the orthogonal projection of e_{res} onto the subspace spanned by s and n is used to define the scale-invariant metrics:

$$\text{SI-SDR} = 10\log_{10}\left(\frac{\|e_{\text{target}}\|^2}{\|e_{\text{res}}\|^2}\right)$$
$$\text{SI-SIR} = 10\log_{10}\left(\frac{\|e_{\text{target}}\|^2}{\|e_{\text{interf}}\|^2}\right)$$
$$\text{SI-SAR} = 10\log_{10}\left(\frac{\|e_{\text{target}}\|^2}{\|e_{\text{artif}}\|^2}\right)$$

Unlike the BSS Eval toolkit, in which there was not a clear, intuitive relationship between the three metrics, the definitions of these scale-invariant metrics are connected by the following formula:

$$10^{-\text{SI-SDR}/10} = 10^{-\text{SI-SAR}/10} + 10^{-\text{SI-SIR}/10}$$

since, due to the orthogonal decomposition, it stands that $\|e_{\text{res}}\|^2 = \|e_{\text{interf}}\|^2 + \|e_{\text{artif}}\|^2$.

Regarding the second point, the metrics that are provided by the BSS Eval toolkit are objective measures of separation quality. That is, they mathematically model how successful or not an algorithm was at separating the signals. However, this does not necessarily mean that humans perceive the results in the same way. For that end, the authors of [74], report that another evaluation toolkit, PEASS [17], seems to correlate better with human perception than BSS Eval.

Chapter 5

Wave-U-Net

Contents

The goal of this chapter is a) to present Wave-U-Net, an architecture that was proposed in [61] as a solution for the music source separation task, and b) to examine a number of extensions that were proposed by [29] and [44] upon this base architecture, as well as their modularity. On this architecture, our main goal was getting familiar with the task, framework, hyperparameter tuning, and performing an ablation study of existing extensions rather than actually promoting novel improvements to it.

5.1 Baseline Architecture

Wave-U-Net [61] is an end-to-end model that processes audio signals on the time domain. This means that, on the one hand, the model estimates the samples of the source signals directly and not by applying a mask to the input or a latent representation and, on the other hand, the latent representations and feature maps, although they might be multi-channeled, are one dimensional, processed by 1D layers.

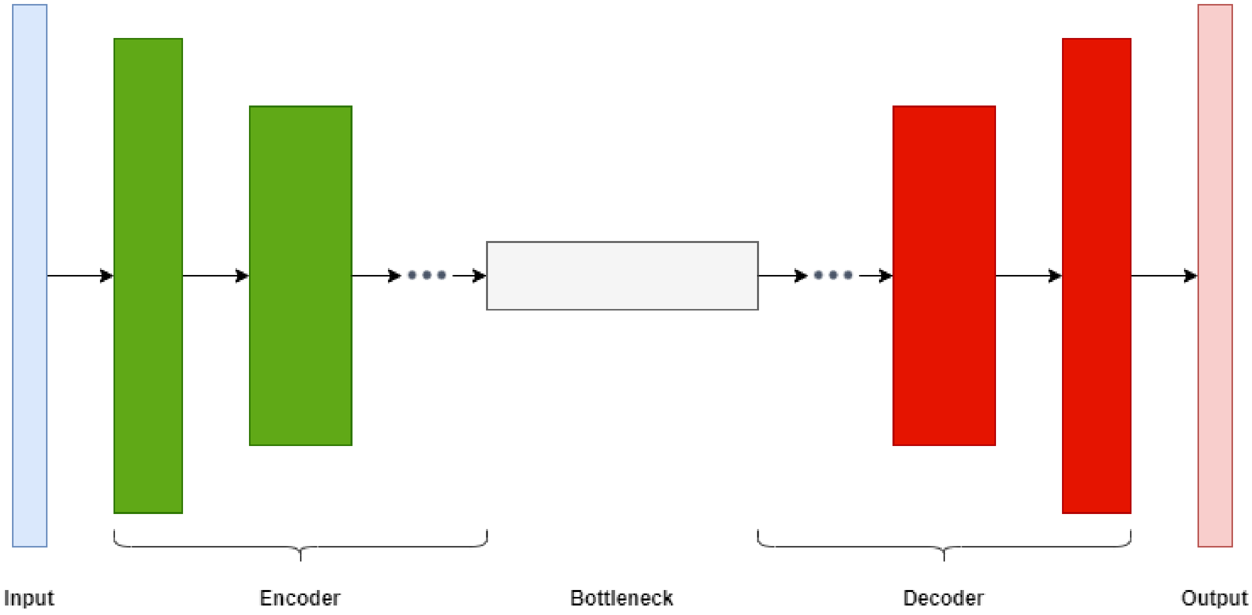


Figure 5.1.1: Abstract representation of an autoencoder network.

Wave-U-Net is the one dimensional adaptation of U-Net [51], a model that is used for biomedical image segmentation. It has an autoencoder architecture, which contains four parts.

- An encoding or downsampling path, which takes the original signal as an input and repeatedly processes and downsamples it until a dense latent representation is generated.
- The bottleneck, which is constituted by the processing and computation performed onto the dense representation.
- A decoding or upsampling path, which operates in an opposite manner than the encoder, as it receives the processed representation and repeatedly combines features from previous layers and upsamples them until it restores the representation to the original resolution.
- An output layer that performs the desired task.

In more detail, Wave-U-Net has a depth of L levels, meaning that each of the encoding and decoding paths contains L processing blocks. Each processing block contains an 1-D convolutional layer with LeakyReLU activation and a resampling operation. At the encoding path the convolutional layer precedes the downsampling, while the opposite stands at the decoding; the feature extraction follows the upsampling. The bottleneck and the output layer consist of a single 1-D convolutional layer, with no resampling involved and with LeakyReLU and tanh activations respectively.

The model receives segmented mixture signals $\mathbf{M} \in [-1,1]^{L_m \times C}$, where L_m is the segment length (in samples) and C corresponds to the number of audio channels (1 for mono, 2 for stereo). The first downsampling block increases the channels number from the original C to a fixed number F_c. The rest of the convolutional layers of the downsampling path and the bottleneck increase the number of channels by F_c, as they aim to extract increasingly information-rich features to form and process the dense representation. At the deepest point, after the bottleneck, the latent representation has a total of $(L+1) \cdot F_c$ channels. The upsampling path works in the opposite way than the downsampling one; the convolutional layers decrease the channel number, in order to match the channels of the same-depth block of the downsampling path and gradually restore the channel number of the representation. With a total of L upsampling blocks, the channels are reduced to F_c. In the original paper, the convolutions of the downsampling path and bottleneck have a kernel size of 15 and those of upsampling path a size of 5.

Between blocks of the same level there are skip connections, which, as their name suggests, enable information from the downsampling blocks to reach the upsampling ones uninterrupted, skipping the intermediate processing. The features coming from the encoder are concatenated with the ones in the decoder, before getting processed by the convolutional layer. This is done for two reasons: Firstly, it allows high level details, that might have vanished due to the downscaling, to flow directly to the signal restoration stage, which can greatly contribute to a correct restoration of the signals. Secondly, it facilitates training for the earlier layers, as it deals with the vanishing gradient problem. This problem is common in deep architectures, where the backpropagation algorithm assigns gradients to the nodes that get lower and lower, as it moves backwards through the network, making gradient descent, and thus training, extremely slow. With the use of skip connections, earlier stages receive gradients from both paths (normal and skip one) which partially mitigates the vanishing gradient problem.

Finally, the output layer receives the multi-channeled representation at the original resolution of the input mixture and performs the separation with a convolutional layer that changes the number of channels to $K \cdot C$, where K is the amount of sources, which can then be split, to retrieve the K source signals. Figure 5.1.2 displays the above architecture in detail.

Regarding the resampling, both operations change the feature map resolution by a factor of two, with the downsampling halving the resolution and the upsampling doubling it. This enables the convolutional layers to extract features on multiple scales, without the need to use kernels of different sizes, that would lead to an increase of the computation cost. Also, the fact that deeper levels process downscaled feature maps compared to shallower ones helps in maintaining the computation cost at reasonable levels, despite the former applying a significantly larger number of filters on the data than the later ones. Thus, for an input segment with a length of L_m, the representation at the bottleneck (deepest point) of the network would have a length of $L_m/2^L$ samples, vastly reducing computation needs. The resampling is implemented with a simple decimation layer for the downsampling, and linear interpolation for the upsampling, avoiding aliasing artifacts that could appear by simply decimating the signal or using transposed convolutions, respectively.

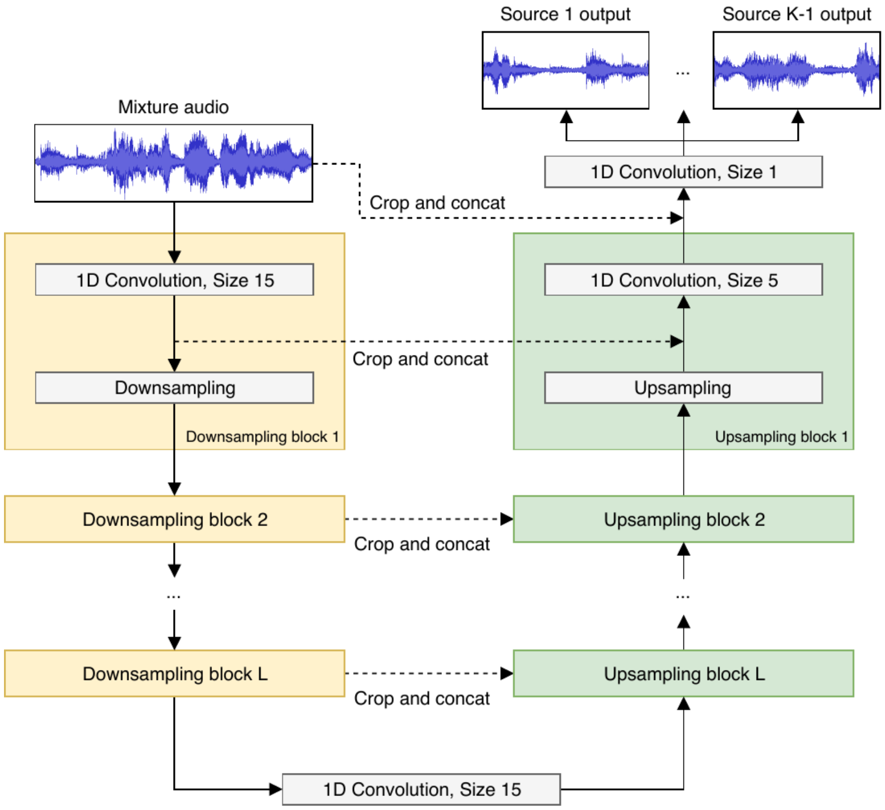


Figure 5.1.2: Wave-U-Net with K sources, a depth of L and with the additions of a) difference output layer and b) bigger input context [61].

5.2 Modifications

The Wave-U-Net architecture has been the subject of research for many papers, and as a result, several modifications that improve the performance or reduce the complexity of the model have been proposed. Here we will present 4 that were used in our experimentation.

5.2.1 Bigger Input Context

The first modification, which is proposed in the original paper [61], deals with a known problem of convolutional layers; that of the feature map size reduction.

As it has already been mentioned, since convolution kernels need to entirely fit inside data, the resulting feature map is smaller than the input. To avoid this shrinkage, which can have adverse effects to the model's performance and training, a padding layer is often used. But, despite the padding technique used, the pads introduce false information, which can corrupt the convolutions near the edges of the signal. There are cases where the effect of padding is negligible, due to the nature of the data. However, in the case of music processing, where the signals are too big to be processed as a whole and are instead cut in segments of a few seconds or less, the use of padding has an effect similar to the addition of false sounds before and after each segment. For instance, zero-padding corresponds to adding silence before and after each segment, which, obviously, is erroneous and ultimately leads to bad results.

This problem is extremely apparent in Wave-U-Net, because the ratio of the kernel size to the feature map size and thus the amount of corrupted information due to padding, increases progressively as we proceed to deeper levels. In fact, depending on the exact hyperparameters of the model, this ratio can even get greater than one, for multiple levels, which means that every output element of the feature map is more or less directly affected by the padding.

So, since using padding to counter the feature map reduction creates more problems to the model, it is proposed to avoid using it entirely and to deal with the reduction just by using larger context windows. In practice, for a given input segment with length L_m, the predicted source signals will have a smaller length $L_s < L_m$, which means that, in order to achieve the same output length, a model with larger context window will need larger input segments, greatly increasing the memory requirements of the model during training.

To incorporate bigger context windows into Wave-U-Net, the concatenation operation requires that the feature maps of the downsampling path that are forwarded through the skip connections are cropped to the size of the respective maps of the upsampling path. Apart from that change which adds a minuscule amount of computation, no further alteration to the model's layers and parameters is needed.

5.2.2 Difference Output Layer

This modification, which, again, is proposed by the original paper's authors [61], makes an assumption about the nature of the data to simplify the output layer. The assumption is that the source signals are combined in an additive manner, which is valid for the MUSDB18 dataset, as the sum of the individual source signals results in the mixture signal, but whether it holds generally depends solely on the data.

Regarding the modification, assuming that for a mixture signal $\mathbf{M}$, that consists of K source signals $\mathbf{S}_j, j = 1 \dots K$, it stands that $\mathbf{M} = \sum_{j=1}^{K} \mathbf{S}_j$, the output layer predicts only $K-1$ source signals and computes the last one as $\widehat{\mathbf{S}}_K = \mathbf{M} - \sum_{j=1}^{K-1} \widehat{\mathbf{S}}_j$. By constraining the model in this way, it doesn't need to learn this rule through training, which could speed up the learning process and improve performance.

5.2.3 RNN Bottleneck

In [29], the addition of a recurrent layer at the bottleneck, such as an LSTM or BiLSTM, before the convolutional layer of the baseline is proposed.

The motivation behind this extension is the fact that convolutional layers, due to their nature and structure, have a small, finite receptive field. Thus, they only process local correlations of the signal and are able to discover local patterns. The current workaround of repeated downsamplings of the signal to increase the convolutions' receptive field, although it has been shown to work, requires very deep and complex architectures that are slow and hard to train and creates very abstract latent representations, which can hinder the performance, due to loss of detail. Recurrent layers, on the other hand, with their infinite receptive field can be included to the bottleneck to solve this issue. It is expected that the integration of these layers will not only enable the use of shallower models, reducing the network's parameters and complexity in the process, but they will also improve the performance, as they are more suitable for the processing of sequential data, like audio signals.

