КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

Кваліфікаційна робота

на здобуття ступеня бакалавра
за спеціальністю 121 Інженерія програмного забезпечення

на тему:

АВТОМАТИЧНА ТРАНСКРИПЦІЯ МУЗИКИ З ВИКОРИСТАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Виконала студентка 4-го курсу
Марина МОНТАГ _____
(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Ярослав ЛІНДЕР _____
(підпис)

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри
інтелектуальних програмних систем
«27» травня 2022 р.,
протокол № 10
Завідувач кафедри
О. І. ПРОВОТАР _____
(підпис)

Київ - 2022

РЕФЕРАТ

Обсяг роботи 56 сторінок, 12 ілюстрацій, 9 таблиць, 20 джерел посилань.

ТРАНСКРИПЦІЯ МУЗИКИ, ПАРТИТУРА, НЕЙРОННА МЕРЕЖА, ГЛИБОКА НЕЙРОННА МЕРЕЖА, РОЗДІЛЕННЯ ДЖЕРЕЛА ЗВУКУ, ТОНАЛЬНІСТЬ, ПЕРКУСІЯ, РОЗРАХУНОК ДОМІНАНТНОЇ ЧАСТОТИ ЗВУКУ.

Об’єктом роботи є процес автоматизації транскрипції музичної композиції за допомогою програмного засобу «Система для автоматичної транскрипції музичної композиції». Предметом роботи є програмний засіб для автоматичної транскрипції музичної композиції, що використовує методи машинного навчання для виокремлення тональних та перкусійних музичних компонент з композиції.

Метою роботи є створення програмного засобу для автоматичної транскрипції музичної композиції та аналіз систем розділення джерела звуку, що використовують методи машинного навчання.

Методи розроблення: аналіз програмних продуктів на основі бібліотеки Tensorflow, з попередньо підготовленими моделями нейронних мереж для поділу на 4 музичні композиції: vocals, drums, bass і other. Створення алгоритму для транскрибування музики. Розробка власного програмного засобу мовою Python. Інструменти розроблення: безплатний, вільно поширюване інтегроване середовище розробки PyCharm Community IDE 2021.2.1, мова програмування Python.

Результати роботи: виконано загальний огляд бібліотеки Tensorflow для роботи з нейронними мережами, проаналізовано системи розділення джерела звуку на основі Tensorflow, розроблено програмний продукт «Система для автоматичної транскрипції музичної композиції», який дозволяє наочно демонструвати процеси поділу музичної композиції на декілька композицій, в яких наявні окремі музичні компоненти, та автоматичної транскрипції музики.

Через те, що основна задача полягає у написанні коду, то й застосовані методи програмування – об'єктноорієнтоване, функціональне та імперативне. Оскільки основою програмного засобу є нейронна мережа, то використовуються методи

машинного та глибокого навчання.

Програмний продукт «Система для автоматичної транскрипції музичної композиції» має багато потенційних застосувань: перевірка музиної композиції на плагіат, пошук музичної композиції по музичним нотам, створення реміксів, виконання музичної композиції на інших інструментах маючи ноти.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

IDE – Integrated Design Environment, інтегроване середовище розробки;

ATM – Автоматична транскрипція музики

НМ – Нейронна мережа;

DNN – Deep Neural Network, глибока нейронна мережа;

FNN – Feed-Forward Networks, мережі прямого зв'язку;

CNN – Convolutional Neural Network, згорткова нейронна мережа;

MSS – music source separation, розділення джерел музики;

MIR – music information retrieval, пошук музичної інформації;

LSTM – Long-short term memory;

AI – artificial intelligence, штучний інтелект;

ML – machine learning, машинне навчання;

SDR – Signal to Distortion Ratios, співвідношення сигналу до спотворення;

ISR – Source to Spatial Distortion Image, джерело зображення просторового спотворення;

SIR – Source to Interference Ratios, співвідношення джерела та перешкод;

SAR – Sources to Artifacts Ratios, відношення джерел до артефактів;

SED – Sound Event Detection, виявленням звукових подій;

MPE – Multi-Pitch Estimation, оцінка на основі декількох тонів;

FFT – Fast Fourier Transformation, швидке перетворення Фур'є

ВСТУП

Оцінка сучасного стану об’єкта розробки. Можливість транскрибувати музичне аудіо в музичну нотацію є захоплюючим прикладом людського інтелекту. Це включає сприйняття (аналіз складних слухових сцен), пізнання (розпізнавання музичних предметів), представлення знань (формування музичних структур) і умовивід (перевірка альтернативи гіпотези). Автоматична транскрипція музики (AMT), тобто розробка обчислювальних алгоритмів для перетворення акустичної музики сигналів у певну форму нотної грамоти, є складним завданням в обробці сигналів і штучному інтелекті. Воно містить кілька підзадач, включаючи оцінку (багато) тону, виявлення початку і зсуву, розпізнавання інструментів, відстеження ударів і ритму, інтерпретацію виразної синхронізації та динаміки, а також набір партитур. Але використання засобів машинного навчання для отримання необхідних музичних компонент, з допомогою яких виконується транскрибування, є досить прогресивним підходом, що полегшує розв'язання цієї проблеми.

Завдяки розвитку глибокого навчання, були запропоновані та досягли значного прогресу різноманітні мережі поділу джерел звуку. Однак дослідження основних механізмів поділу все ще знаходяться на стадії становлення. Розділення джерела звуку є важливою частиною машинного прослуховування і є корисним для багатьох реальних аудіододатків. Розділення мовлення часто служить передньою частиною високорівневої обробки, наприклад автоматичного розпізнавання мовлення та системи слухових апаратів. Розділення звуку навколишнього середовища є критичним у моніторингу аномального звуку та інтелектуальної оцінки шуму. Музичне розділення також необхідне для пошуку звукової інформації та автоматичної транскрипції музики.[1]

Актуальність роботи та підстави для її виконання. На сьогодні на просторах Інтернету існує величезна кількість музичних треків, і з кожним днем ця кількість тільки зростає. Часто музику починають перегравати на інших інструментах, але щоб зробити це більш якісно, а не на слух, виникає потреба

вилучення музичної нотації з цих композицій.

Також музичне транскрибування може використовуватись для перевірки треку на плагіат, оскільки відмінність темпу музичної композиції або використання інших музичних інструментів не скасовує провини у плагіаті, в тому випадку якщо музичні ноти ті самі.

Ще однією ціллю використання створеного програмного засобу може бути пошук композицій за наданою музичною нотацією.

Мета й завдання роботи. Метою кваліфікаційної роботи є створення програмного засобу для розв’язування задачі розділення аудіо джерел. Для досягнення цієї мети поставлено такі завдання.

- Дослідити наявні електронні засоби розділення джерел музики.
- Дослідити різні моделі нейронних мереж для розв'язання задачі розділення джерел музики.
- Створити алгоритм для втілення автоматизації транскрибування музичної композиції.
- Формування користувацьких вимог до програмного засобу
- Розробка та тестування програмного засобу

Об'єкт, методи й засоби розроблення. Об’єктом розроблення програмного засобу «Система для автоматичної транскрипції музичної композиції» є процес автоматизації транскрибування музичної композиції.

Розробці програмного засобу передувало аналіз та вибір системи розділення джерела звуку, що виокремлює необхідні для подальшої роботи музичні компоненти. Основу для цього склав аналіз різних типів нейронних мереж для розділення джерела звуку.

Під час розробки програмного продукту використана еволюційна модель, заснована на таких принципах. Виконується аналіз вже наявних сучасних моделей нейронних мереж, які виконують поставлене перед нами завдання. Після цього виконується аналіз систем розділення джерела звуку, що використовують

відповідні типи нейронних мереж та вибір тієї, що має найкращі показники якості. Далі відбувається робота алгоритму, що використовує музичні компоненти отримані в результаті розділення джерела звуку.

Як інструмент створення програмного засобу було обрано PyCharm 2021.2.1 – інтегроване середовище розробки (IDE) мовою програмування Python.

Переваги використання мови Python для машинного навчання та проєктів, керованих штучним інтелектом, включають простоту та послідовність, гнучкість, доступ до потужних бібліотек і фреймворків AI та машинного навчання (ML), незалежність від платформи та великі спільноти. Ці речі підвищують популярність мови.

Можливі сфери застосування. Ця тема має багато цікавих застосувань, таких як редагування аудіо, вилучення звукових зразків, перепросторованість або змішування, перевірка музичних треків на плагіат та пошук треків за звуковим зразком або фрагментом музичної нотації.

Взаємозв'язок з іншими роботами. Робота виконувалася сумісно з використанням вже готової бібліотеки для розділення джерела звуку Spleeter, написаної мовою програмування Python.

РОЗДІЛ 1 НЕЙРОННІ МЕРЕЖІ ДЛЯ РОЗДІЛЕННЯ ДЖЕРЕЛА ЗВУКУ

1.1 Як працює розділення джерела звуку?

Проблема поділу джерел звуку є добре вивченою проблемою, також відома як проблема коктейльної вечірки, тобто проблема відокремлення конкретного джерела звуку, що цікавить, у середовищі, повному слухових стимулів і шумів. Проблема привернула увагу дослідників протягом останніх 25 років, і було запропоновано кілька рішень, які можуть розв'язувати проблему в ряді дуже особливих випадків. Проте, існує кілька сценаріїв, коли запропоновані методи дають збій, що робить проблему нерозв’язаною. Нещодавно завдяки широкому успіху онлайн-провайдерів музики, які транслюють аудіоконтент мільйонам користувачів, розділення джерел звуку знову стало модним, щоб надати передплатникам більше інтерактивного аудіовмісту.

За останні десятиліття було розроблено багато моделей поділу джерел з різними характеристиками, які можна розділити на два типи. Один, обчислювальний аналіз слухових сцен (CASA), імітує слухову систему для окремих джерел, тоді як інший розглядає розділення джерела як контрольовану проблему навчання та вирішує її за допомогою розробки статистичних моделей. Перший підхід (моделі CASA) має тенденцію до розділення джерел на основі слухових механізмів поділу. Ранні моделі були зосереджені на виділенні деяких акустичних атрибутів, таких як тональність, початок і модуляція амплітуди, і згодом були засновані на близькості, подібності або спільній долі цих атрибутів. Більшість з цих моделей біологічно правдоподібні і їх легко пояснити. Однак нинішнього розуміння слухової нейронауки недостатньо для розробки системи, настільки розумної, як людина. Ці моделі зазвичай ефективні для простих стимулів, але не можуть адаптуватися до природних джерел у складних акустичних сценах. Другий підхід розробляє контрольовані моделі на основі оптимізації завдання на певному наборі навчальних даних. Завдяки розвитку глибокого навчання нещодавно запропоновані системи глибокого поділу джерел досягли значного прогресу, який почав працювати на

людському рівні для природного поділу джерел. Другий підхід привернув значну увагу через його чудову продуктивність. Однак, на відміну від багатьох досліджень, спрямованих на пошук кращих мережевих структур або параметрів для покращення продуктивності розділення, це дослідження зосереджується на основних механізмах розділення цих глибоких контрольованих мереж.

1.2 Підходи розділення джерела звуку

Поділ музики на окремі інструменти є складною проблемою. У цьому підрозділі описуються дві різні архітектури глибокої нейронної мережі для цього завдання, пряма і рекурентна, і показано, що кожна з них дає найсучасніші результати в наборі даних DSD100. Для рекурентної мережі використовується збільшення даних під час навчання і показано, що навіть прості мережі поділу схильні до переобладнання, якщо не використовується розширення даних. Крім того, пропонується поєднати обидві системи нейронної мережі, де лінійно комбінуються їхні вихідні дані, а потім виконується багатоканальний фільтр Вінера після обробки. Ця схема змішування дає найкращі результати, про які повідомлялося на сьогодні в наборі даних DSD100.

У цій роботі вивчається проблема поділу музики на музичні треки. Зокрема, розглядається поділ на вокал і трек супроводу, або, більш дрібно, на треки для вокалу, баса, ударних тощо. Різні програми вимагають таких оцінок треку, починаючи від систем караоке, які використовують поділ на інструментальну та вокальну доріжку, до збільшення мікшування, де намагаються отримати багатоканальну версію пісні.

Попри свій складний характер, розділення джерел музики (MSS) викликає все більший інтерес протягом останніх років. Особливо професійно змішане завдання поділу музики, яке називається MUS, допомогло розбудити цей інтерес. Був підготовлений новий набір даних MSD100, що складається з частин для навчання та тестування з 50 піснями кожна. Для кожної пісні доступні суцільна композиція і

чотири її джерела: бас, ударні, інше та вокал, і цей набір даних дозволив вперше оцінити різні методи поділу джерел у найрізноманітніших музичних жанрах. Набір даних MSD100 був додатково покращений за допомогою програмного забезпечення цифрової аудіо робочої станції та налаштувань мікшування, які використовують професійні аудіоінженери. Щоб відрізнити цей набір даних від попередньої версії, він називається DSD100.

Найкращі результати для завдання MUS були отримані за допомогою підходів, які використовували глибокі нейронні мережі (DNN), і це спостереження для MSS відповідає багатьом іншим програмам, де DNN перевершували попередні найкращі системи. Використовуючи навчальні корпуси (наприклад, частину Dev DSD100), можна вивчити нейронні мережі, які витягують конкретний цільовий інструмент із музичної композиції. Крім того, DNN також показали дуже хорошу продуктивність для відповідних проблем виділення мови з музики або для відокремлення вокалу від музики.

У цьому підрозділі надаються результати для підходу з прямим зв’язком із постобробкою багатоканального фільтра Вінера (MWF), оскільки до цього часу використовували лише одноканалові фільтри Вінера (SWF). Також досліджується використання рекурентної нейронної мережі для вилучення інструментів. Рекурентна структура має перевагу, що дозволяє краще враховувати контекстну інформацію з сусідніх кадрів суміші, що важливо для отримання інформації про тимчасову структуру пісні. Також вноситься пропозиція використовувати збільшення даних під час навчання. Це особливо важливо для навчання повторюваних нейронних мереж, оскільки використовувалась лише частина Dev DSD100 для вивчення цих мереж і не використовувались додаткові дані. Показано, що розширення даних допомагає навчитися розділяти мережі, які краще узагальнюють. Остаточний внесок полягає в тому, щоб показати, що змішування вихідних даних мережі перед постобробкою MWF може значно покращити продуктивність поділу для всіх джерел, а результати в наборі даних DSD100 є

найкращими, про які повідомлялося досі. Змішування, також відоме як злиття, алгоритмів MSS є досить новою темою в літературі з розділення джерел, де оцінки масок різних систем об'єднуються для покращення мовлення та розділення голосу відповідно. Крім того, обговорюється підготовка мультиконтекстних мереж - підхід кооперативного глибокого стекування, коли дві мережі з різними входами об'єднуються на рівні рівня. На відміну від цих підходів, тут змішуються вихідні дані двох різних мережевих структур, прямої та повторюваної, а також додатково виконуємо постобробку розширеного розділення фільтром Вінера. Цей додатковий крок дозволяє додатково покращити поділ, оскільки MWF обчислюється на основі кращих вихідних оцінок, які отримуються після змішування.

У цій статті використовуються такі позначення: **x** позначає вектор-стовпець, а **X** — матриця, де, зокрема, **I** — ідентична матриця. Транспонування матриці та евклідова норма позначаються $(.)^T$ і ||.|| відповідно.

Далі представлено основну проблему MUS. Нехай $\mathbf{x}(n) \in \mathbb{R}^2$ позначає стереосуміш у часовій області, яка, як відомо, складається з чотирьох джерел баса $s_B(n)$, барабанів $s_D(n)$, інших $s_O(n)$ та вокалу $s_V(n)$,

$$\mathbf{x}(n) = s_B(n) + s_D(n) + s_O(n) + s_V(n) = \sum_{i \in T} s_i(n) \quad (1)$$

з $T := \{B, D, O, V\}$. Мета MSS полягає в тому, щоб отримати хороші оцінки джерела стереосигналу $\hat{s}_i(n)$ таким чином, щоб вони були якомога ближчими до справжніх джерел $s_i(n)$. Популярним показником ефективності для оцінки якості поділу є BSS Eval, який також використовується в задачі MUS для порівняння різних підходів.

Нарешті, більшість підходів здійснюють поділ в області короткочасного перетворення Фур'є (STFT), і ми будемо позначати $\mathbf{X}(m, f) \in \mathbb{C}^2$, $S_i(m, f)$ і $\hat{S}_i(m, f)$ суміш, джерела і оцінки в області STFT, відповідно, де m дає індекс кадру, а f — індекс частотного бігу.