The integration of recurrent layers to the current architecture needs some attention, because the semantic conversion from multi-channeled feature maps to sequences of feature vectors is non trivial. In this case, since the feature map dimension is the temporal one, it is treated as the sequence dimension from the RNN, while the channel dimension has the role of the feature vector. This semantic remark is expressed in practice, as a transposition of these dimensions before and after the recurrent layer is necessary to get correct results.

5.2.4 Incorporation of Discrete Wavelet Transformation

The last proposed improvement to the architecture is the use of the Discrete Wavelet Tranformation (DWT) at the resampling blocks of the encoding and decoding path [44].

The motivation behind this modification is that the decimation process causes aliasing and is not perfectly invertible, which means that parts of the feature maps are discarded. Both of these issues can cause loss of performance, as on the one hand, aliasing corrupts information, creating audio artifacts that the rest of the model cannot easily get rid of and on the other hand, the discarded information may be vital to the separation task. Although the existence of skip connections can compensate partially or totally for the discarded information, whether this happens heavily depends on training, as there is no training bias imposed on the architecture, resulting in an inconsistent behaviour that is undesired.

The use of wavelet transformation as a resampling operation can solve both these issues, as it has an anti-aliasing filter and satisfies the perfect reconstruction property. The integration of the transformation into the architecture is done by creating two new resampling modules, one for the direct and one for the inverse transformation, that replace the original ones. The transformation module operates as following:

Let us consider the feature map $\mathbf{z} \in \mathbb{R}^{T \times C}$, where T is the number of audio samples, which is assumed even for simplicity (if it wasn't even, a padding of one sample could be used). The feature map of each channel is first split in odd and even parts, $\mathbf{z}_c^{\text{odd}} \in \mathbb{R}^{T/2}, \mathbf{z}_c^{\text{even}} \in \mathbb{R}^{T/2}$ depending on the time sample indexing. The even component is considered as the one that will proceed to the rest of the network, while the odd one has a supporting role to the whole procedure. More specifically, the odd component is predicted by the even one using the prediction operator $\mathcal{P}$ leading to an error

$$\mathbf{e}_c = \mathbf{z}_c^{\text{odd}} - \mathcal{P}\mathbf{z}_c^{\text{even}}$$

This error term contains high frequency information. Because $\mathbf{z}_c^{\text{even}}$ contains aliasing artifacts due to the decimation, the error term is used to generate a smoothed version of it, termed as $\mathbf{s}_c$, using the update operator $\mathcal{U}$, as

$$\mathbf{s}_c = \mathbf{z}_c^{\text{even}} + \mathcal{U}\mathbf{e}_c$$

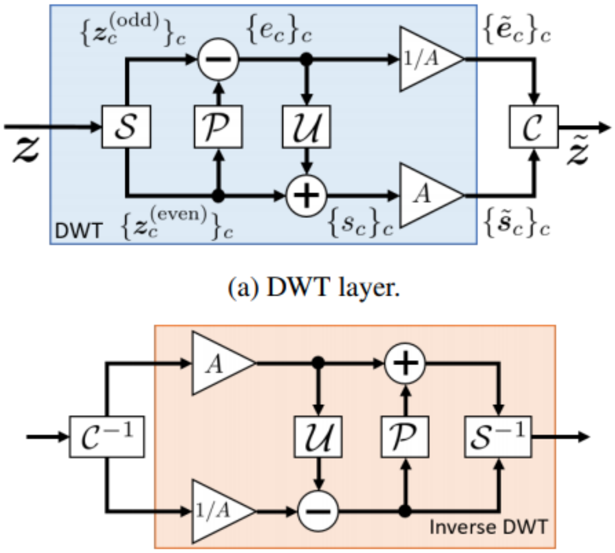
Both $\mathbf{s}_c$ and $\mathbf{e}_c$ components are then scaled by a normalization constant A and its reciprocal respectively, resulting in

$$\tilde{\mathbf{s}}_c = A\mathbf{s}_c, \tilde{\mathbf{e}}_c = \mathbf{e}_c/A$$

Finally, these normalized components are concatenated with form the downsampled feature map

$$\tilde{\mathbf{z}}_c = [\tilde{\mathbf{e}}_1, \ldots, \tilde{\mathbf{e}}_C, \tilde{\mathbf{s}}_1, \ldots, \tilde{\mathbf{s}}_C] \in \mathbb{R}^{T/2 \times 2C}$$

The values of operators $\mathcal{P}, \mathcal{U}$ and the constant A depend on the wavelet used. For the Haar wavelet, for example, the values of the operators are $\mathcal{P} = I_{T/2}, \mathcal{U} = 0.5 \cdot I_{T/2}$ and of the constant $A = \sqrt{2}$, where I_T is the identity matrix of size $T \times T$. In this case, the signal s_c is equivalent to the resulting signal after applying average pooling. The inverse module performs the opposite operation. Figure 5.2.1 displays the block diagrams of the two modules.



(a) DWT layer.

(b) Inverse DWT layer.

Figure 5.2.1: Block diagrams of the proposed DWT resampling layers. C and S denote the concatenation and splitting operation respectively, while C^{-1} and S^{-1} are their inverses [44].

In spite of the two new resampling modules not being learnable by any way, as they only contain fixed parameters, they impose a small, fixed amount of computation, which along with the subsequent increase in input channels of the following convolutional layer from C to $2C$, can have a slightly noticeable impact on training times. However, it is expected that the benefits in performance greatly surpass the additional computation cost.

5.3 Experiments

Our goal with the experiments on the Wave-U-Net architecture was to investigate in what way the hyperparameters and the modifications affect the performance and whether the latter can be

	L	F_c	#params	Input Context	Difference Layer	LSTM	DWT
M1	12	24	6.07M	✓	×	×	×
M2	6	48	3.63M	✓	×	×	×
M3	12	24	13.45M	✓	×	✓	×
M4	9	32	10.13M	×	×	✓	×
M5	12	24	6.07M	✓	✓	×	×
M6	12	24	13.45M	✓	✓	✓	×
M7	12	24	7.12M	✓	✓	×	✓
M8	9	32	10.98M	✓	✓	✓	✓

Table 5.1: Configurations of the Wave-U-Net trained models.

combined, or not. Since the individual modifications have been previously evaluated by the respective papers, we focused on training models that combine modifications and hyperparameters configurations.

Experimental Setup

Regarding the experiment setup, we used the MUSDB18 dataset, with a 75-25 train-validation split, at stereo format, downsampled to 22.05kHz, as commonly done in bibliography to reduce the training cost [61, 29, 44]. Regarding the data augmentation, we apply random amplification on the signals' magnitude, in the range $[0.7, 1]$.

The models were trained on the task of singing voice separation, thus we used the stems corresponding to the whole mixture and the vocals, and created the ground truth accompaniment signals by subtacting the vocals from their corresponding mixtures. All models output an audio segment of 16384 samples, that is around 0.74 seconds. We trained each model using the Adam optimizer with a cyclic lr [59] of two cycles in the range $[5 \cdot 10^{-5}, 10^{-3}]$. The loss function was L2 and the models were trained for 50 epochs, using early stopping if the validation loss was not improving for 20 consecutive epochs. The loss was calculated as the average of the vocals and accompaniment loss. The models with the difference output layer modification predicted the accompaniment component and calculated the vocals one, by subtracting the accompaniment from the mixture.

To evaluate our results, we used the SDR, SAR and SIR metrics, using the median-of-medians protocol devised in [62]. According to this protocol, the metrics were calculated for both the estimated vocal and accompaniment components of each segment. The segment-wise scores were aggregated over each song by calculating their median and then the median of the per-song scores is computed throughout the whole test set, resulting in a single value.

Table 5.1 displays the features and the number of parameters of the models that were trained, which are the following:

- M1 is the baseline of this set of experiments, which is a reproduction of of [61] with the use of bigger input context.
- M2 is a shallower version of M1, having half its depth. In order for the latent representation to

achieve a similar number of channels, the number of channels altered per block, F_c, is doubled.

- M3 includes an LSTM to M1 model. The incorporation of the recurrent layer is performed before the convolutional layer of the bottleneck, as mentioned before. LSTM's hidden dimension is set to 600.
- M4 is a shallower version of M3, without the input context.
- M5 includes a difference layer to the M1 model.
- M6 incorporates both LSTM and difference output layer extensions to M1, acting as a combination of M3 and M5 models.
- M7 adds DWT-inspired resampling blocks to M5 model.
- M8 incorporates all three extensions to a shallower baseline M1 model. The hyperparamters regarding the depth of the architecture and the width of the latent representation are similar to M4 model.

First of all, as it can easily be observed by the experiments, we took bigger input context for granted and included it in all but one experiments. This is due to the fact that, apart from the increased memory usage which wasn't a problem at this stage, the use of bigger context is the ideal way of performing the convolution operation and is beneficial for Wave-U-Net for the aforementioned reasons, something that is confirmed by the results from the original paper.

Secondly, as for the DWT resampling blocks, again, the results from the original paper displayed an increase in performance, so we proceeded in using them in conjunction with other modifications, without examining their contribution as a standalone.

Results and Discussion

Table 5.2 displays the results. Regarding our results, the first observation coming from M1 and M2 models is that the shallower model performed the same or worse in almost every aspect. This was expected as less parameters are equivalent to a more restricted expressibility. However, the decrease in parameters and thus model size is great compared to the small decrease in performance, which leads us to the conclusion that the performance is affected more by the channels and dimensionality of the latent representations and less by the depth and the sheer number of parameters. In this case, the shallower model has an equal number of channels to the deeper one at the bottleneck, due to the increased channel step F_c, while having more samples to process, due to the fewer downsampling operations.

The addition of the recurrent layer comes with a great increase in number of trainable parameters and training time. As far as performance in concerned, judging by M1 and M3 models, LSTM provides a boost in all metrics except vocal SIR and accompaniment SAR. The better SDR score can be associated with the better modeling and processing of sequential data by the LSTM. For M4 we wanted to check whether the LSTM can deal with longer sequences coming from a shallower model that also foregoes the larger input context used in M1-M3. The fact that 3 metrics are better than M3 and 5 are better or almost equal than M1 hints that LSTM might in fact benefit from longer sequences of less abstract features. However, we presume that there must be a turning point, where a further increase of the sequence length will hinder the LSTM's performance, as it is not

		M1	M2	M3	M4	M5	M6	M7	M8
	SDR	4.48	4.43	4.77	4.60	4.78	4.52	**5.30**	5.09
Voc.	SIR	**12.35**	10.53	12.07	11.62	10.93	10.87	11.81	11.90
	SAR	5.28	5.29	5.61	5.74	5.70	5.57	5.96	**6.01**
	SDR	9.99	10.11	10.10	10.19	10.15	10.15	**10.84**	10.81
Acc.	SIR	13.80	13.77	14.37	13.78	13.95	13.95	**15.53**	14.83
	SAR	13.15	13.02	12.91	13.30	12.98	13.10	13.18	**13.50**

Table 5.2: Results for M1-M8 models, in terms of vocal and accompaniment SDR, SIR and SAR. The values in bold are the top performing among all models.

made to handle extremely long sequences, and, combined with a slight reduction in performance due to the shallower depth, will worsen the overall separation capabilities of the model.

Regarding the DWT, M7 scores the best performance among all compared and trained models in 3 of the metrics, including the vocal SDR. At the same time, it outperforms M5 in every aspect, with only a relatively small increase in parameters. This comes in accordance with the results of the original paper and clearly shows that the separation of features in high and low frequency ones, along with the avoidance of information loss, thanks to the high-frequency component of the DWT, provide the rest of the model with meaningful information for the separation task. The addition of LSTM and the decrease in depth that occurs in M8, worsens the performance in 3 metrics, including the vocal SDR, but improve it in the rest, indicating that LSTM can benefit from the enhanced features provided by the DWT-inspired pooling layer.

Finally, M5 with the difference output layer has substantially better performance than the simple M1 model, pointing out that it is beneficial to enforce the additive property, instead of leaving the model to learn it by itself. However, M6 has worse overall performance than both M5 and M3, contradicting the previous point. We can't really explain this, but we presume that either the results the modifications are inconsistent, or that there might be a difficulty of instrument processing by the LSTM, as in the M6 case we train based only on the loss coming from the accompaniment.

To conclude, if there are no memory constraints, we find no reason not to include bigger input context to Wave-U-Net, especially since the impact of padding is significant. DWT is a technique that seems very promising for tasks and architectures that involve multi-resolution analysis of a signal. The transformation fits the Wave-U-Net perfectly, by improving its performance and appears to be modular enough to connect with other modifications. Therefore we think that it should be included in any similar variant. The difference output layer might be functional, but the fact that it is based on the unconfirmed assumption that sources are mixed in an additive manner along with the inconsistent results leads us to be very reserved about its efficiency. Finally, as far as recurrent layers are concerned, they might be a useful asset to Wave-U-Net architectures, although the results we got did not point towards a definitive performance improvement. Nevertheless, we believe that the extra parameters and training overhead constitute a handicap and therefore, any potential performance boost is is not worth it.

Chapter 6

Conv-TasNet

Contents

This chapter has the goal of a) presenting Conv-TasNet, a waveform based architecture that was proposed in [37] as a solution for the speech separation task, but has been adapted for music separation by [14], b) experiment with several novel extensions and c) introduce a band splitting method that significantly outperforms the baseline.

6.1 Baseline Architecture

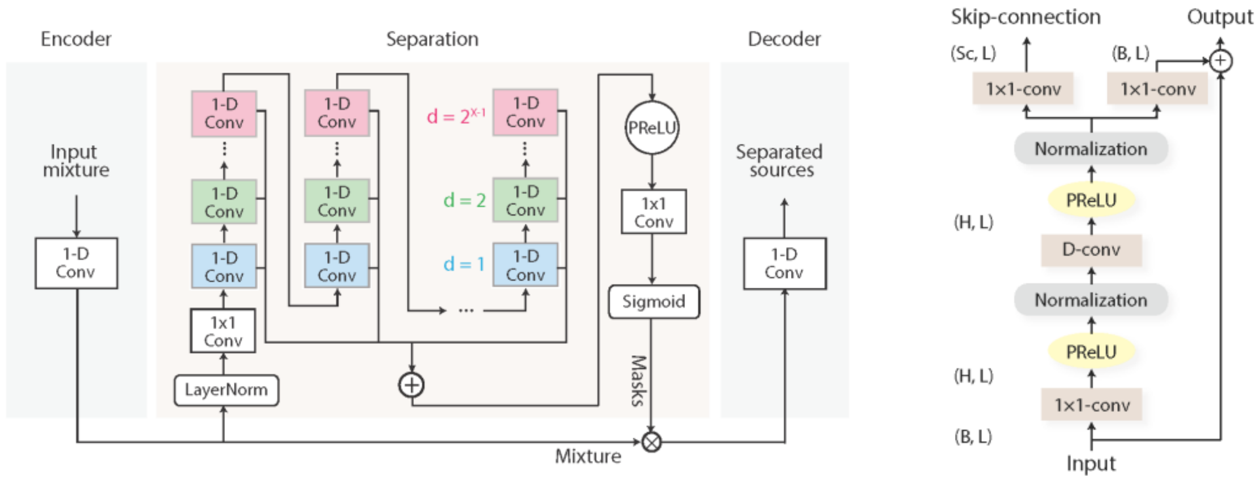
Conv-TasNet is a model that processes audio mixtures in the time domain. The model separates the source signals by predicting and applying masks on their latent representations, a concept that is very popular with spectrogram-based techniques, but wasn't tried out before in time-domain architectures.

The Conv-TasNet architecture consists of three processing stages: an encoder, a separation module and a decoder. In a high level approach, the model works as following:

- The encoder transforms the input signal to an $N \times T$ "time-frequency" representation, that is suitable for the separation task.
- The separator processes the input and tries to extract information to find a weighting function for each source. This function is applied multiplicatively to the encoded representation, resembling a masking operation.
- The decoder reconstructs the source waveforms by transforming each signal back to the original representation.

In more detail, the encoder transforms overlapping segments of the input mixture into a latent, high-dimensional representation, by applying 1D strided convolutions with a relatively large kernel size L. The stride is half the kernel size, in order to create a 50% overlap between the consecutive segments, a value that is common in sound processing techniques. The convolution applies multiple filters, increasing the number of channels from the original A to N, generating a multi-channeled feature map. This feature map resembles structurally a spectrogram, but because it is learnable, it is possible to create real-valued representations of the signal that are much more suitable for the separation process than the STFT or other fixed transformations. Also, information about the signal phase, that is omitted by many STFT techniques, is included in the latent representation. The output of the encoder is passed through a ReLU function to ensure that the representation is non-negative. Similar to NMF, this constraint is imposed to create perceptually meaningful representations, by additively combining a set of basis functions. Before the separator, the feature map is normalized in both channel and time dimensions to speed up training and passes through an 1x1 convolutional layer that changes the channels from the encoding dimension N to the bottleneck dimension B.

The separation module utilizes a multi-block, residual temporal convolutional network (TCN) [33], that uses serially connected stacks of R sub-modules. Each sub-module consists of X depthwise separable convolutional blocks, each with an increasing dilation factor, $d_i = \{1, 2, .., 2^{X-1}\}$. The multiple dilation factors enable the sub-modules to capture data patterns in multiple scales, because the individual blocks work as filters with different, gradually increasing receptive fields. Each convolutional block transforms the latent representation along the channel dimension from the bottleneck dimension, B to an internal hidden channel dimension, H, in order to perform the depthwise separable convolution. Before and after the convolution, the respective feature maps are, again,



(a) (b)

Figure 6.1.1: (a) Flowchart of the Conv-TasNet architecture. (b) The design of the 1-D convolutional block used at the separator's TCN. [37]

normalized. The above procedure produces a mask estimate, which is then used to form the two outputs. The mask follows two distinct paths, in both of which there is an 1x1 convolutional layer that transforms the channels number back to B, resulting in two representations. The first one is fed through a skip connection to the outside of the module, while the second one, summed with the input of the block, which is provided by a residual connection, makes up the input of the next block. These residual connections essentially stack the dilation factor of each block and sub-module, making the whole block operate as a single filter with a huge receptive field that is capable of discovering global dependencies of the signal, apart from local ones. The individual mask estimates coming from the skip connections are summed together in order to produce the overall multiplicative mask, which is then passed through a PReLU, an 1x1 convolutional layer that changes the channels number from B to $C \cdot N$, where C is the number of sources, and a sigmoid, resulting in the final mask matrix. The convolutions of the separator are zero-padded to avoid feature map shrinkage. Contrary to the Wave-U-Net, both the input mixture segment and the feature map size of the latent representation are large enough, so that the effect of padding is less profound.

The mask that is provided by the separator is split in C masks, matching the number of sources that are then applied on the encoded representation, resulting in multiple signals in the latent space. Finally, the source signals are reconstructed by the decoder using strided transposed 1D convolutions with the same stride and kernel size as the encoder to transform the latent signals back to the original space.

Formally, given an input mixture $\mathbf{s} = \sum_{i=1}^{C} \mathbf{x_i} \in \mathbb{R}^{A \times T'}$, where $\mathbf{x_i}$ is a source signal, C is the total number of source signals, T' is the segment length in samples and A is the channels of the input (1 for mono, 2 for stereo etc): the encoder transforms it into a latent representation $\mathbf{w} = \text{encoder}(\mathbf{s}) \in \mathbb{R}^{N \times T}$, where N is the encoding dimension and T is the feature map size of the latent representation. The separator generates the masks as $\mathbf{M} = \text{separator}(\mathbf{w}) \in \mathbb{R}^{C \times N \times T}$, where each matrix along the first dimension is an instrument-specific mask. Finally, the decoder transforms the masked representations back to the initial domain, so as to get the estimated source signals, as $\mathbf{s_i}' = \text{decoder}(\mathbf{w} \odot \mathbf{m_i}) \in \mathbb{R}^{A \times T'}$.

In the [14] variant of Conv-TasNet, which is the one we used the most in our experiments, there are no skip-connections. Instead, the separator outputs the residual path representation of the last convolutional block as the estimated mask matrix. Moreover, the decoder, instead of using an 1D transposed convolutional layer to change both the channel and time dimensions, uses a linear transformation to change the number of channels from N to $A \cdot L$ and then reconstructs the signal using an overlap-add method, that restores the dimensionality of the original signal, by taking into consideration the overlap between successive segments.

6.2 Modifications

As we saw in chapter 4, Conv-TasNet has been the center of intensive research, with many techniques and models been inspired by it. We experimented a lot with several novel extensions, focusing both on the encoder-decoder and the separator modules.

6.2.1 Better Encoders

The encoder of the Conv-TasNet has a major role in the architecture, because the whole separation process is performed on the latent space that it provides. So, changing and improving the encoder can have an huge positive impact on the performance of the model.

Discrete Wavelet Transformation

Influenced by the results we got from our experimentation on Wave-U-Net, we wanted to investigate whether this transformation can be used in Conv-TasNet. Our idea is based on the assumption that the search for a good latent space by the encoder can benefit from the features that DWT provides, as they are information-rich and split based on their frequency in low and high frequency ones.

The transformation can't be used on its own, as it doesn't have a way to explicitly set the representation's channels to a desired number, which is crucial in creating a high-dimensional representation. So, we use it in conjuction with the existing convolutional encoder. The DWT block can be placed either before or after the encoder and the opposite for the decoder, operating on the original or latent space respectively. Taking into consideration that the DWT module alters both the channel and feature map dimensions, which are doubled and halved respectively, we change the encoder's parameters for kernel size and output channels accordingly, to keep the rest of the model the same. More specifically, regardless of the transformation's position, by halving the layer's output channels and doubling the kernel size, the combined encoder has the same characteristics as before, and can be used without further alteration on the separator's side.

Stronger Encoder

This improvement was proposed in [56] and is actually a more complex encoder, capable of capturing more features from the input signals. It does so by combining two sets of features, coming from an array of convolutional layers and from an STFT magnitude spectrogram.

Regarding the implementation, for the first set of features, instead of using a single convolutional layer, the encoder incorporates K such layers connected in parallel, with different kernel sizes, to capture features with a wider frequency range, which are then concatenated and passed through a ReLU activation. The k-th layer has a kernel size equal to $\frac{1}{2^k}$ of the original and a channel dimension equal to $\frac{2^k}{2^K}$ of the original. For the second set of features, the encoder calculates features coming from an STFT spectrogram, that are then normalized and passed through a linear transformation, implemented with a fully-connected layer. The concatenation of these two types of features is processed by two 1D convolutional layers, separated by a ReLU, that correct the channels to match those of the original encoder, that is N. Hence, this encoder can replace the original one without any further change for the rest of the model components.

The decoder works in a similar fashion, transforming the latent representations with a convolutional layer and a ReLU activation and then splitting it in K parts and passing them through the same number of convolutional layers with the same kernel sizes as the encoder, before summing them to produce the estimated source signals.

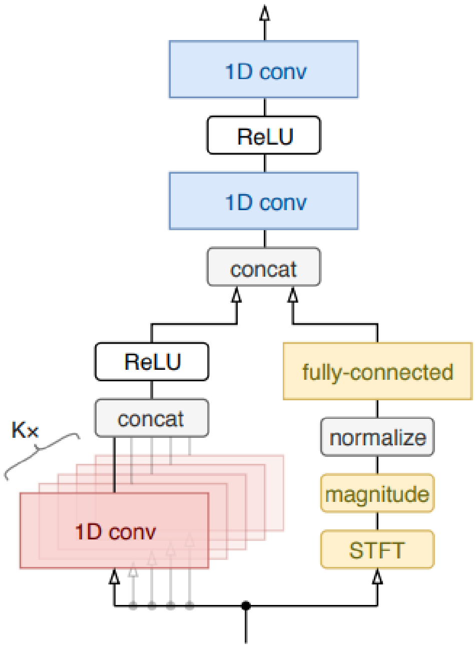


Figure 6.2.1: Design of the stronger encoder module [56].

Encoder with Gammatone Filterbank

In [15] a fixed filterbank originating from the field of DSP, the multi-phase gammatone filterbank (MP-GTF) was proposed as a replacement to the original learned encoder, for the task of speech separation task. We expect this filterbank to be beneficial for the task of singing voice separation as well.

The MP-GTF is based on the auditory gammatone filterbank (A-GTF) [46], a filterbank that tries to model the basilar membrane motion in the human auditory system. This filterbank contains a number of non-linearly spaced narrow-band filters with an increasing bandwidth over the filter's center frequency. The impulse response of each filter is defined as

$$\gamma(t) = \alpha t^{(p-1)} e^{-2\pi bt} \cos(2\pi f_c t + \phi)$$

where f_c is the center frequency, ϕ the phase shift, α the amplitude, $t > 0$ the time in seconds, p the filter order and b the filter bandwidth parameter. The center frequencies f_c are placed in equally spaced positions on the so called ERB scale

$$\text{ERB}_{\text{scale}}(f_{Hz}) = 9.265 \log\left(1 + \frac{f_{Hz}}{24.7 \times 9.265}\right)$$

This scale is derived by integrating $\frac{1}{\text{ERB}(f_c)}$ across frequency, where ERB is the equivalent rectangular bandwidth. The equivalent rectangular bandwidth of the filters corresponds to the bandwidth of a rectangular filter with same maximum gain and total energy, and is given by the following formula:

$$\text{ERB}(f_c) = 24.7 + \frac{f_c}{9.265}$$

The MP-GTF uses filters of filter order $p = 2$, whose impulse response is truncated at 2ms and whose bandwidth is given by $b = \text{ERB}(f_c)/1.57$. The spacing between the center frequencies is set to 1 in the ERB scale, meaning that

$$f_{i+1} = \text{ERB}_{\text{scale}}^{-1}(\text{ERB}_{\text{scale}}(f_i) + 1)$$

where i is the filter index.

Also, to satisfy the non-negativity constraint of the separator's input, the filters come in pairs, meaning that the negative of each fiter is included in the filterbank. In terms of phase, a negative filters correspond to a filter with a phase shift of π.

Due to the predefined spacing between the center frequencies, the total number of filters, including the negative ones, is limited to

$$\#\text{filters} = 2 \cdot \#\text{center\_frequencies} = 2 \cdot (\lfloor \text{ERB}_{\text{scale}}(f_{max}) \rfloor - \lfloor \text{ERB}_{\text{scale}}(f_{min}) \rfloor)$$

where f_{max} and f_{min} correspond to the highest and lowest desired center frequency. Typically chose values for these parameters are $f_{max} = f_{\text{Nyquist}}/2$ and $f_{min} = 100\text{Hz}$. This number might be low (only 48 filters for the 100-4000Hz frequency range) for the application, so the authors of the paper suggest including filters of the same center frequency, but with shifted phases. As only half the filters can be picked freely, due to the filters coming in pairs, the phase shifts are picked so they are equidistantly distributed on the interval $[0, \pi)$.

Figure 6.2.2 displays the magnitude of the filterbank that we used in our re-implementation. Note the non-linear distribution of filters in frequencies; there are many filters at low frequencies and less at high frequencies.

6.2.2 Multi-Conv-TasNet

As the name of this section suggests, we created new structures by combining multiple Conv-TasNets. The premise behind this modification is that the two networks will specialize in different tasks, or different parts of the signals, increasing the overall performance.

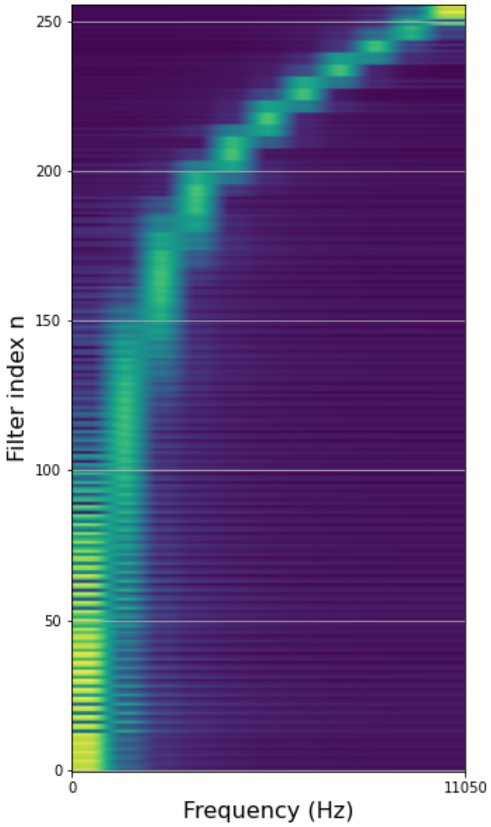


Figure 6.2.2: Frequency domain representation of the gammatone filterbank. We use 256 filters in total, with 48 distinct center frequencies.