1.2.1 Загальний підхід DNN

На рис. 1 показано загальний підхід DNN для MSS, який використовується.

Замість використання однієї DNN для оцінки величин STFT, ми використовуємо окрему DNN для кожного інструменту, який навчений вилучати цей інструмент із суміші, що дозволяє нам краще масштабувати до сумішей з різними складовими інструментами. Після додаткового зниження дискретизації обчислюється STFT музики, яку ми хочемо відокремити. Величини STFT передаються – можливо разом з деякими попередніми/наступними контекстними кадрами – через DNN і отримується оцінка величин STFT для кожного джерела. Ці оцінки величини потім об'єднуються разом із фазою суміші для обчислення зворотного STFT. Нарешті, для покращення ефективності поділу застосовується постобробка SWF або MWF. Застосування SWF/MWF можна розглядати як постобробку, яка гарантує, що сума чотирьох оцінок дає вихідну суміш. Це значно зменшує перешкоди від інших джерел і артефакти поділу, і, отже, SWF/MWF є важливим етапом поділу. Багатоканальний фільтр Вінера передбачає модель сигналу

$$\mathbf{X}(m,f) = S_i(m,f) + Z_i(m,f) \qquad (2)$$

з $i \in T$ і $Z_i(m,f) = \sum_{j \in T \setminus i} S_j(m,f)$, де кожен частотно-часовий контейнер $S_i(m,f)$ STFT є комплексним Гауссовим з нульовим середнім і матрицею коваріацій $v_j(m,f)R_j(f)$. $v_j(m,f)$ – це спектральна щільність потужності (PSD), а $R_j(f)$ – незалежна від часу просторова коваріаційна матриця, відповідно. Оцінка мінімальної середньоквадратичної помилки для $S_i(m,f)$з $\mathbf{X}(m,f)$ добре відома і визначається формулою

$$\hat{S}_i(m,f) = v_j(m,f)R_j(f)\left(\sum_{j \in T} v_j(m,f)R_j(f)\right)^{-1}\mathbf{X}(m,f). \quad (3)$$

Щоб застосувати MWF, нам потрібно оцінити PSD $v_i(m,f)$ і матриці просторової коваріації $R_i(f)$, що в цьому випадку робиться на основі вихідних даних кожного інструменту DNN. Зокрема, ми оцінюємо їх з M послідовних кадрів за

$$\hat{v}_i(m,f) = \frac{1}{2}\|\hat{S}_i(m,f)\|^2,\ \hat{R}_i(m,f) = \frac{\sum_{m=1}^{M} \hat{S}_i(m,f)\hat{S}_i(m,f)^H}{\sum_{m=1}^{M} \hat{v}_i(m,f)}. \quad (4)$$

Оцінку для просторової коваріаційної матриці можна вважати зваженою версією класичної оцінки максимальної правдоподібності. Більша вага приділяється часовим

частотним контейнерам з високою енергією, оскільки ми можемо припустити, що тоді ми маємо краще співвідношення сигнал/завада. Згодом ми побачимо, що MWF покращує продуктивність поділу. Особливо під час прослуховування результатів поділу можна спостерігати набагато більш стабільне стереорозташування вилучених джерел і не з'являється тривожних ефектів флангу.

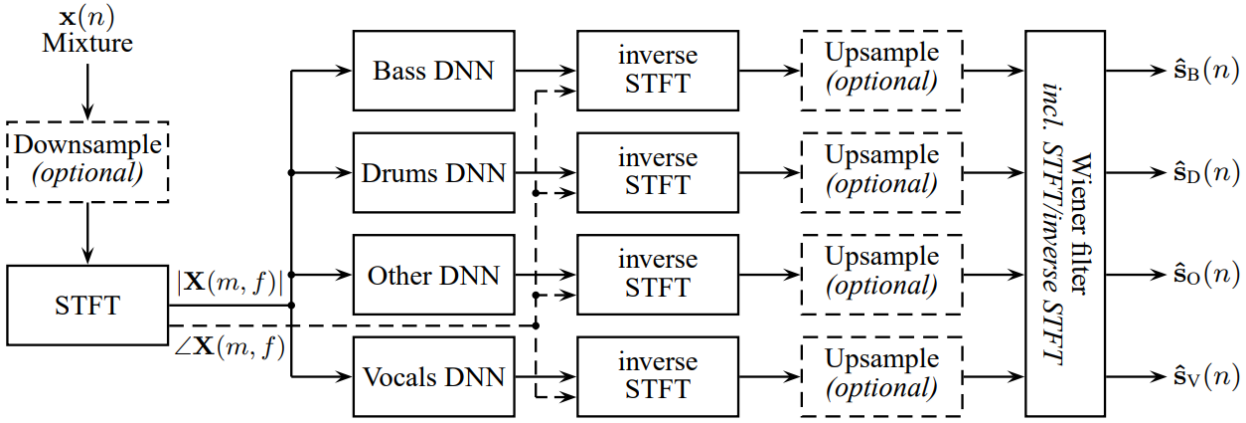


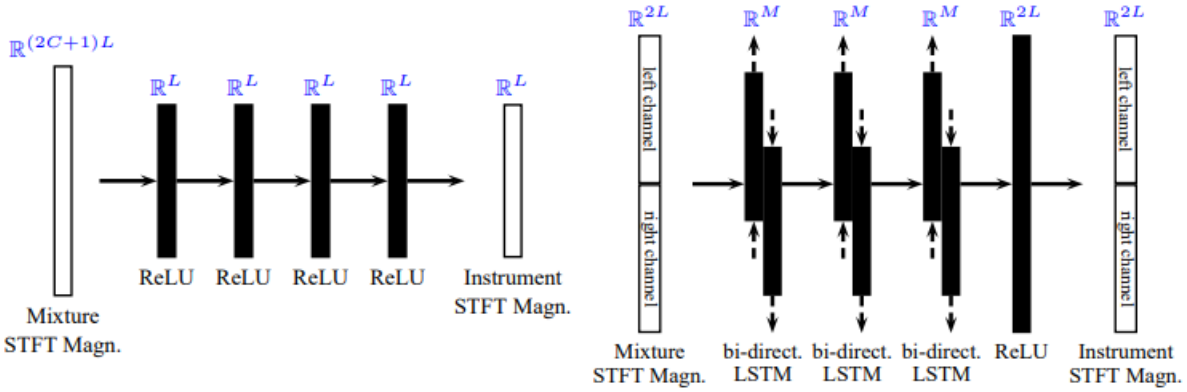
Рисунок 1. Загальний підхід DNN для MSS музики

1.2.2 Мережі прямого зв'язку (FNN)

Перший підхід використовує архітектуру прямої подачі і схематично показаний на рис. 2(a). Розглядаються два різних набори мереж:

- **FNN-1:** Для кожного приладу ми тренуємо мережу випрямлених лінійних одиниць (ReLU) з K = 3 шарами з P = 2 · 106 навчальних вибірок. Навчальний матеріал складається з коротких інструментальних циклів, які не залежать від частини Dev або Test DSD100. Навчальні вибірки створюються шляхом випадкового вибору циклів для кожного інструменту та змішування їх із випадковими амплітудами. Кількість кадрів контексту, що не перекриваються, дорівнює C = 3, і, оскільки ми використовуємо розмір FFT 1024, вхідний вектор має довжину (2C + 1) · 513 = 3 591. Нарешті, цей набір мереж використовує одноканалову постобробку фільтра Вінера.
- **FNN-2**: Цей набір нейронних мереж нещодавно навчений. На додаток до

набору даних FNN-1 ми використовуємо не кровоточиві стебла з MedleyDB та матеріал, який міститься в частині Dev DSD100. Крім того, він використовує вхідні кадри з 50% перекриттям і C = 8 контекстних кадрів. Аналіз головних компонентів (PCA) використовується до половини розміру вхідного вектора. Це дозволяє проводити тренування з більшою кількістю навчальних вибірок, а саме P = 1,2 · 107. Крім того, FNN-2 використовує K = 4 шари ReLU, тоді як усі інші налаштування та процедура навчання такі ж, як і для FNN-1.