Sequential connection

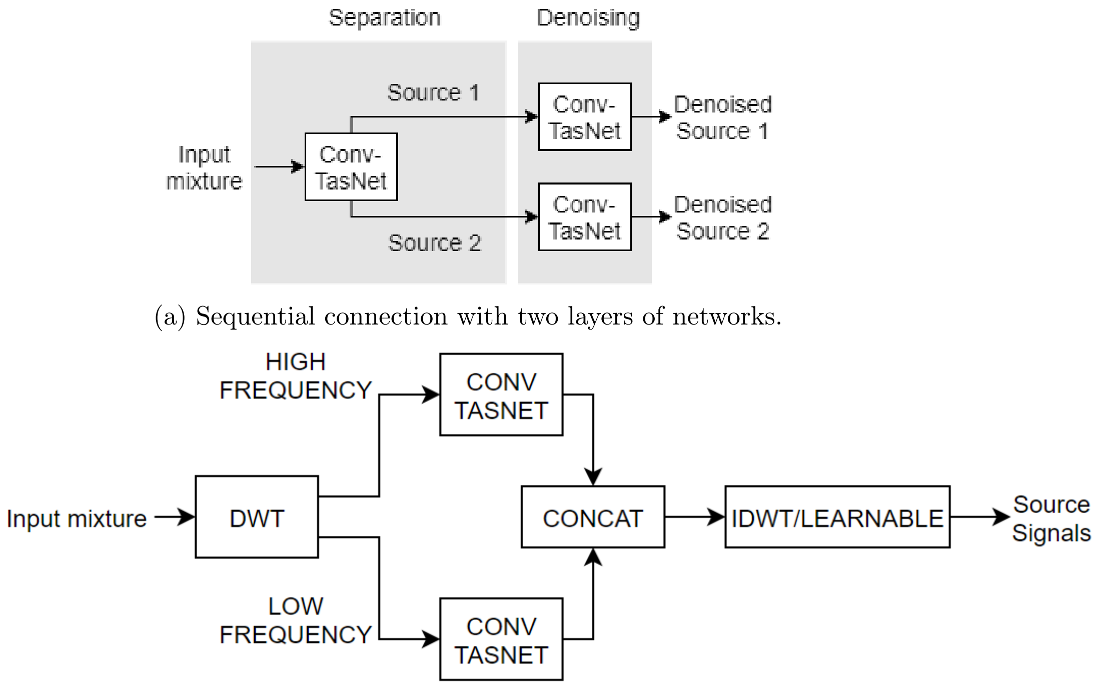
For the sequential variant, we use three Conv-TasNets forming a pyramid-like structure. More specifically, we incorporate 2 layers of Conv-TasNets, connected serially. The first layer has 1 network, that performs an initial separation of the mixture signal to the two source components. Then, each signal is fed to a distinct network to be processed a second time. In this second layer of networks, each network generates one mask and thus outputs one signal. This means that the signals are not further separated, but rather they are processed.

Since the first layer is left unaltered, we expect it to perform the separation task, as before. For the second layer, we expect that its processing will operate as a denoiser, cleansing the signals from annoying artifacts, and hence improving their quality.

Parallel connection

For this variant, we want to connect multiple networks at the same level, in parallel. One way is to create an ensemble, that is a combination of multiple independently trained networks. In that case, the networks would share the input signal, process it in isolation and and then group the results with an aggregation function, such as averaging.

Although ensembles are effective, improving performance, as in [57], we opted for a more sophisticated integration. Therefore, we used DWT at the input, before passing it to the networks, to split



(a) Sequential connection with two layers of networks.

(b) Parallel connection with DWT separating the features in high and low frequency.

Figure 6.2.3: Block diagrams of the proposed Multi-Conv-TasNet modifications.

the signal in high and low frequency components. These components are processed separately by the two networks, resulting in a network specializing in high frequency features and the other in low frequency ones. Finally, the separated signals are concatenated on the channel dimension and passed through an IDWT or learnable transformation to produce the source signals.

6.2.3 Multi-band Separation

The multi-band extension for Conv-TasNet was inspired by MMDenseLSTM [63], a time-frequency domain model that yielded very competitive results in singing voice separation. This model splits the spectrogram representation in multiple frequency bands and processes each band individually, before combining the outputs. We try to create an as similar as possible structure, considering that the domain of operation differs (waveform instead of time-frequency) and that the representation we use is derived from a learnable transformation, while the STFT is fixed, with known properties and interpretation. We expect the separators to specialize in the specific band they are processing and that they will force the encoder to learn better representations to facilitate the separation procedure.

Parallel Multi-band Separation

In order to create multiple frequency bands, as in [63], we approached the latent representation in a way similar to a time-frequency one. More specifically, regardless of the number of channels and the length of the input signal, the encoder produces a multi-channeled representation of 1D series of features. This representation can be thought of having a “latent frequency” (channel) and a “latent time” (feature map) dimension, resulting in a “latent spectrogram”.

Having made the above semantic remark we proceed to the modification in the model's architecture. The encoded latent spectrogram is split along the channel dimension to create Q latent frequency bands $\mathbf{w_i} \in \mathbb{R}^{B_i \times L'}, i = 1 \dots Q$, where B_i corresponds to the number of channels that are assigned to each band. We note that channels are not necessarily assigned to bands in a mutually exclusive basis, meaning that each channel can be included in more than one bands. These bands are then fed to Q individual separator components that work in parallel, and each separator generates its respective sub-masks. These masks are concatenated along the channel axis and, in order to restore the channel dimension of the encoded latent representation, are then processed by a fully-connected layer. Afterwards, the rest of the network proceeds as normal. In the case where the relationship between channels and bands becomes mutually exclusive (each channel is assigned to one band, and is thus processed by one separator), we omit the fully-connected layer, as there is no need to correct the channel dimension. Unless stated otherwise, the bottleneck size is adapted to match the number of channels B_i that is assigned to each band every time. The proposed modification is displayed in Figure 6.2.4

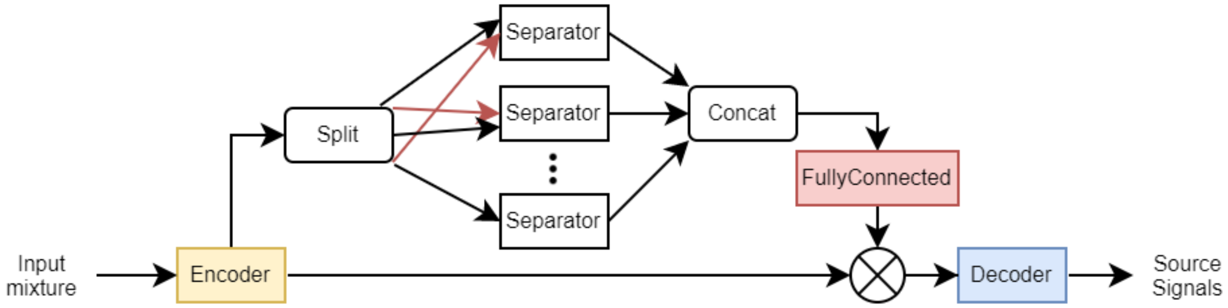


Figure 6.2.4: Block diagram of the parallel multi-band extension. The red elements are used only if the assignment of channels to bands isn't 1-1. Furthermore, the red arrows indicate that a channel can be fed to two separators.

Dilation-split Separation

As in the Multi-Conv-TasNet extension, we can also connect the specialized separator modules in series. In this case, our proposed variant doesn't operate by splitting at the channel dimension into latent frequency bands. Instead it creates "receptive field" bands, by splitting the TCN in multiple separator components, based on the dilation factor of the contained convolutional blocks. All separators output one mask that is used as input by the next separator, with the exception of the last one, that outputs as many masks as the number of sources. This model is depicted in Figure 6.2.5.

Implementation-wise, we propose the segmentation of the separator TCN module into a cascade of Q serially connected separators, each retaining the original dimension of 3 stacks, but including only $D' = D/Q$ dilated convolutional blocks, with ascending dilation factors $d_{ik} = 2^{D'(k-1)}\{1, ..., 2^{D'-1}\}$, $k = 1, ..., Q$. Thus, by manually tuning the receptive field of these separators using the dilation factors of their convolutional blocks, we expect each separator to specialize on extracting more local or global patterns, thus leading to improvements in the performance of the network.

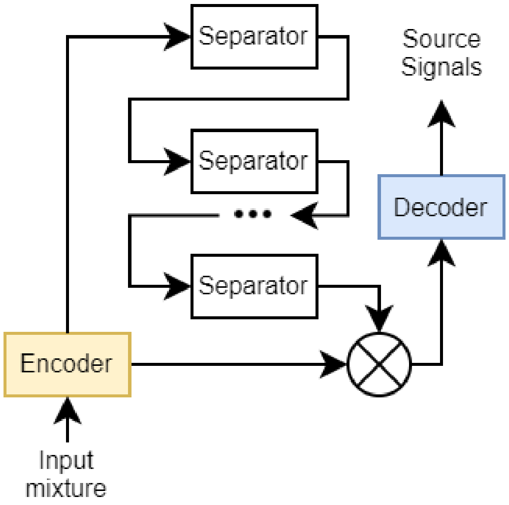


Figure 6.2.5: Block diagram of the dilation split model. Each separator has a fraction of the total dilation factors. Each separator but the last output only one mask.

6.3 Experiments

Experimental Setup

Our research was done in three parts; The first part covers some exploratory experiments to investigate the effectiveness of the multi-ConvTasNet, multi-band separation and DWT encoder modifications. The second part extends the research on multi-band separation, regarding hyperparameters and its scalability. In the third part, we tested the modularity of the parallel multi-band separation, which yielded the best results among those tested in the second part, by combining it with the stronger and the gammatone encoder [56, 15].

For the experimental setup, we used the MUSDB18 dataset with the same properties as in Chapter 5, but using a 86-14 train-validation split. Regarding the data augmentation, we perform normalization on the songs, before processing them, so an amplification would be pointless. Additionally, we follow the augmentation procedure introduced in [67]. That is, we flip the signal's sign, which is equivalent to a phase-shift of π and flip the stereo channels at random. Also, we use a random shift of up to 2 seconds to each stem individually and a remixing of stems between batches. All these transformations result in a generation of a unique training set at each epoch, which can improve the training process and avoid overfitting.

All models were trained with segments of 2 seconds in length, that is 44100 samples and separated the mixture in two components, vocals and accompaniment, as dictated by the singing voice separation task. The optimizer used is Adam [30], with a learning rate of of 0.003. Unless stated otherwise, the loss function used was the L1 loss and the hyperparameters of the Conv-TasNet are as in Table 6.1. The evaluation protocol used is the same as in the experiments of Chapter 5, that is the median-of-medians protocol, devised in [62].

Complementary to the evaluation results and in order to assess their validity over the whole distributions of the metric values, we performed the paired Wilcoxon signed rank test, over all the reported metrics between each model we developed and its respective baseline. The values that indicate a statistically significant improvement over the baseline are denoted with a highlight in the respective tables.

6.3.1 Exploratory Experiments

In this part we trained many models for a small number of epochs to check whether the modified versions offer any noticeable improvements or not. In Table 6.2 we display 8 such models, that were trained for 50 epochs, using early stopping when a model hasn't reduced its validation loss for 20 epochs.

- A1 is our baseline, which as we previously mentioned, is the Conv-TasNet variant of [14], which has no skip connections and its decoder uses a linear transformation and overlap-add method.
- A2 is a Multi-Conv-TasNet in a sequential connection, containing 3 networks in the 2 layer pyramid structure, as described before.
- A3 is a Multi-Conv-TasNet with 2 networks connected in parallel and with DWT and IDWT transforming the input and output, respectively.
- A4 is similar to A3, but with a learnable 1D transposed convolutional layer performing the inverse transformation.
- A5 is similar to A3, with the transformation window of the Conv-TasNet encoders and decoders halved, $L = 10$, to compensate for the resolution loss caused by the DWT.
- A6 is a dilation-split model, with two separators, each containing half the dilation factors of the original, in an ascending order.
- A7 is a parallel multi-band model, with $Q = 2$ bands, where each channel is uniquely assigned to a band.
- A8 is like A7, but incorporated DWT at the encoder in order to perform band-splitting, where the transformation is performed on the latent space. Fig. 6.3.1 depicts the architecture.

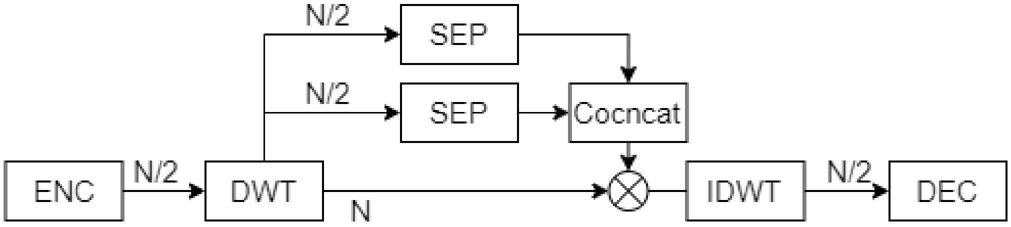


Figure 6.3.1: Block diagram of the multi band extension combined with a DWT in latent space.

Discussion

First of all, it must be noted that as the Multi-ConvTasNet models (A2 to A5) have significantly more parameters than the baseline, 50 epochs of training may not be as efficient as on the other models, and as such, a direct comparison to the other ones might be unfair.

Symbol	Description	Value
N	Channel dimension of latent space	256
L	Transformation window length	20
B	Number of channels in bottleneck	256
H	Hidden dimension	512
P	Kernel size for non 1x1 convolutional layers	3
X	Number of convolutional blocks in each sub-module	8
R	Number of sub-modules in a TCN	3

Table 6.1: Values of the Conv-TasNet hyperparameters used.