(a) Прямий зв'язок (тут: FNN-2) (b) BLSTM (тут: BLSTM-3)

Рисунок 2. DNN архітектури для вилучення інструменту

У таблиці 1 порівнюються два набори DNN у тестовій частині DSD100, де значення отримані шляхом усереднення значень відношення сигнал/спотворення (SDR) для кожної пісні, а потім обчислення медіани для всіх 50 пісень. Ця таблиця містить також нижню базову лінію BL і верхню базову лінію BU, де BL використовує вихідну суміш, масштабовану на $\frac{1}{4}$ як розділення, а BU дає продуктивність передбачення ідеальної маски співвідношення. Щоб створити оцінку для акомпанементу, тобто для всіх джерел, крім вокалу, обчислюємо

$\hat{s}_A(n) = x(n) - \hat{s}_V(n)$.

Network	SDR in dB				
	Bass	Drums	Other	Vocals	Acco.
BL	0.72	0.95	1.43	0.35	6.82
BU	6.26	7.96	7.76	10.16	16.38
FNN-1	2.22	3.08	2.48	3.63	10.19
FNN-2	2.54	3.75	2.92	4.47	11.12
BLSTM-1	2.30	3.71	2.98	3.69	10.33
BLSTM-2	2.77	3.78	3.44	4.91	11.35
BLSTM-3	2.89	4.00	3.24	4.86	11.26

Таблиця 1. Мережі FNN і BLSTM на тестовій вибірці DSD100

З цієї таблиці ми можемо спостерігати, що FNN-2 демонструє в середньому на 0,6 дБ кращий SDR, ніж FNN-1. Дві основні відмінності між двома мережами полягають у тому, що FNN-2 додатково використовує частину Dev DSD100 і MWF замість SWF. Ми також навчали набір мереж лише з кожною з цих змін і могли помітити, що використання частини Dev покращує SDR в середньому на 0,4 дБ, а використання MWF покращує ще на 0,2 дБ.

1.2.3 Двонапрямні мережі LSTM

Другий підхід використовує архітектуру рекурентної нейронної мережі з двонапрямними шарами LSTM (BLSTM) і показаний на рис. 2(b). У порівнянні з традиційними рекурентними нейронними мережами, мережі (B)LSTM мають перевагу, оскільки вони не страждають від проблеми градієнта, що зникає/вибухає, і, отже, досить популярні для проблем, які потребують пам’яті. Для проблеми поділу такий рекурентний підхід має перевагу, бо ми краще беремо до уваги інформацію про контекст, ніж використовуємо супервектори з сусідніми кадрами величин. Мережа може запам’ятовувати довші залежності, і це допомагає покращити продуктивність MSS.

Використано три різні навчені набори мереж, які відрізняються кількістю шарів BLSTM. Кожен шар BLSTM складається з 250 прямих і 250 зворотних комірок LSTM, вихідні дані яких об’єднані, щоб утворити загальний вихід шару.

Інструментальні цикли, які ми використовували для навчання FNN занадто короткі для навчання послідовності мереж BLSTM. Тому їх навчено виключно на частині Dev DSD100, а збільшення даних використовується, щоб уникнути перенавчання. Вхідними сигналами до мереж BLSTM є стереокадри амплітуди з лівого та правого каналів, які отримані за допомогою розміру кадру 1024 вибірки з 50% перекриттям. Виходом є оцінений кадр стереозв'язку цільового інструменту. Вихідні дані знову обробляються використовуючи MWF для покращення результатів.

У таблиці 1 показані результати трьох мереж BLSTM. Ми можемо помітити, що дві глибоші мережі з K = 2 і K = 3 шарами BLSTM показують значно кращу продуктивність, ніж BLSTM-1. Порівнюючи BLSTM-2 і BLSTM-3 з FNN-2, ми можемо помітити, що BLSTM-3 демонструє більш послідовне покращення для всіх інструментів у порівнянні з FNN-2 (порівняно значення SDR для барабанів.

1.2.4 Збільшення даних під час навчання

Добре відомо, що розширення даних може значно покращити продуктивність DNN. Розглядалась така модифікація даних використана на льоту, коли створювалась навчальна мініпакетна послідовність для BLSTM:

- випадкова зміна лівого/правого каналу для кожного інструменту,
- випадкове масштабування з рівномірними амплітудами від 0,25 до 1,25 включно,
- випадковий поділ на послідовності для кожного інструменту,
- довільне змішування інструментів з різних пісень.

Кожен мініпакет складається з десяти послідовностей, де кожна послідовність має довжину 500 стереокадрів STFT. Щоб довести ефективність збільшення даних, ми навчили мережу BLSTM-1 для вилучення вокалу також без збільшення, і результати наведені в таблиці 2, де акомпанемент знову оцінюється як $\hat{s}_A(n) = x(n) - \hat{s}_V(n)$. У цій таблиці наведено вихідні значення SDR без постобробки SWF/MWF, оскільки ціллю було чітко показати ефект збільшення даних.

SDR in dB				
Інстр.	**Мережа**	**DEV**	**Test[Всі]**	**Test[Нові виконавці]**
Вокал	BL	0.91	0.35	0.77
	BLSTM-1 **без збільшення даних**	7.13	3.37	3.19
	BLSTM-1 **зі збільшенням даних**	6.19	3.59	3.79
Супровід	BL	6.57	6.82	6.49
	BLSTM-1 **без збільшення даних**	13.33	9.71	9.53
	BLSTM-1 **зі збільшенням даних**	12.23	9.93	9.64

Таблиця 2. Ефект збільшення даних (data augmentation) для BLSTM-1

З таблиці 2 ми бачимо, що збільшення є вигідним. Вокал і акомпанемент збільшуються в середньому на 0,2 дБ, якщо ми розглянемо всі тестові пісні, і навіть на 0,35 дБ, якщо розглядати лише підмножину тестових пісень, де немає пісні того самого виконавця в частині Dev. Результати цієї підмножини можна використовувати для кращої оцінки ефективності узагальнення мережі, і ми бачимо, що розширення даних допомагає вивчати мережі, які краще узагальнюють для нових виконавців.

1.2.5 Змішування мереж

Для подальшого покращення продуктивності MSS ми пропонуємо об’єднати результати прямої передачі та мережі BLSTM. Добре відомо, що змішування різних систем може покращити продуктивність, якщо помилки кожної окремої системи не корельовані. Оскільки ми об’єднуємо дві мережі, які відрізняються структурою мережі та навчальним матеріалом, ми можемо припустити, що обидві системи досить різні, щоб їх поєднання було вигідним.

Зокрема, ми використовується незмінне в часі змішування, яке приймає форму

$$\hat{s}_{i,BLEND}(n) = \lambda \hat{s}_{i,FNN}(n) + (1 - \lambda)\hat{s}_{i,BLSTM}(n), \quad (5)$$

тобто вихідні дані DNN лінійно змішуються для кожного інструменту $i \in T$. Нарешті, використовується постобробка MWF для покращення результатів. На рис. 3 показано покращення SDR щодо BL для змішування двох найкращих систем із

розд. 1.2.2 і 1.2.3, тобто FNN-2 і BLSTM-3. Обирається вагу змішування λ, щоб мати значення λ = 0,25, оскільки це найкраще середнє значення покращення для вибірки Dev DSD100. Дивлячись на тестову вибірку, можна помітити, що всі інструменти покращуються завдяки такому вибору і що вона дає в середньому покращення SDR на 0,2 дБ у порівнянні з BLSTM-3. Зокрема, для вокалу та акомпанементу найбільше покращення – 0,4 дБ. Оскільки цікавить ефективність саме узагальнення підходу змішування, проведено неформальний тест на прослуховування з популярними хітами, і було помічено значне покращення цієї системи змішування. Особливо покращується тимчасова стабільність вилучених джерел, тобто в розділених інструментах не відсутня частина джерел атаки/розпаду.