Model	Description	#params
A1	Baseline	6.55
A2	Sequential Multi-Conv-TasNet + DWT/IDWT	19.64
A3	Parallel Multi-Conv-TasNet + DWT/IDWT	13.09
A4	Parallel Multi-Conv-TasNet + DWT/Learnable	13.09
A5	Parallel Multi-Conv-TasNet, $P = 10$ + DWT/IDWT	13.07
A6	Dilation-split	6.61
A7	Multi-Band, $Q = 2$	6.47
A8	Multi-Band, $Q = 2$ + DWT/IDWT	6.48

Table 6.2: Description and number of trainable parameters for the models of the first phase of experiments.

Starting off commenting on the results with A2, we can clearly see that the multiple networks structured in such a way do not provide any improvement to the model, as all metrics score below the original baseline. One reason might be that due to the model's high depth and complexity, our expectation of having two layers of networks specialized in different tasks (1st layer separation, 2nd layer denoising) might not be met, as there isn't any training bias towards that behaviour. As such, the model finds its own way of solving the task, which may lead to the creation of obscure latent representations and a high number of artifacts as indicated by the low SAR value. A possible solution to this issue could be the inclusion of deep supervision during training. This means that instead of computing only one loss function at the output of the network, we compute multiple loss functions, in various positions of the architecture, thus imposing additional constraints on the training process. However, generating these intermediate goals and picking the right spots to enforce them, in order to properly guide the training process, isn't trivial and as there are successful neural network architectures with different denoisers [16], and further research is required.

Regarding the models with two Conv-TasNets connected in parallel, each model excels in a metric, (accompaniment SAR for A3, vocal SAR for A4, vocal SIR for A5), while having average performance overall. A3 seems to be the best of the three, with only one metric significantly lower than the baseline, A5 the next, with two metrics lower and A4 the last, with three metrics lower.

The main difference between A3 and A4 is that A3 uses IDWT to combine the two reconstructed signals, instead of a learnable transformation. The better all-around performance of the former leads

	A1 [14]	A2	A3	A4	A5	A6	A7	A8
SDR	5.39	4.82	5.40	5.42	5.38	5.03	**5.48**	5.21
Voc. SIR	13.43	13.05	13.36	11.98	**14.04**	12.06	13.78	11.56
SAR	6.39	4.98	6.32	**6.47**	5.72	6.40	6.25	6.24
SDR	11.08	10.59	11.16	11.14	11.11	11.16	**11.39**	10.95
Acc. SIR	**15.53**	13.63	14.00	15.25	13.99	14.63	15.18	15.42
SAR	13.88	13.73	**14.13**	13.71	13.89	13.87	13.96	13.59

Table 6.3: Results of the first phase of experiments, in terms of vocal and accompaniment SDR, SIR and SAR. Values in bold are the highest in each metric. Underlined values denote a statistically significant improvement (p=0.05) over the baseline A1.

us to think that the learnable transformation doesn't learn the IDWT efficiently.

Between A5 and the other two models, the main difference concerns the length of the transformation window, which is halved so that it compensates for the halving in resolution caused by the DWT. This change has a direct effect on the receptive field of the separator since it provides it with more (double the) samples to work with. The right size of the receptive field, that is whether it is better for it to be large or small, is completely up to the application, as both cases have their strong and weak points. For instance, a large receptive field means the separator "sees" a larger part of the original signal in each processing step, meaning that it can discover more global dependencies compared to a smaller receptive field. Although this may be vital for the separation process, improving the overall performance, it might also be disastrous, as the redundant information distracts the separator away from the potentially important local patterns. On the other hand, we believe that regarding the resolution, the things are clearer. A higher resolution, although it increases processing time, gives the separator more samples to work with, without increasing the parameters due to the convolutional layers' property of parameter sharing. Also, it prevents a further downsampling of the feature map and mitigates the effects of padding. Judging by the results, where A3 is slightly better than A5, which in turn is better than A4, we can't justify or contradict the above points and more in-depth research is needed.

Proceeding with the multi-band models, the band-split model A7 excels in two metrics (vocal and accompaniment SDR), while recording a very good performance overall, compared to the baseline. On the contrary, A6 and A8 perform, generally, worse comparatively to the baseline.

Regarding A6, it is clear that the idea of halving the depth of the separators didn't cause the specialization we expected. A possible explanation is that by splitting the dilation factors and damaging the convolution dilation continuity, the separators lose the property of a unified filter, which leads to bad results after the first separator.

The results of A7 justify our goal, as the model excels in both SDR metrics that are considered the most important ones. The two bands seem to contribute in the formation of more specialized separators. At the same time and judging by A8, we can deduce that either DWT is incompatible with this kind of band-splitting method or it only works with representations that contain some kind of temporal continuity, like the waveforms and thus it can't be applied on latent representations. This can be further justified by the performance of A3, where the DWT was placed on the original space, instead of the latent one.

6.3.2 Multi-Band in-depth Experiments

Following the good results of the parallel multi-band model from the previous section, we performed additional experiments. In this part we test the scalability of the technique and what happens if we revoke the constraint of mutually exclusivity between the channels and the bands.

For all experiments, we used the same database and hyperparameter configurations as before, but increased the total amount of epochs from 50 to 150, in order to get more trustworthy results.

We trained the following models:

- B1 is our baseline, A1, trained for 150 epochs.
- B2 shares the same architecture with A7, but is trained for 150 epochs.
- B3 is similar to B2 but with double the bottleneck size, to test its effect on the separation capacity.
- B4 is a band-splitting model with $Q = 4$ bands, with each band consisting of N/Q unique channels. This model was trained to test the scalability of B2.
- B5 is a band-splitting model with $Q = 3$ bands. As in [63], apart from the 2 standard, mutually exclusive bands, we include a third "full band", containing all N channels. These 3 separators produce a representation with a total of $2N$ channels and thus we use a linear transformation to reduce the channels' number to N.

Model	#params	#bands (Q)	Bottleneck Coefficient	Full Band
B1	6.55M	1	×1	×
B2	6.47M	2	×1	×
B3	12.87M	2	×2	×
B4	6.51M	4	×1	×
B5	6.61M	3	×1	✓

Table 6.4: Number of trainable parameters and architectural details of the models of the second phase of experiments. The bottleneck coefficient is used to change the bottleneck size of each separator, which is normally adapted to match the number of channels of each band.

Discussion

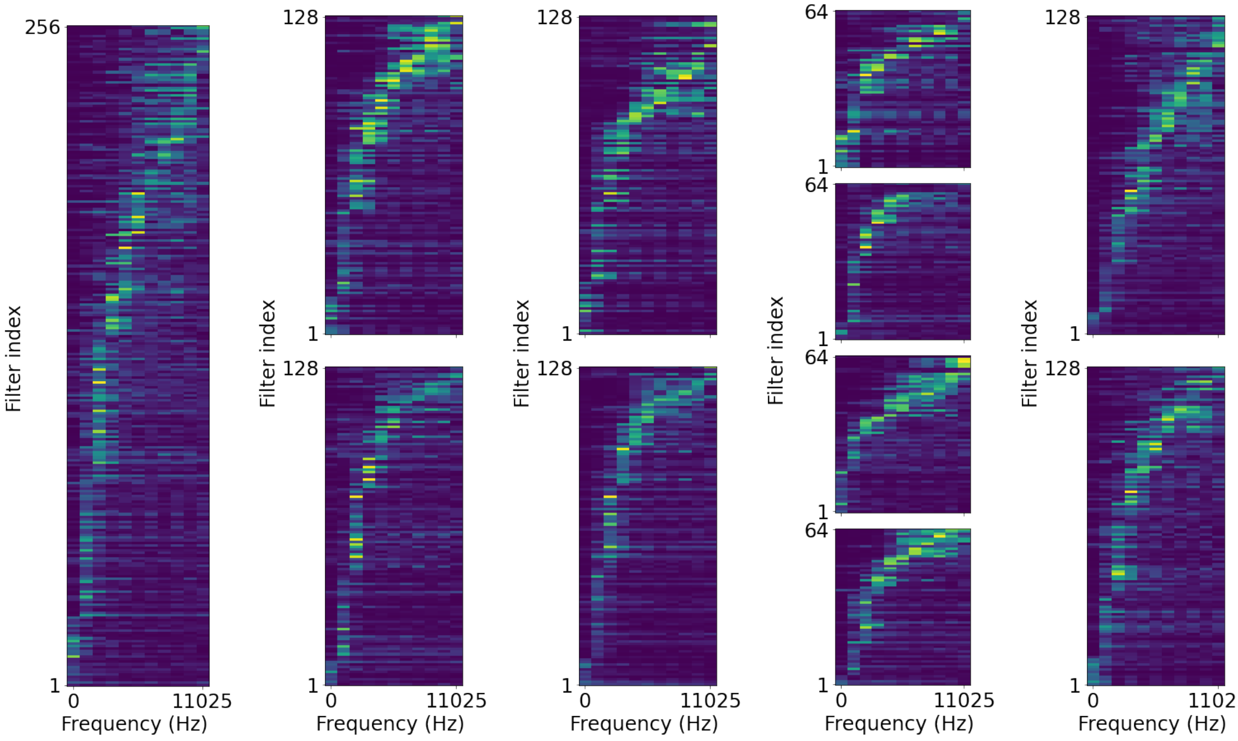
Table 6.4 showcases the number of parameters for each trained model, Table 6.5 shows the evaluation results in terms of SDR, SIR and SAR and Figure 6.3.2 displays the frequency domain representation of filters of the encoder for the B1-B5 models. The filters of the encoder are split based on the separator they belong to. Also, Figure 6.3.3 displays the learning curves of the models.

The two-band models B2 and B3 achieve the best overall performance among latent frequency band models, with B2 scoring better in vocal SDR and B3 in accompaniment SDR. We note that B2 achieves a lower SIR compared to B3, but compensates by recording a higher SAR. The frequency domain representations of the filters the separators receive are too similar between the two models

		B1 [14]	B2	B3	B4	B5
Voc.	SDR	5.81	**6.37**	6.11	6.05	5.94
	SIR	14.13	14.25	**14.69**	14.61	14.23
	SAR	6.59	**7.12**	6.59	6.98	6.78
Acc.	SDR	11.78	12.21	**12.47**	11.66	11.76
	SIR	16.01	16.69	**16.76**	16.04	16.01
	SAR	14.24	**14.52**	14.25	14.10	14.37

Table 6.5: Results of the second phase of experiments, in terms of vocal and accompaniment SDR, SIR and SAR. Values in bold are the highest in each metric. Underline values show statistically significant improvement (p=0.01) over baseline model B1.

and they both indicate that our goal of creating specialized separators has been achieved, as the encoder processes the input of each separator with a set of filters with distinct characteristics. More specifically, as we see in Figure 6.3.2b)-c), one of the separators is assigned more filters with higher central frequencies and the other more filters with lower central frequencies, narrower passbands and lower energy. However, considering that B3 has double the parameters of B2 and the baseline model, we deduce that the increased expressibility coming from the larger bottleneck doesn't translate to better separation capabilities.



(a) B1 (b) B2 (c) B3 (d) B4 (e) B5

Figure 6.3.2: Frequency domain representation of the filters of the encoder. The sub-figures display the sub-space that is processed the respective separator, each time. The filters of each band have been sorted in ascending order of base frequency.

The four-band model, B4, outperforms the baseline model B1, but not the two-band models B2 and B3, recording nevertheless a high vocal SIR. This implies that using more separators to process "narrower" frequency bands has diminishing returns and that the technique might not be scalable in that term. The low performance could be attributed to the lower number of bands being assigned to each separator and to the decrease of the bottleneck factor which is done to keep the amount of parameters the same; as the latent information available to each separator decreases, they struggle to independently converge to good solutions that are based on mutually exclusive information, as the separators are not aware of what the other ones are learning and thus extracting overlapping information, potentially limiting the separation capacity. This is also visible from the frequency representation of the encoder filters; as we can see, although the middle two bands – corresponding to the two "middle" separators – receive distinct filters, the top and bottom bands receive filters that roughly cover the same frequencies.

The model that additionally processes the entire latent representation, B5, recorded significantly lower scores than the 3 previous models in all metrics except SAR, in which it has comparable results. It seems that processing the entire representation in addition to separate bands has a nullifying effect on the specialization of the separators. This is apparent from Figure 6.3.2e, where we can see that while there are some differences between the frequency domain representation of the filters assigned to each separator, the effect is not as profound as in the cases where no separation path with access to the full latent spectrogram exists.

As can be seen at the Figure 6.3.3, the baseline B1, the four band model B4 and the full band model B5 converge faster than the two band models B2 and B3, as they stop due to the used early stopping protocol. Additionally, at the 100 epochs mark, the B5 model achieves its lowest loss value, both in terms of training and validation. However, it seems that both B4 and B5 face some kind of diminishing returns of training, due to overfitting, as the reduction of train loss does not translate to less validation loss at the last epochs, in contrary to models B2 and B3. Of course, it should be noted that for this task, the L1 loss and the evaluation metrics, SDR, SAR and SIR, are not fully correlated, which means that although lower loss values generally lead to higher metric scores, two models with the same loss may have substantially different performance.

6.3.3 Multi-Band Modularity Experiments

We proceed with experiments regarding the modularity of the parallel multi-band modification, by combining the two band model with the stronger encoder and the gammatone filterbank encoder.

For this set of experiments we are more interested in the performance of the modified versions compared to the respective baseline, than in the metrics scores themselves. We trained the following models:

- C1 is the baseline for C models. It is the simple architecture (A1, B1) but instead of using the original, learnable encoder and decoder we use the stronger encoder of [56], with 6 different convolutional layers for the time-domain processing path and 256 features for the STFT processing path.
- C2 is similar to C1, but the joint representation is split into $Q = 2$ bands, with N/Q channels each, essentially merging the models B2 and C1.