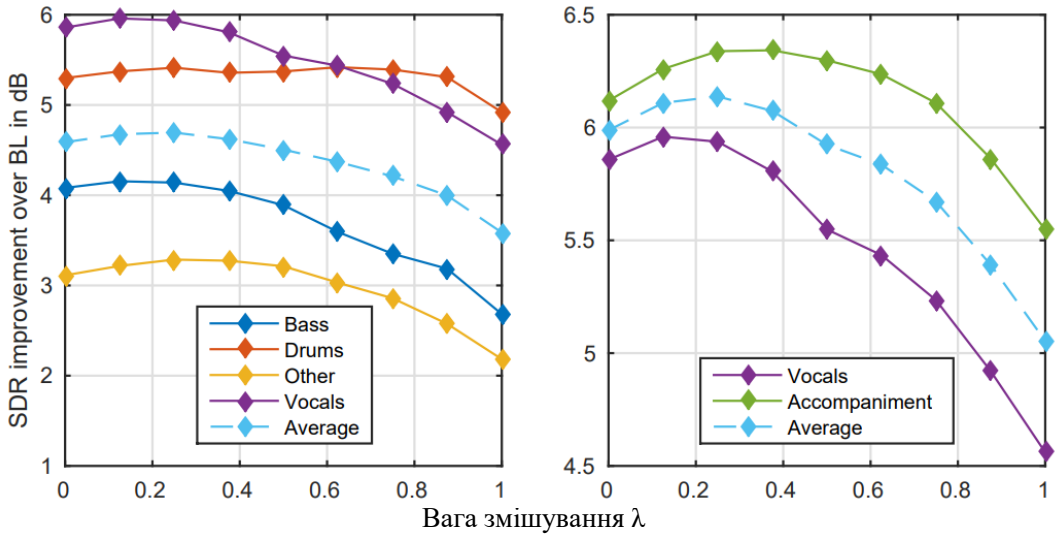
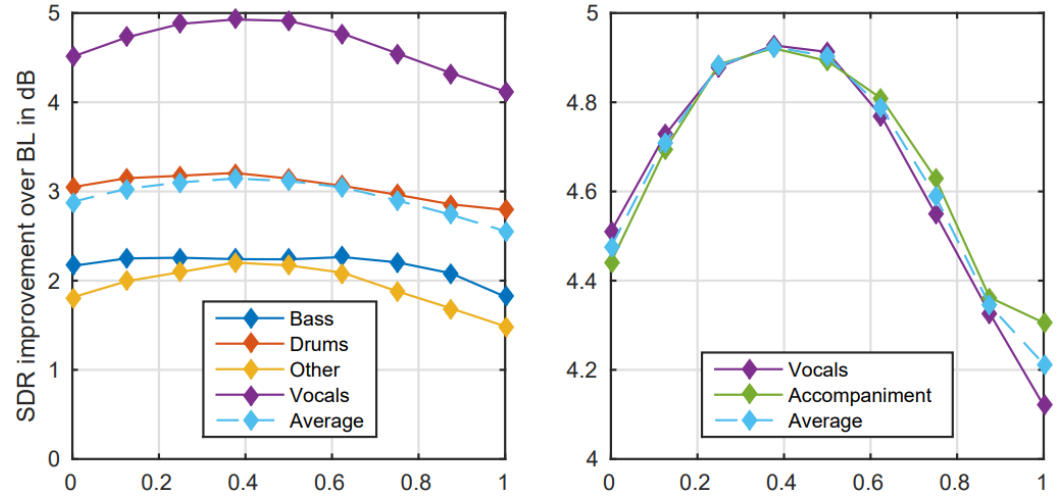


Рисунок 3(а). Dev вибірка DSD100



Вага змішування λ

Рисунок 3(б). Тестова вибірка DSD100

Нашу схему змішування (5) можна розглядати як розширення вивченого тимчасового злиття. Замість того, щоб лінійно об'єднувати системи після MWF, вихідні дані кожної DNN змішуються, а потім виконується постобробка MWF. Цей остаточний MWF допомагає зменшити перешкоди та досягти кращих результатів, оскільки отримуються кращі оцінки джерела, що, своєю чергою, забезпечує кращу багатоканальну фільтрацію Вінера. Порівнюючи обидві схеми, ми можемо помітити, що наше злиття в середньому на 0,1 дБ краще для SDR і 0,3 дБ для відношення сигнал/завада (SIR), ніж засвоєна схема тимчасового злиття.

Нарешті, також було досліджено ефект змішування для різних музичних жанрів у DSD100. Результати показані на рис. 4, і можна помітити, що змішування FNN-2 і BLSTM-3 часто навіть краще, ніж найкращі окремі мережі (20 разів із 35), що показує ефективність змішування.

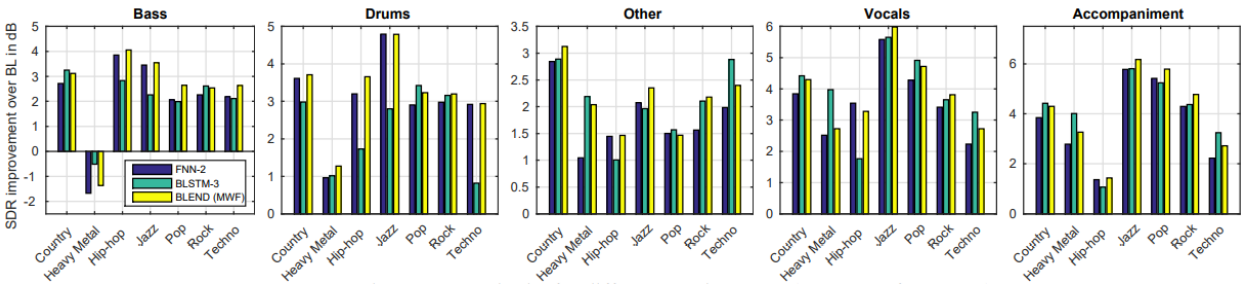


Рисунок 4. Покращення SDR в дБ для різних музичних жанрів (тестова вибірка DSD100)

1.2.6 Порівняння з іншими підходами

У цьому підрозділі змішана система MSS порівнюється з чотирма іншими добре відомих підходів з літератури:

- sNMF: підхід контрольованої розрідженої невід'ємної матриці (sNMF) з коефіцієнтом розрідженості μ = 5.
- dNMF: підхід із контрольованою дискримінаційною невід'ємною матричною факторизацією (dNMF), де використовується

дискримінаційна функція вартості, щоб краще вивчати базові вектори з пісень з Dev вибірки.

- DeepNMF: невід'ємна архітектура глибокої мережі, яка є результатом розгортання ітерацій NMF та знаходження їх параметрів.
- NUG: цей підхід використовує складну Гауссову частотно-часову модель, де оновлення спектра обчислюються за допомогою багатоканальних DNN. У порівнянні з підходом FNN, цей підхід ітеративно оновлює просторові та спектральні оцінки за принципом максимізації очікування (МО), тоді як ми використовуємо фільтр Вінера лише як постобробку, і можна вважати FNN окремим випадком виконання лише однієї ітерація МО. Система NUG, яка розглядається, показала дуже хорошу продуктивність і була позначена як NUG1.

Результати на тестовій вибірці DSD100 наведені в таблиці 3 і показані у вигляді квадратних діаграм на рис. 5. Для трьох підходів NMF вектори базису Q вивчаються для кожного інструменту та пісні з вибірки Dev з DSD100, які об'єднуються разом, щоб отримати $50 \cdot Q$ базисних векторів по кожному інструменту. Використовувалась різна кількість Q базисних векторів для однієї пісні, а саме $Q \in \{5, 10, 15, 20, 25, 30\}$, і була отримана найкраща середня продуктивність для $Q = 25$, яка наведена в таблиці 3. Серед методів NMF можна помітити, що метод DeepNMF працює найкраще. Оскільки всі системи NMF є одноканаловими підходами, також повторно використовується підхід змішування з SWF, щоб забезпечити більш справедливе порівняння. Використовуючи SWF, втрачається в середньому 0,2 дБ SDR у порівнянні з MWF, але підхід змішування все ще краще, ніж підхід DeepNMF.

	Approach	SDR in dB					Comments
		Bass	Drums	Other	Vocals	Acco.	
Одно-канальні методи	BLEND (SWF)	2.76	3.93	3.37	5.13	11.53	$\lambda = 0.25$
	sNMF	-0.84	1.12	1.82	2.17	8.58	$Q = 25$
	dNMF	0.91	1.87	2.43	2.56	8.88	$Q = 25$
	DeepNMF	1.88	2.11	2.64	2.75	8.90	$Q = 25$
Багато-канальні методи	BLEND (MWF)	2.98	4.13	3.52	5.23	11.70	$\lambda = 0.25$
	NUG	2.72	3.89	3.18	4.55	10.29	

Таблиця 3. Порівняння на тестовій вибірці DSD100

Загалом, ми можемо зробити висновок, що системи DNN NUG і BLEND (MWF) працюють найкраще там, де, зокрема, система BLEND (MWF) досягає в середньому на 0,4 дБ кращого SDR, ніж NUG.