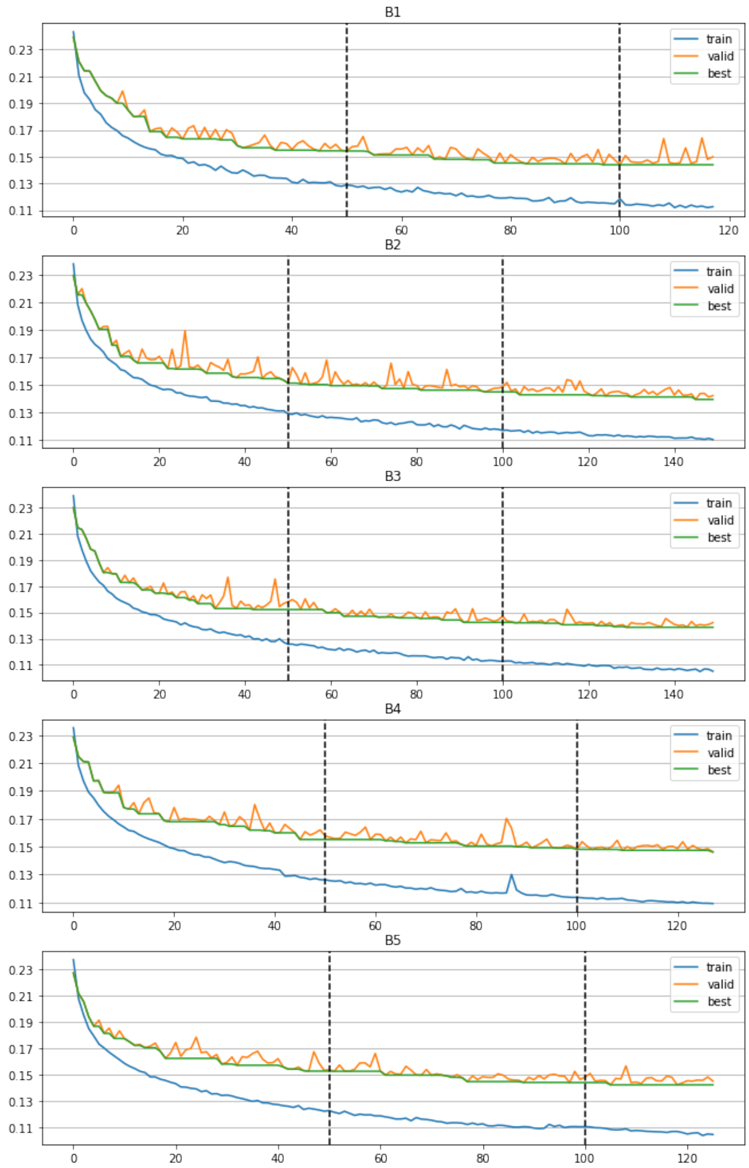


Figure 6.3.3: Graphs of training, validation and best validation loss per epoch for B models.

- C3 is similar to C2, but we modify the stronger encoder to not concatenate the features from the two paths. Instead, the two paths are passed through an 1x1 convolutional layer each to change their channels to N/Q and then are fed directly to the two different separators, one for the features extracted from the signal waveform and one for those corresponding to the STFT magnitude spectrogram.

- D1 is the baseline for D models. It is also based on the vanilla (A1) Conv-TasNet architecture, with the caveat that it utilises a gammatone filterbank as its encoder, as in [15]. Regarding the filterbank's hyperparameters, instead of 24 center frequencies we use 48, which are spaced equidistantly in the ERB scale, but with a smaller step size than 1 unit, and also similar to [55]. Additionally, instead of the linear/overlap-add decoder of [14], that was used in every other model, we use the decoder of the original paper, implemented as an 1D transposed convolutional layer.

- D2 is similar to D1 but with $Q = 2$ bands, containing N/Q unique channels each. Similar to C2, this model merges B2 and D1 models. In this case, the channels are assigned to bands based on their central frequency, in ascending order.

- D3 is similar to D2, but instead of feeding each separator with the first or last N/Q channels, we distribute the channels to the two separators depending on their respective filter phase. One separator gets the channels that correspond to positive filters (phase $\in [0, \pi)$) and the other those that correspond to negative filters (phase $\in [\pi, 2\pi)$).

- D4 is similar to D2 but we employ a linear transformation on the channel dimension on the output of the filterbank before feeding it to the separators. This transformation doesn't change the number of channels, but it enables each separator to linearly combine the channels it processes.

As before, we used the same database and hyperparameter configurations. The C models were trained for 150 epochs, while the D models were trained for additional 100 epochs for a total of 250, with the L2 loss, instead of L1, following the concurrent literature [15, 55].

Model	Description	#params
C1	Stronger Encoder Baseline	7.28M
C2	Stronger Encoder + Multi-Band, $Q = 2$	7.22M
C3	Stronger Encoder + Multi-Band, $Q = 2$ + Split Feature Paths	7.07M
D1	MP-GTF Baseline	6.52M
D2	MP-GTF + Multi-Band, $Q = 2$	6.44M
D3	MP-GTF + Multi-Band, $Q = 2$ + Channel distribution based on phase	6.44M
D4	MP-GTF + Multi-Band, $Q = 2$ + Channel distribution based on linear layer	6.44M

Table 6.6: Number of trainable parameters and architectural details of the models of the third phase of experiments.

	C1 [56]	C2	C3	D1 [55]	D2	D3	D4
SDR	**6.39**	6.36	6.24	5.55	5.31	5.49	**5.69**
Voc. SIR	14.39	**14.92**	13.84	14.96	14.92	14.63	**15.06**
SAR	6.82	7.09	**7.25**	7.28	7.09	7.23	**7.39**
SDR	**12.23**	12.03	11.78	8.06	8.03	7.99	**8.09**
Acc. SIR	**17.57**	17.51	17.08	18.40	18.14	**18.56**	17.77
SAR	14.20	14.07	**14.25**	14.65	14.57	14.64	**14.94**

Table 6.7: Results of the third phase of experiments, in terms of vocal and accompaniment SDR, SIR and SAR. Values in bold are the top performing in each group of models. Underlined values show statistically significant improvement (p=0.01) over the baseline model of each group of models (C1 for C models, D1 for D models).

Discussion

Regarding the C models, we observed that while the C2 model performed comparably to the baseline C1, the C3 model yielded worse results. Hence, we deduce that the features provided by the stronger encoder are incompatible with the band-splitting technique. This may be due to the heterogenous nature of the provided features. What we mean by that is that on the one hand, there are two processing paths that deal with completely different features (waveform vs T-F domain) and on the other hand, the distinct convolutional layers of the waveform path do not result in a unified filterbank, as is the case with the original or the gammatone encoder.

Our explanation about the difference in performance between the multi-band models C2 and C3 is that the success of this stronger encoder lies in the combination of time and T-F domain features. Thus, splitting the representation channels in mutually exclusive bands after merging the time-domain and T-F-domain features, without any interconnections between the two separators, as in the case of C2 model, allows the encoder to suitably combine both its input domains. On the other hand, processing each feature type individually, as in C3 model, greatly restricts the potential benefit that this encoder offers.

As can be seen by the Figure 6.3.4, the baseline model converges faster than the C2 model, which in turn converges faster than the C3 model. The graphs are, also, in accord with the previous points about the performance of these three models, as each model reaches a lower train and validation loss than the next in the sequence. Both of these points constitute further evidence that the multi-band modification does not cooperate well with the stronger encoder. Also, all C models and especially C1 converge at a substantially lower train and validation loss than the B models, supporting the generally better performance that is reported by the results.

Regarding the D models, the first, obvious observation is that all accompaniment SDR metrics are considerably lower than every other model. This most likely has to do with the fact that the gammatone filterbank was designed to model human speech, instead of instrument sounds. Moving on to the individual models, we see that D4 excels in 5 out of 6 metrics (all but accompaniment SIR), D3 excels in one metric but has mediocre performance at the others and D2 has comparatively average to bad performance.

The main difference of the 3 models is the distribution of the frequency ranges that the filterbank covers into bands. Starting of with D3, its separators receive components of the whole frequency range, that are just processed by filters with negative phases. Judging by the results, this doesn't seem to create any kind of specialization at the separators, which leads us to think that in order for the technique to work, a selection of frequency bands is needed. On the far opposite side, D2 distributes them based strictly on frequency range, splitting the gammtone filterbank representation into high and low frequency components. This also seems to hinder the separation capacity, even more severely than in the case of phase-uniform bands hinting that although each separator may specialize in a specific band of frequencies, it still needs to process some information corresponding to other frequency ranges. Finally, D4, with the use of a linear layer seems to enjoy the best of both worlds, as it picks and combines those frequencies that are important to the separation process, achieving extremely good results.

As can be seen by the Figure 6.3.5, all D models converge more or less at the same train and validation loss value, with the D2's values being a bit greater and D4's value a bit lower. However,

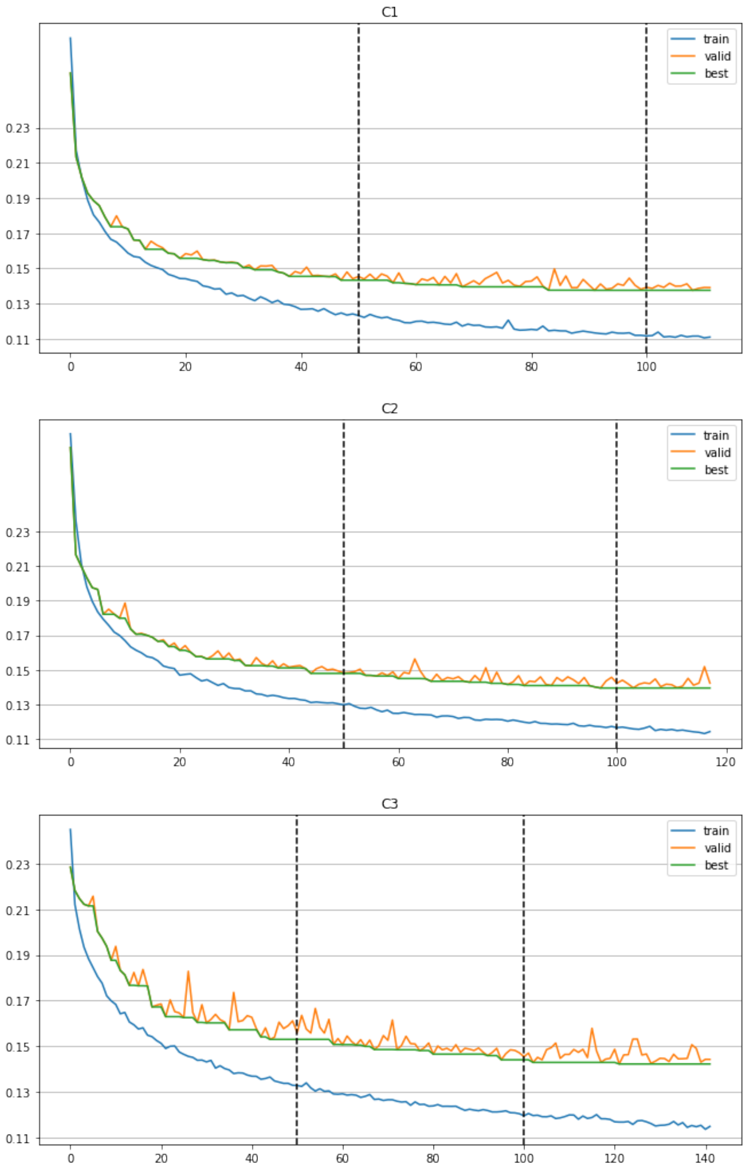


Figure 6.3.4: Graphs of training, validation and best validation loss per epoch for C models.

the rate of convergence differs significantly, with the top performing model, D4, converging at around 150 epochs, that is 100 less than the total training, which could ultimately be taken advantage of to reduce the resources cost.

6.4 Overall Discussion

Figure 6.8 concentrates the results of the baseline and the best model of the two sets of in-depth experiments. Namely, it displays the metrics for the models B1, B2, C1, D4. The C1 and B2 models score the top SDR for both sources, outperforming the baseline B1, in contrary to D4, which scores lower than the baseline, especially on the accompaniment. However, regarding the rest of the metrics, D4 clearly achieves the best performance, by a medium to large margin from the second best model each time, which is either B2 or C1.

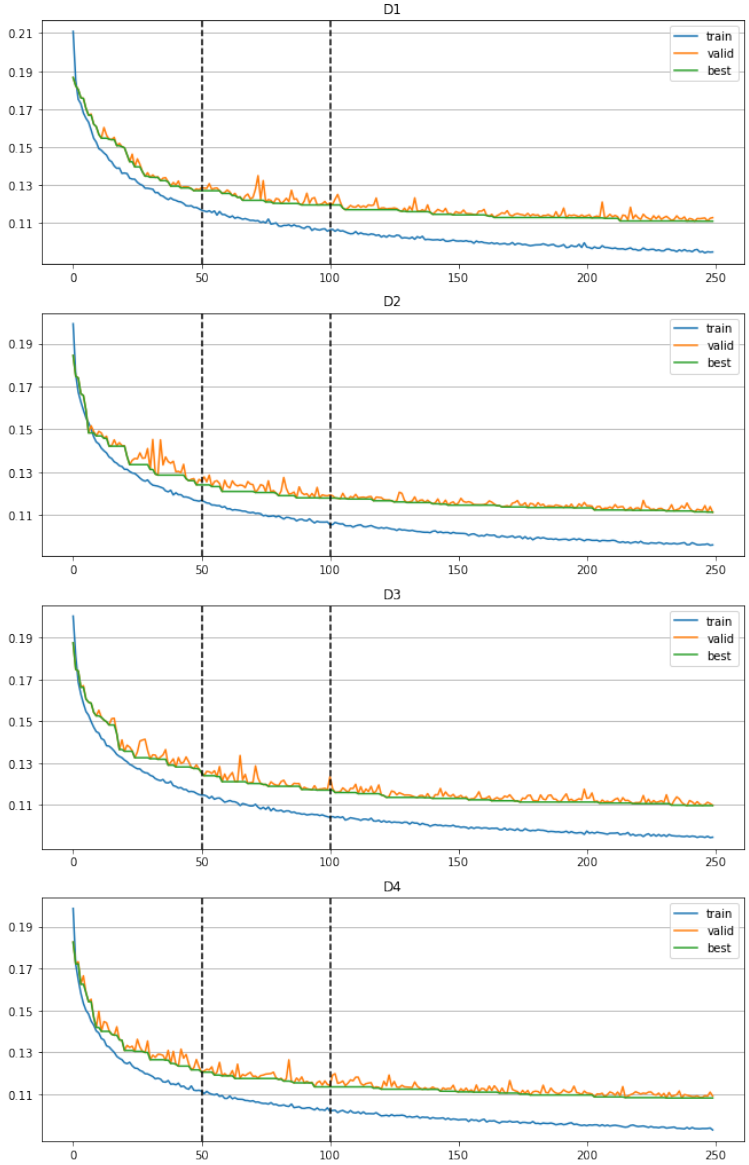


Figure 6.3.5: Graphs of training, validation and best validation loss per epoch for D models.

So, to conclude, the multi-band technique has proven to be beneficial for the baseline architecture, so we believe that it should be considered for addition in Conv-TasNet architectures. Regarding the encoders, the stronger encoder provides a strong performance boost in all metrics and outperforms the MP-GTF encoder in the most important one, SDR, so it is probable that the former one can contribute in creating Conv-TasNet architectures with state-of-the-art performance. However, we think that the MP-GTF encoder can still be a considerable option, as it achieved top performance in the non SDR metrics, it does not depend on training due to its fixed nature and it seems to be flexible enough to be combined with other techniques, such as the multi band modification.

To showcase the effect of the models on the mixture signal, a visual representation with the use of spectrograms is presented below. For the demonstration, two segments of two seconds long are used; the first is from the song "Falcon 69" by the "The Easton Ellises" and the second from "A Place

		B1[14]	B2	C1 [56]	D4 [55]
Voc.	SDR	5.81	6.37	**6.39**	5.69
	SIR	14.13	14.25	14.39	**15.06**
	SAR	6.59	7.12	6.82	**7.39**
Acc.	SDR	11.78	12.21	**12.23**	8.09
	SIR	16.01	16.69	17.57	**17.77**
	SAR	14.24	14.52	14.20	**14.94**

Table 6.8: Aggregated results of the baseline model and the top performing model from each set of models.Values in bold are the best in each metric.

For Us" by the "Carloz Gonzalez", both of which are included in the test split of the MUSDB18 dataset. Figure 6.4.1 and 6.4.2 displays the spectrograms of the ground-truth mixture and vocal component and the estimated vocal signal of top performing models of each category (B2, C1, D4) for the first audio channel of the segment of the first and second song, respectively.

The Table 6.9 includes the metric scores for the vocal component of the models for the whole song and for the particular segment.

	First song			Second song		
	SDR	SAR	SIR	SDR	SAR	SIR
B2	6.10 (2.54)	10.39 (7.04)	7.31 (3.99)	5.57 (4.42)	7.26 (4.45)	13.47 (12.61)
C1	5.89 (2.24)	9.88 (6.41)	7.02 (3.59)	7.29 (4.44)	7.85 (4.50)	16.67 (12.43)
D4	6.94 (3.08)	11.61 (6.31)	9.44 (5.19)	4.82 (3.85)	7.38 (4.90)	17.98 (14.65)

Table 6.9: The segment-wise and total metric (SDR, SAR, SIR) scores, in parenthesis, for the two song segments for the vocal component. The total score is calculated as the median value of all segments.

Regarding the first song, the spectrogram of C1 model is noticeably different than the ground truth. In spite of including the fundamental frequency of the vowel sound, the model has cut off many harmonics, something that is shown by the sparse distribution of horizontal pink lines in higher frequencies. Also, it has erroneously generated a near silent part near the 1/3 mark of the segment, which is displayed by the absence of lines. This wrong estimation has contributed to the worse performance compared to the other two models.

The spectrograms of D4 and B2 are evidently more similar to the ground truth than that of C1. In both spectrograms the fundamental and the overtone frequencies can be observed. Additionally, the intermediate part, just before the second, prolonged sound has been predicted correctly, a bit better for the D4 than the B2 model. The principal difference between the two spectrograms is that the B2 one contains significantly more frequencies, which comes from the accompaniment and appear as "noise" over the frequency dimension than that of D4. This can also be heard in the audio playbacks, with the B2 including a lot more instrumental sounds that the D4.

Regarding the second song, the spectrogram of D4, although it has captured the base frequency of the main sound and the beginning and end of the segment, it has missed the two higher base frequencies along with several overtone frequencies, probably hindering its performance. The spectrogram of B2 generally predicts both the base frequencies and the overtones correctly. However, it misses a major base frequency at the beginning of the segment and prolongs a vocal sound in a silent segment, leading to the mediocre reported SDR. Finally, the spectrogram of C1 seems to be closer to the ground truth, as it seems to capture for the most part the base and overtone frequencies, without missing the beginning and end of the song.

It must be stated that the reason the segment-wise metrics reported here do not necessarily match the results of the previous sections, is that our evaluation protocol, according to which several conclusions were drawn, aggregates the metrics from all segments and songs, which is completely different to judging the models by a single, arbitrary snapshot.

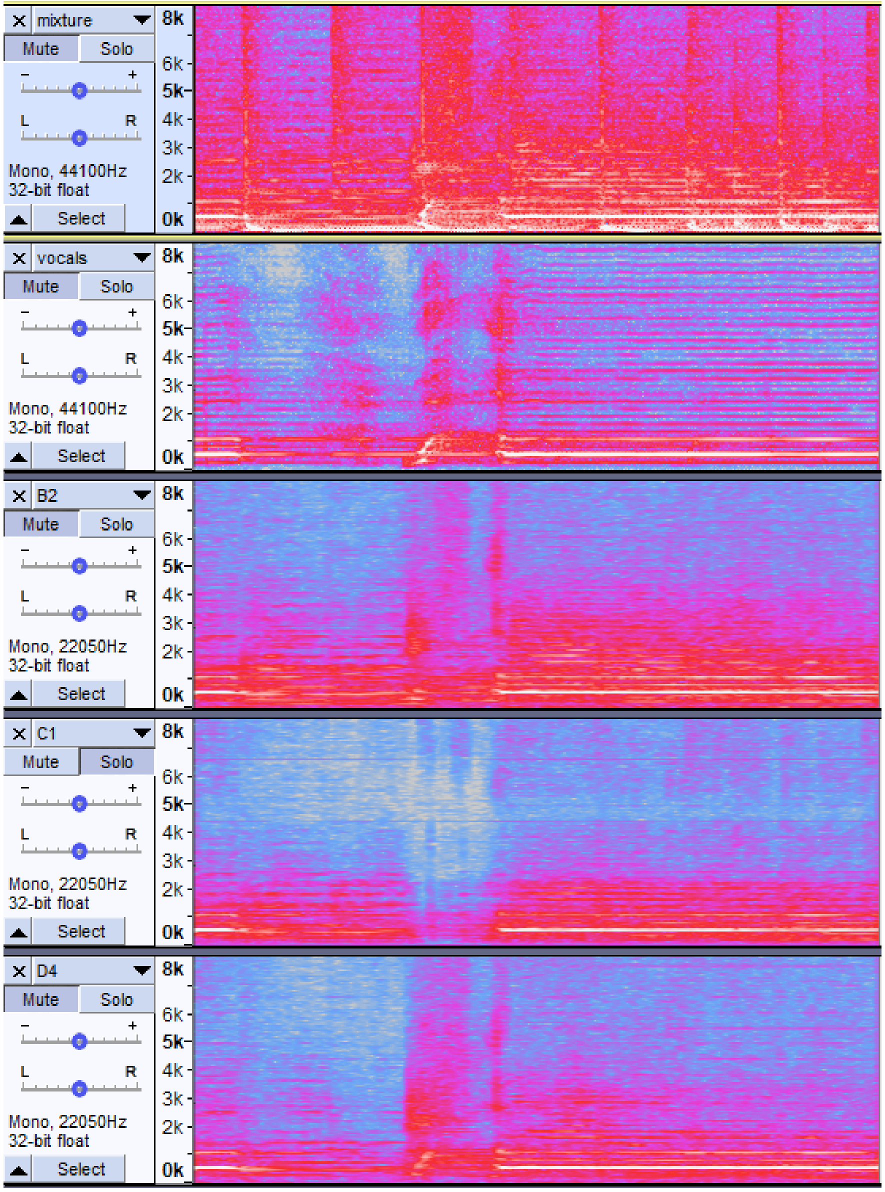


Figure 6.4.1: The spectrograms of the vocal component of the segment of the song "The Easton Ellises - Falcon 69".

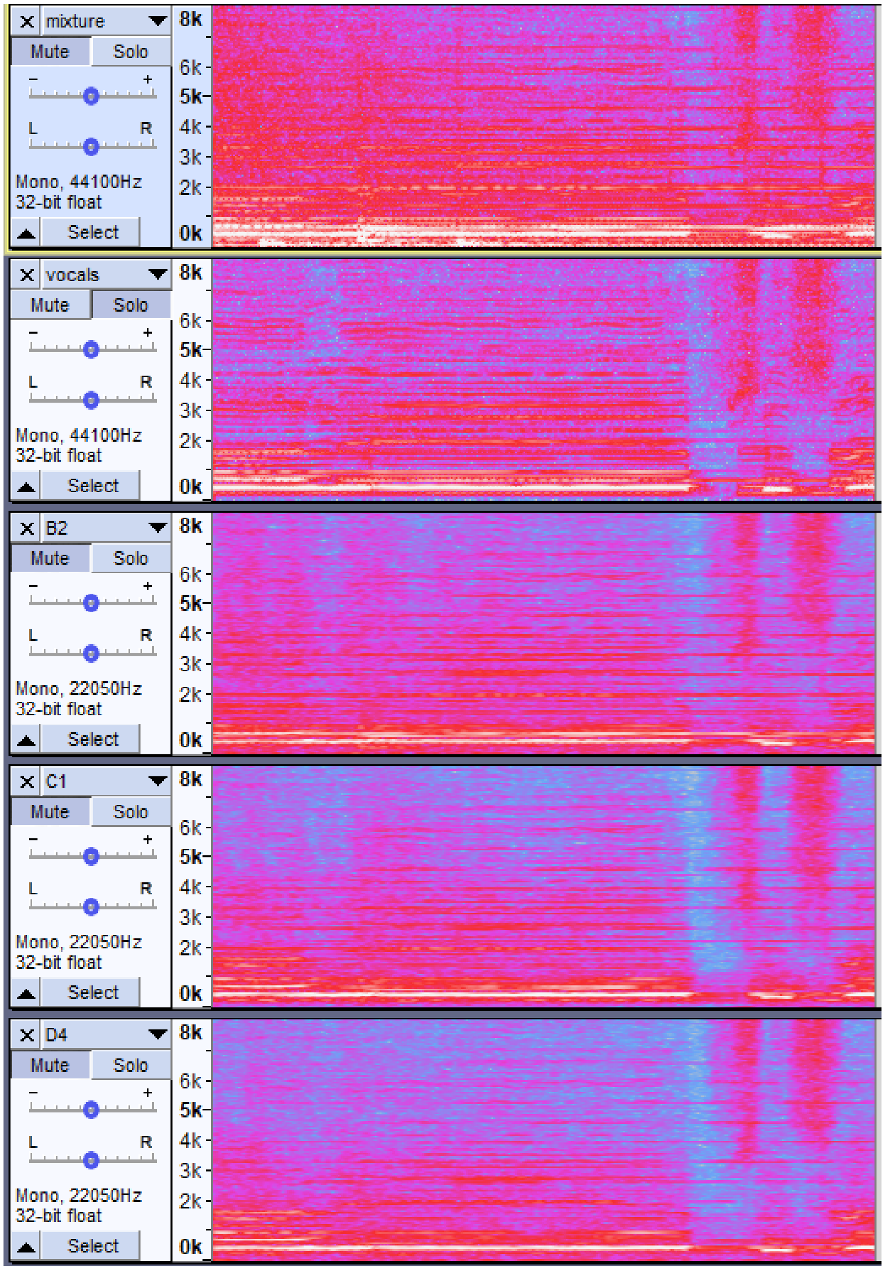


Figure 6.4.2: The spectrograms of the vocal component of the segment of the song "Carloz Gonzalez - A Place For Us".

Chapter 7

Conclusion

7.1 Summary

In this Thesis, we conducted a meticulous study on the problem of Singing Voice Separation and experimented with two popular and successful DNN architectures, namely Wave-U-Net and Conv-TasNet.

Regarding the first architecture, we examined the basic model and tested the effect of several modifications and extensions, proposed on both the original and other studies, on its performance. More specifically, among the modifications proposed in [61], we experimented with the bigger input context for convolutional layers, a technique that was used to get rid of the negative effects that padding has on the architecture, and the difference output layer, which, based on the nature of data, imposes a constraint on the output layer to simplify its training. On the other hand, among extensions proposed in other studies [29, 56], we incorporated an RNN layer on the bottleneck, to examine whether its specialization on processing sequential data could prove beneficial for the model and we replaced the existing processing blocks with ones that apply DWT to the extracted feature maps, in order to tackle the problem of aliasing and information loss. The results showed using a bigger input context, along with the DWT processing blocks do improve the performance of the baseline by a small but noticeable margin. In the contrary, the results of the other two modifications are not clear enough and thus a conclusion cannot be drawn.

Regarding the second architecture, Conv-TasNet, three sets of experiments were constructed. In the first part, we proposed several novel modifications to the baseline, in order to find an effective one to further analyse it. More concretely, we proposed two architectures that incorporated multiple Conv-TasNets, combined in parallel and sequentially, respectively, an architecture that altered the separator module, by splitting it in multiple separators based on the included dilation factors, and an architecture that used multiple separators that process the latent representation independently. All but the last modifications performed the same or worse than the baseline, indicating that they are either inoperative, or they need to be incorporated in another way.

The last modification, multi-band separation, yielded average to good results and thus, in the second part of experiments, we performed an in-depth analysis. The experiments included changing the number of bands and tuning the separators' hyperparameters to check the efficacy and the scalability of the technique. The results indicated that, depending on the hyperparameter configuration, a noticeable amount of specialisation on each separator occurs, which was the initial goal, rendering the technique successful.

In the last part of experiments, we combined the multi-band separation with two different encoders, a stronger encoder from [56] and a fixed, gammatone encoder from [46], in order to examine whether the technique can be generalized with other versions of the baseline architecture. The received results were conflicting, as the effect of the technique was significantly beneficial for the second encoder and negligible for the first one. Our explanation for the root of this discrepancy is the structure of the first encoder, which, unlike that of the second one, differs a lot from the original regarding the nature of the extracted features, since they originate from both temporal and time-frequency domains, instead of just the waveform domain, as in the original.