1.3 Глибока згорткова нейронна мережа U-Net

Розкладання музичного звукового сигналу на його вокал і компоненти фонової доріжки аналогічно трансляції зображення в зображення, де змішана спектрограма перетворюється на її складові джерела. Розглядається нове застосування архітектури U-Net — спочатку розробленої для медичної візуалізації — для завдання поділу джерела звуку, враховуючи його доведену здатність відтворювати дрібні деталі низького рівня, необхідні для високоякісного відтворення звуку.

Область пошуку музичної інформації (MIR) займається, серед іншого, аналізом музики в багатьох її аспектах, таких як мелодія, тембр або ритм. Серед цих аспектів популярна західна комерційна музика (“поп” музика), можливо, характеризується акцентом переважно на аспектах мелодії та акомпанементу; хоча це, безумовно, надмірне спрощення в контексті всього жанру. У роботі обмежується фокус до аналізу музики, яку добре описувати в термінах основної мелодійної лінії (передній план) і акомпанемент (фон). Зазвичай співається мелодія, тоді як акомпанемент виконує один або кілька інструменталістів; співак виконує текст, а бек музиканти забезпечують гармонію, а також жанрові та стильові репліки.

Завдання автоматичного поділу музичної композиції полягає в тому, щоб оцінити, як би співана мелодія та акомпанемент звучали окремо. Чистий голосовий сигнал корисний для інших пов’язаних завдань MIR, таких як ідентифікація співака та транскрипція лірики. Що стосується комерційних застосувань, то очевидно, що караоке-індустрія, яка оцінюється в мільярди доларів у всьому світі, отримає пряму користь від такої технології.

Розглядається адаптація архітектури U-Net до завдання поділу голосу. Архітектура була введена в біомедичну візуалізацію, щоб покращити точність і локалізацію мікроскопічних зображень нейронних структур. Архітектура побудована на основі повністю згорткової мережі і подібна до деконволюційної мережі. У деконволюційній мережі набір згорткових шарів — де кожен шар зменшує вдвічі розмір зображення, але подвоює кількість каналів — кодує зображення у невелике та глибоке уявлення. Потім це кодування декодується до вихідного розміру зображення за допомогою стека шарів підвищення дискретизації.

При відтворенні природного зображення зміщення лише на один піксель зазвичай не сприймається як серйозні спотворення. Однак у частотній області навіть незначний лінійний зсув у спектрограмі має катастрофічний вплив на сприйняття: це особливо актуально для музичних сигналів через логарифмічне сприйняття частоти; на додаток, зсув у вимірі часу може стати чутним у вигляді тремтіння та інших артефактів. Тому дуже важливо, щоб відтворення зберігало високий рівень деталізації. U-Net додає додаткові з’єднання між шарами на одному ієрархічному рівні в кодері та декодері. Це дозволяє інформації низького рівня надходити безпосередньо з входу високої роздільної здатності до виходу з високою роздільною здатністю.

1.3.1 Архітектура мережі

Мета архітектури нейронної мережі полягає в тому, щоб передбачити голосові та інструментальні компоненти її входу опосередковано: вихід кінцевого шару

декодера являє собою м’яку маску, яка поелементно перемножується на змішану спектрограму для отримання остаточної оцінки. На рис. 5 показано архітектуру мережі. Розглядається тренування двох окремих моделей для вилучення інструментальних і вокальних компонентів сигналу, щоб у майбутньому дозволити використовувати більш різні схеми навчання для двох моделей.

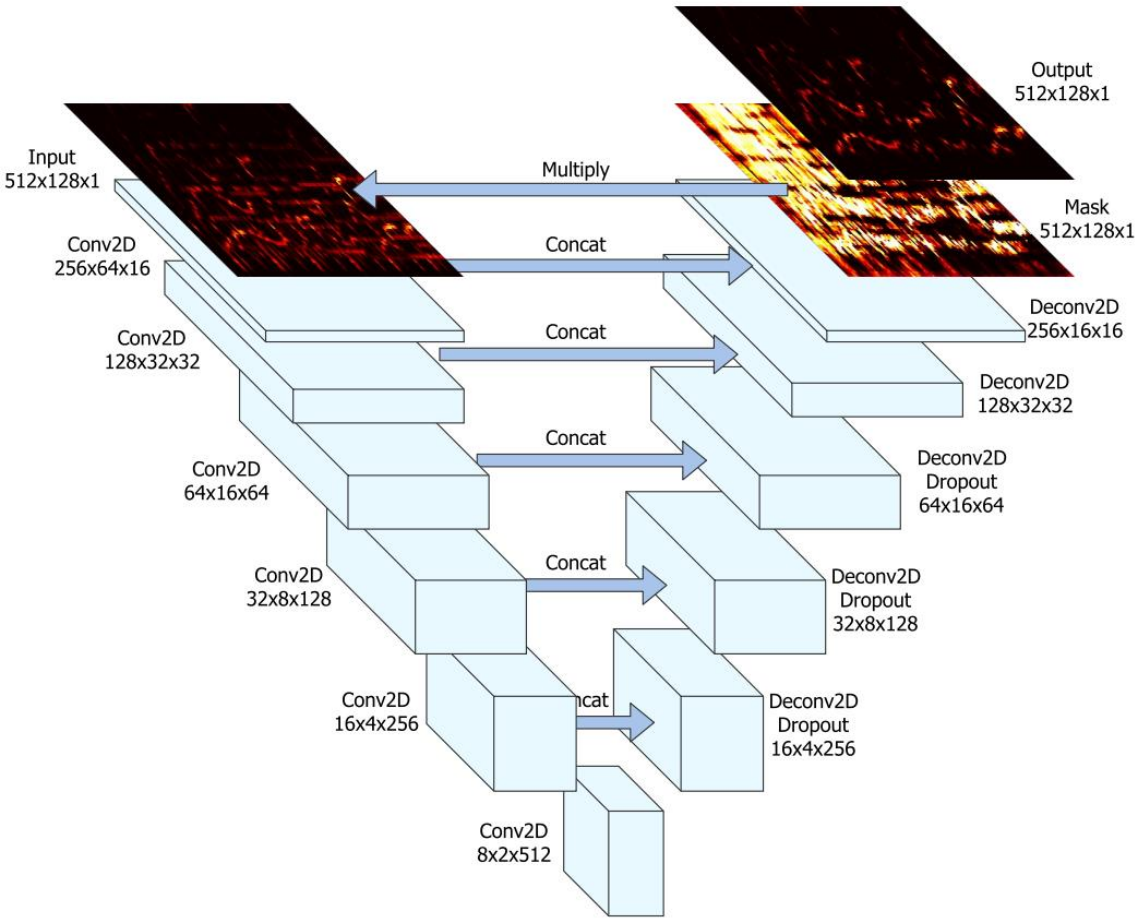


Рисунок 5. Архітектура нейронної мережі U-Net

Нехай X позначає величину спектрограми вихідного змішаного сигналу, тобто аудіо, що містить як вокальний, так і інструментальний компоненти. Нехай Y позначає величину спектрограм цільового аудіо; останнє відноситься або до вокального (Yv), або до інструментального (Yi) компонента вхідного сигналу.

Функція втрат, яка використовується для навчання моделі, є нормою $L_{1,1}$ різниці цільової спектрограми та замаскованої вхідної спектрограми:

$$L(X, Y ; \Theta) = ||f(X, \Theta) \odot X - Y ||_{1,1} \quad (6)$$

де f(X, Θ) – вихід моделі мережі, застосований до входу X з параметрами Θ – тобто маска, створена моделлю.

Норма $L_{1,1}$ матриці є просто сумою абсолютних значень її елементів.

Дві U-мережі, Θv і Θi, навчені передбачати вокальну та інструментальну спектрограмні маски відповідно.

У цій реалізації U-Net кожен шар кодера складається з стрибкоподібної двовимірної згортки кроку 2 і розміру ядра 5x5, пакетної нормалізації та нещільних випрямлених лінійних блоків (ReLU) з витоком 0,2. У декодері ми використовуємо стрибкоподібну деконволюцію (іноді її називають транспонованою згорткою) із кроком 2 і розміром ядра 5x5, пакетною нормалізацією, звичайним ReLU та використовуємо 50% випадання для перших трьох шарів. На останньому шарі ми використовуємо сигмоїдну функцію активації. Модель навчається за допомогою оптимізатора ADAM.

Враховуючи важкі обчислювальні вимоги навчання такої моделі, ми спочатку зменшуємо дискретизацію вхідного звуку до 8192 Гц, щоб прискорити обробку. Потім ми обчислюємо короткочасне перетворення Фур'є з розміром вікна 1024 і довжиною переходу 768 кадрів і витягуємо латки зі 128 кадрів (приблизно 11 секунд), які ми надаємо як вхідні дані та цілі в мережу. Спектрограми величини нормовані на діапазон [0, 1].

Модель нейронної мережі працює виключно на величині звукових спектрограм. Аудіосигнал для окремої (вокальної/інструментальної) компоненти відтворюється шляхом побудови спектрограми: вихідна величина задається шляхом застосування маски, передбаченої U-Net, до величини вихідного спектру, тоді як вихідна фаза є фазою вихідного спектру. Вихідний спектр, незмінний.

1.3.2 Навчальні дані

Як зазначено вище, опис архітектури моделі передбачає, що навчальні дані

були доступні у вигляді триплету (оригінальний сигнал, вокальний компонент, інструментальний компонент). Якщо хтось не знаходиться в надзвичайно щасливому положенні, що має доступ до величезної кількості незмішаних багатодоріжкових записів, потрібно знайти альтернативну стратегію, щоб навчити модель, подібну до описаної.

Розв'язання проблеми було знайдено шляхом використання певного, але великого набору комерційно доступних записів для «побудови» навчальних даних: інструментальних версій записів.

Нерідко артисти випускають інструментальні версії треків разом з оригінальним міксом. Ми використовуємо цей факт, витягуючи пари (оригінальних, інструментальних) треків із великої комерційної музичної бази даних. Кандидатів можна знайти шляхом перевірки метаданих на наявність треків із відповідною тривалістю та інформацією про виконавця, де назва треку (нечітко) збігається, за винятком рядка «Інструментальний», що зустрічається точно в одній назві в парі. Пул треків скорочується, виключаючи точні збіги вмісту.

Вищезазначений підхід забезпечує велике джерело пар спектрограм величин X (змішаних) і Yi (інструментальних). Спектрограма голосової величини Yv отримується з їх напівхвильової випрямленої різниці. Якісний аналіз великої кількості прикладів показав, що ця техніка створила досить ізольований вокал.

Остаточний набір даних містить приблизно 20 000 пар доріжок, що призводить до безперервного аудіо на майже два місяці. Наскільки відомо, це один із найбільших наборів навчальних даних, який коли-небудь застосовувався для поділу музичних джерел. У таблиці 4 показано відносний розподіл найпоширеніших жанрів у наборі даних, отриманий з метаданих каталогу.

Genre	Percentage
Pop	26.0%
Rap	21.3%
Dance & House	14.2%
Electronica	7.4%
R&B	3.9%
Rock	3.6%
Alternative	3.1%
Children's	2.5%
Metal	2.5%
Latin	2.3%
Indie Rock	2.2%
Other	10.9%

Таблиця 4. Відсоткове жанрове співвідношення навчальної вибірки

1.3.3 Оцінка роботи нейронної мережі

Розглядається порівняння моделі U-Net з моделлю Chimera, яка дала найвищі оцінки в кампанії MIREX Source Separation 2016 року. Слід зазначити, що на вебсервері Chimera працює покращена версія алгоритму, який брав участь у MIREX, з використанням гібридної архітектури «багато головок», яка поєднує глибоку кластеризацію зі звичайною нейронною мережею.

Для цілей оцінки будується додаткова базова модель; вона нагадує модель UNet, але без з'єднань пропуску, по суті створює згортковий кодер-декодер.

Розглядається оцінка трьох моделей на стандартному наборі даних iKala і MedleyDB. Набір даних iKala використовувався як стандартизована оцінка для щорічної кампанії MIREX протягом кількох років, тому існує наявних результатів, які можна використовувати для порівняння. З іншого боку, MedleyDB нещодавно був запропонований як високоякісний, комерційний набір багатоколійних стебел. Ми генеруємо ізольовані інструментальні та вокальні треки, зважуючи суми інструментальних/вокальних основ за їх відповідними коефіцієнтами мікшування, які надаються MedleyDB Python API. Оцінка обмежується кліпами, які, як відомо, містять вокал, використовуючи транскрипцію мелодії, надану в iKala і MedleyDB.

Для вимірювання продуктивності використовуються наступні функції:

відношення сигнал-спотворень (SDR), відношення сигнал-перешкода (SIR) і відношення сигнал-артефакт (SAR). Нормоване SDR (NSDR) визначається як

$$NSDR(Se, Sr, Sm) = SDR(Se, Sr) - SDR(Sm, Sr), \quad (7)$$

де Se — розрахунковий ізольований сигнал, Sr — еталонний ізольований сигнал, а Sm — змішаний сигнал. Показники продуктивності обчислюються за допомогою набору інструментів mir eval.

Таблиці 5 і 6 показують, що U-Net значно перевершує базову модель і Chimera за всіма трьома показниками продуктивності для обох наборів даних. На рис. 6 ми показуємо огляд розподілів для різних показників оцінки.

	U-Net	Baseline	Chimera
NSDR Vocal	**11.094**	8.549	8.749
NSDR Instrumental	**14.435**	10.906	11.626
SIR Vocal	**23.960**	20.402	21.301
SIR Instrumental	**21.832**	14.304	20.481
SAR Vocal	**17.715**	15.481	15.642
SAR Instrumental	**14.120**	12.002	11.539

Таблиця 5. Середні показники iKala

	U-Net	Baseline	Chimera
NSDR Vocal	**8.681**	7.877	6.793
NSDR Instrumental	**7.945**	6.370	5.477
SIR Vocal	**15.308**	14.336	12.382
SIR Instrumental	**21.975**	16.928	20.880
SAR Vocal	**11.301**	10.632	10.033
SAR Instrumental	**15.462**	15.332	12.530

Таблиця 6. Середні показники MedleyDB

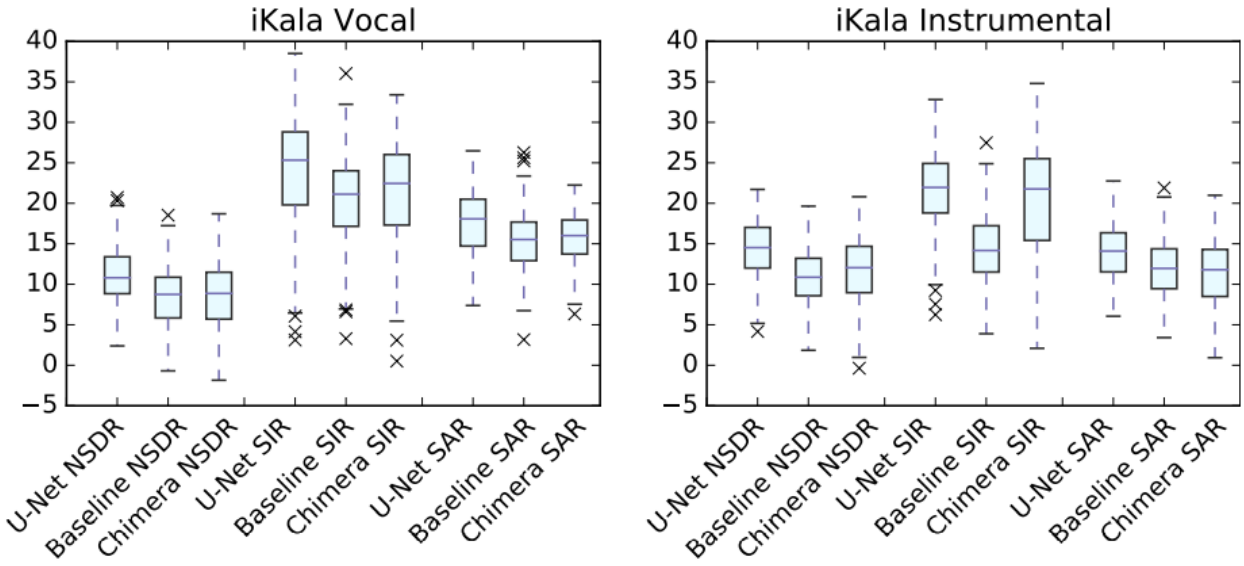


Рисунок 6. iKala - вокальні та інструментальні розподіли

Припускаючи, що розподіл треків у наборі утримування iKala, який використовується для оцінок MIREX, збігається з розподілом доріжок у загальнодоступному наборі iKala, ми можемо порівняти результати з учасниками завдання MIREX 2016 «Виокремлення голосу». Таблиці 7 і 8 показують оцінки NSDR для наших моделей у порівнянні з найкращими алгоритмами кампанії MIREX 2016 року.