7.2 Future Work

Concerning the multi-band separation, future research could focus on the nature of the technique, studying the properties of the latent subspace that is created by the encoder and its relationship with the latent frequency bands. Apart from understanding the effect of the technique better, this could open the way for a more fine-grained tuning of its hyperparameters that would be beneficial in terms of the overall performance of the model.

Additionally, an interesting direction of research would be to manually craft the bands for the separators, instead of leaving everything up to training, as it is now. In that way, the bands could be constructed to have certain properties that would facilitate the separation process in general, or adapt it to the needs of a specific task.

Finally, it should be investigated in what extent the proposed technique can be used in other separation frameworks that follow the encoder-separator-decoder architecture. This could open the way for the integration of the technique in other audio separation tasks, such as music source or speech separation.

Chapter 8

Bibliography

[1] URL: https://source-separation.github.io/tutorial/intro/src_sep_101.html.

[2] URL: https://datacated.com/datacated-challenge/the-bias-and-variance-tradeoff.

[3] URL: https://ai.plainenglish.io/the-rise-and-fall-of-the-perceptron-c04ae53ea465.

[4] URL: https://www.d2l.ai/chapter_computer-vision/transposed-conv.html.

[5] URL: https://en.wikipedia.org/wiki/Discrete_wavelet_transform.

[6] Alexandridis, A., Griffin, A., and Mouchtaris, A. "Capturing and Reproducing Spatial Audio Based on a Circular Microphone Array". In: *Journal of Electrical and Computer Engineering* 2013 (Mar. 2013). DOI: 10.1155/2013/718574.

[7] Bashit, A. A. and Valles, D. "A Mel-Filterbank and MFCC-based Neural Network Approach to Train the Houston Toad Call Detection System Design". In: Nov. 2018, pp. 438–443. DOI: 10.1109/IEMCON.2018.8615076.

[8] Bittner, R. et al. "MedleyDB: A multitrack dataset for annotation-intensive MIR research. 15th International Society for Music Information Retrieval Conference, ISMIR 2014". English (US). In: 15th International Society for Music Information Retrieval Conference, ISMIR 2014 ; Conference date: 27-10-2014 Through 31-10-2014. 2014, pp. 155–160.

[9] Bronkhorst, A. "The Cocktail Party Phenomenon: A Review of Research on Speech Intelligibility in Multiple-Talker Conditions". In: *Acta Acustica united with Acustica* 86 (Jan. 2000), pp. 117–128.

[10] Casey, M. and Westner, A. "Separation of Mixed Audio Sources By Independent Subspace Analysis". In: (Jan. 2000).

[11] Chan, T.-S. et al. "Vocal activity informed singing voice separation with the iKala dataset". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 718–722. DOI: 10.1109/ICASSP.2015.7178063.

[12] Chandna, P. et al. "Monoaural Audio Source Separation Using Deep Convolutional Neural Networks". In: *Latent Variable Analysis and Signal Separation*. Ed. by P. Tichavský et al. Cham: Springer International Publishing, 2017, pp. 258–266. ISBN: 978-3-319-53547-0.

[13] Chandna, P. et al. "A Vocoder Based Method For Singing Voice Extraction". In: *CoRR* abs/1903.07554 (2019). arXiv: 1903.07554. URL:

[14] Défossez, A. et al. *Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed.* 2019. arXiv: 1909.01174 [cs.SD].

[15] Ditter, D. and Gerkmann, T. "A Multi-Phase Gammatone Filterbank for Speech Separation Via Tasnet". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (May 2020). DOI: 10.1109/icassp40776.2020.9053602. URL:

[16] Drossos, K. et al. *MaD TwinNet: Masker-Denoiser Architecture with Twin Networks for Monaural Sound Source Separation*. 2018. arXiv: 1802.00300 [cs.SD].

[17] Emiya, V. et al. "Subjective and Objective Quality Assessment of Audio Source Separation". In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.7 (2011), pp. 2046–2057. DOI: 10.1109/TASL.2011.2109381.

[18] Every, M. and Szymanski, J. "Separation of synchronous pitched notes by spectral filtering of harmonics". In: *Audio, Speech, and Language Processing, IEEE Transactions on* 14 (Oct. 2006), pp. 1845–1856. DOI: 10.1109/TSA.2005.858528.

[19] Feng, J. et al. "Reconstruction of porous media from extremely limited information using conditional generative adversarial networks". In: *Physical Review E* 100 (Sept. 2019). DOI: 10.1103/PhysRevE.100.033308.

[20] 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* 32.2 (2015), pp. 55–66. DOI: 10.1109/MSP.2014.2369251.

[21] Giri, R., Isik, U., and Krishnaswamy, A. "Attention Wave-U-Net for Speech Enhancement". In: *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2019, pp. 249–253. DOI: 10.1109/WASPAA.2019.8937186.

[22] Grais, E. M., Sen, M. U., and Erdogan, H. *Deep neural networks for single channel source separation*. 2013. arXiv: 1311.2746 [cs.NE].

[23] Hershey, J. and Casey, M. *Audio-Visual Sound Separation Via Hidden Markov Models*. 2001.

[24] Huang, P. et al. "Singing-Voice Separation from Monaural Recordings using Deep Recurrent Neural Networks". In: *ISMIR*. 2014.

[25] Hyvärinen, A. and Oja, E. "Independent component analysis: algorithms and applications". In: *Neural Networks* 13.4 (2000), pp. 411–430. ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-6080(00)00026-5. URL:

[26] Jansson, A. et al. "Singing Voice Separation with Deep U-Net Convolutional Networks". In: *ISMIR*. 2017.

[27] Kadioglu, B. et al. *An empirical study of Conv-TasNet*. 2020. arXiv: 2002.08688 [eess.AS].

[28] Karttunen, L. "Applications of Finite-State Transducers in Natural-Language Processing". In: vol. 2088. Nov. 2000. DOI: 10.1007/3-540-44674-5_2.

[29] Kaspersen, E., Kounalakis, T., and Erkut, C. "Hydranet: A Real-Time Waveform Separation Network". In: May 2020, pp. 4327–4331. DOI: 10.1109/ICASSP40776.2020.9053357.

[30] Kingma, D. and Ba, J. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations* (Dec. 2014).

[31] Lambert, R. "Difficulty measures and figures of merit for source separation". In: *Proc. Int. Symp. ICA and BSS (ICA 99)* (Nov. 1999).

[32] Le Roux, J., Ono, N., and Sagayama, S. "Explicit Consistency Constraints for STFT Spectrograms and their Application to Phase Reconstruction". In: (Jan. 2008).

[33] Lea, C. et al. *Temporal Convolutional Networks: A Unified Approach to Action Segmentation*. 2016. arXiv: 1608.08242 [cs.CV].

[34] Leonard, M. K. et al. "Perceptual restoration of masked speech in human cortex". In: *Nature Communications* 7.1 (Dec. 2016), p. 13619. ISSN: 2041-1723. DOI: 10.1038/ncomms13619. URL:

[35] Luo, Y., Chen, Z., and Yoshioka, T. *Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation*. 2020. arXiv: 1910.06379 [eess.AS].

[36] Luo, Y. and Mesgarani, N. "TasNet: time-domain audio separation network for real-time, single-channel speech separation". In: *CoRR* abs/1711.00541 (2017). arXiv: 1711.00541. URL:

[37] 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* 27.8 (Aug. 2019), pp. 1256–1266. ISSN: 2329-9304. DOI: 10.1109/taslp.2019.2915167. URL:

[38] Manilow, E. et al. "Cutting Music Source Separation Some Slakh: A Dataset to Study the Impact of Training Data Quality and Quantity". In: *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE. 2019.

[39] McFee, B., Salamon, J., and Bello, J. P. *Adaptive pooling operators for weakly labeled sound event detection*. 2018. arXiv: 1804.10070 [cs.SD].

[40] Meseguer-Brocal, G., Cohen-Hadria, A., and Peeters, G. "Dali: A large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm". In: 2019.

[41] Mikolov, T. et al. "Recurrent neural network based language model". In: *INTERSPEECH*. 2010.

[42] Mimilakis, S. I. et al. "Monaural Singing Voice Separation with Skip-Filtering Connections and Recurrent Inference of Time-Frequency Mask". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 721–725. DOI: 10.1109/ICASSP.2018.8461822.

[43] Mimilakis, S. I. et al. "A Recurrent Encoder-Decoder Approach with Skip-filtering Connections for Monaural Singing Voice Separation". In: *CoRR* abs/1709.00611 (2017). arXiv: 1709.00611. URL:

[44] Nakamura, T. and Saruwatari, H. *Time-Domain Audio Source Separation Based on Wave-U-Net Combined with Discrete Wavelet Transform*. 2020. arXiv: 2001.10190 [cs.SD].

[45] Ono, N. et al. "The 2015 Signal Separation Evaluation Campaign". In: *International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*. Vol. 9237. Latent Variable Analysis and Signal Separation. Liberec, France, Aug. 2015, pp. 387–395. DOI: 10.1007/978-3-319-22482-4_45. URL:

[46] Patterson, R. et al. "An efficient auditory filterbank based on the gammatone function". In: (Jan. 1988).

[47] Perraudin, N., Balazs, P., and Søndergaard, P. "A Fast Griffin–Lim Algorithm". In: Oct. 2013, pp. 1–4. DOI: 10.1109/WASPAA.2013.6701851.

[48] Al-Radhi, M. S., Csapó, T. G., and Németh, G. *RNN-based speech synthesis using a continuous sinusoidal model*. 2019. arXiv: 1904.06075 [cs.SD].

[49] Raffel, C. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. 2019.

[50] Rafii, Z. et al. *The MUSDB18 corpus for music separation*. Dec. 2017. DOI: 10.5281/zenodo.1117372. URL:

[51] 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*. Ed. by N. Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[52] Ronneberger, O., Fischer, P., and Brox, T. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: (May 2015).

[53] Roux, J. L. et al. *SDR - half-baked or well done?* 2018. arXiv: 1811.02508 [cs.SD].

[54] Roweis, S. “One Microphone Source Separation”. In: *Adv Neural Inform Process Syst* 13 (Feb. 2001).

[55] Saito, K. et al. *Sampling-Frequency-Independent Audio Source Separation Using Convolution Layer Based on Impulse Invariant Method*. 2021. arXiv: 2105.04079 [cs.SD].

[56] Samuel, D., Ganeshan, A., and Naradowsky, J. *Meta-learning Extractors for Music Source Separation*. 2020. arXiv: 2002.07016 [cs.SD].

[57] Shi, Z. et al. “Deep Attention Gated Dilated Temporal Convolutional Networks with Intra-Parallel Convolutional Modules for End-to-End Monaural Speech Separation”. In: Sept. 2019, pp. 3183–3187. DOI: 10.21437/Interspeech.2019-1373.

[58] Smaragdis, P. “Blind separation of convolved mixtures in the frequency domain”. In: *Neurocomputing* 22.1 (1998), pp. 21–34. ISSN: 0925-2312. DOI: https://doi.org/10.1016/S0925-2312(98)00047-2. URL:

[59] Smith, L. N. *Cyclical Learning Rates for Training Neural Networks*. 2017. arXiv: 1506.01186 [cs.CV].

[60] Srivastava, R. K., Greff, K., and Schmidhuber, J. “Highway Networks”. In: *CoRR* abs/1505.00387 (2015). arXiv: 1505.00387. URL:

[61] Stoller, D., Ewert, S., and Dixon, S. “Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation”. In: *CoRR* abs/1806.03185 (2018). arXiv: 1806.03185. URL:

[62] Stöter, F.-R., Liutkus, A., and Ito, N. “The 2018 Signal Separation Evaluation Campaign”. In: *Proc. LVA/ICA 2018*. Guildford, UK, 2018.

[63] Takahashi, N., Goswami, N., and Mitsufuji, Y. “MMDenseLSTM: An efficient combination of convolutional and recurrent neural networks for audio source separation”. In: *CoRR* abs/1805.02410 (2018). arXiv: 1805.02410. URL:

[64] Takahashi, N. et al. “PhaseNet: Discretized Phase Modeling with Deep Neural Networks for Audio Source Separation”. In: Sept. 2018, pp. 2713–2717. DOI: 10.21437/Interspeech.2018-1773.

[65] Tolonen, T. “Methods for separation of harmonic sound sources using sinusoidal modeling”. In: (Jan. 2012).

[66] Uhlich, S., Giron, F., and Mitsufuji, Y. “Deep neural network based instrument extraction from music”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 2135–2139. DOI: 10.1109/ICASSP.2015.7178348.

[67] Uhlich, S. et al. “Improving music source separation based on deep neural networks through data augmentation and network blending”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 261–265. DOI: 10.1109/ICASSP.2017.7952158.

[68] Vembu, S. and Baumann, S. “Separation of Vocals from Polyphonic Audio Recordings”. In: *ISMIR*. 2005.

[69] Vincent, E., Gribonval, R., and Fevotte, C. “Performance measurement in blind audio source separation”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.4 (2006), pp. 1462–1469. DOI: 10.1109/TSA.2005.858005.

[70] 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* 15.3 (2007), pp. 1066–1074. DOI: 10.1109/TASL.2006.885253.

[71] Virtanen, T. “ALGORITHM FOR THE SEPARATION OF HARMONIC SOUNDS WITH TIME FREQUENCY SMOOTHNESS CONSTRAINT”. In: (Jan. 2003).

[72] Virtanen, T. and Klapuri, A. “Separation of harmonic sound sources using sinusoidal modeling”. In: vol. 2. Feb. 2000, II765–II768 vol.2. ISBN: 0-7803-6293-4. DOI: 10.1109/ICASSP.2000.859072.

[73] Wang, Y., Narayanan, A., and Wang, D. “On Training Targets for Supervised Speech Separation”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.12 (2014), pp. 1849–1858. DOI: 10.1109/TASLP.2014.2352935.

[74] Ward, D. et al. “BSS Eval or Peass? Predicting the Perception of Singing-Voice Separation”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 596–600. DOI: 10.1109/ICASSP.2018.8462194.

[75] Zhang, J. and Man, K. “Time series prediction using RNN in multi-dimension embedding phase space”. In: *SMC’98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*. Vol. 2. 1998, 1868–1873 vol.2. DOI: 10.1109/ICSMC.1998.728168.