Model	Mean	SD	Min	Max	Median
U-Net	**14.435**	3.583	4.165	21.716	**14.525**
Baseline	10.906	3.247	1.846	19.641	10.869
Chimera	11.626	4.151	-0.368	20.812	12.045
LCP2	11.188	3.626	2.508	19.875	11.000
LCP1	10.926	3.835	0.742	19.960	10.800
MC2	9.668	3.676	-7.875	22.734	9.900

Таблиця 7. iKala NSDR Instrumental, MIREX 2016

Model	Mean	SD	Min	Max	Median
U-Net	**11.094**	3.566	2.392	20.720	**10.804**
Baseline	8.549	3.428	-0.696	18.530	8.746
Chimera	8.749	4.001	-1.850	18.701	8.868
LCP2	6.341	3.370	-1.958	17.240	5.997
LCP1	6.073	3.462	-1.658	17.170	5.649
MC2	5.289	2.914	-1.302	12.571	4.945

Таблиця 8. iKala NSDR Vocal, MIREX 2016

Щоб оцінити ефект пропуску з’єднань U-Net, ми можемо візуалізувати маски, згенеровані U-Net і базовими моделями. З рис. 7 видно, що, хоча базова модель фіксує загальну структуру, спостерігається брак дрібнозернистих деталей.

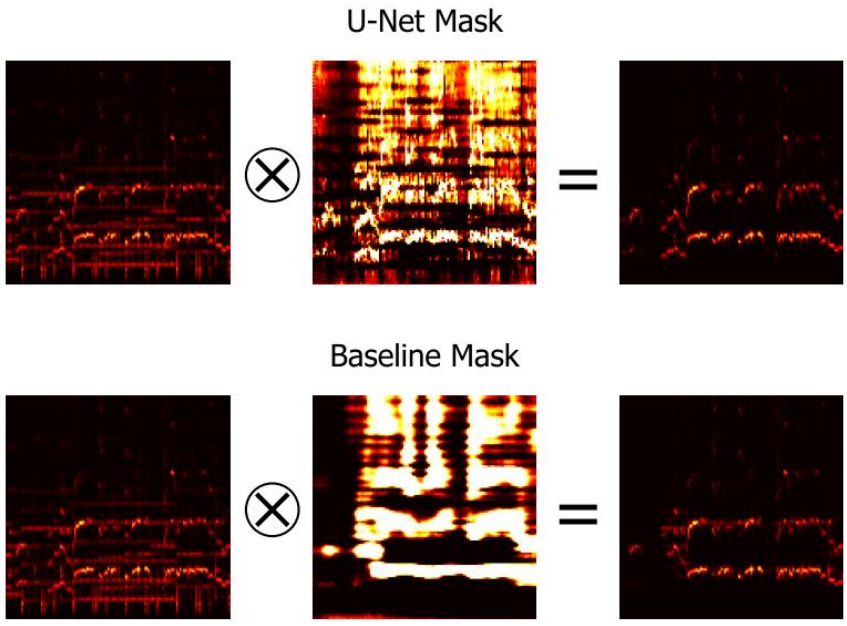


Рисунок 7. U-Net і базові маски

Було запроваджено протокол суб'єктивної оцінки алгоритмів поділу джерел. Пропонується поставити людям чотири запитання, які в цілому відповідають показникам SDR/SIR/SAR, а також додаткове запитання щодо загальної якості звуку.

Коли було задано ці чотири запитання суб’єктам без музичної підготовки, випробовувані вважали їх неоднозначними, наприклад, у них були проблеми з

розрізненням між відсутністю артефактів і загальною якістю звуку. Тож для кращої ясності було розділено опитування на наступні два питання у випадку вилучення голосу:

- Якість: «Оцініть якість голосу в прикладах нижче».
- Перешкоди: «Наскільки добре були вилучені інструменти у кліпі?»

Для інструментального вилучення були поставлені подібні запитання:

- Якість: «Оцініть якість звуку наведених нижче прикладів у порівнянні з оригіналом».
- Вилучення інструментів: «Оцініть, наскільки добре інструменти ізольовані в прикладах нижче щодо повного міксу вище».

Дані були зібрані за допомогою CrowdFlower, онлайн-платформи, де люди виконують мікрозавдання, такі як класифікація зображень, простий пошук в Інтернеті тощо.

В опитуванні користувачів CrowdFlower попросили прослухати три кліпи ізольованого аудіо, згенерованого U-Net, базової моделі та Chimera. Порядок трьох кліпів був рандомізований. Кожне запитання ставило одне з питань якості та перешкод. Відповіді давались за семиступінчатою шкалою Лайкерта, починаючи від «погано» до «ідеально».

Для опитування було використано 25 кліпів із набору даних iKala та 42 кліпи з MedleyDB. Для інструментального тесту було отримано 44 респонденти та 724 відповіді, а для голосового тесту 55 респондентів надали 779 відповідей.

На рис. 8 показано середнє значення та стандартне відхилення для відповідей, наданих на CrowdFlower. Алгоритм U-Net перевершує дві інші моделі з усіх питань.

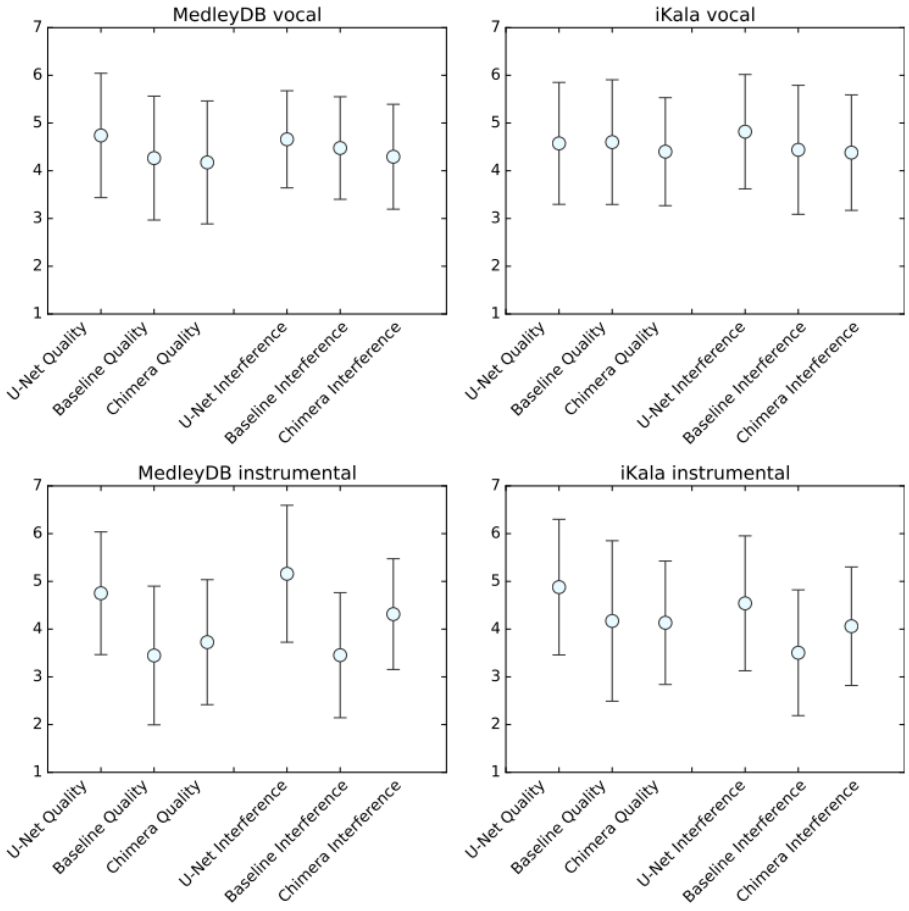


Рисунок 8. Результати оцінки CrowdFlower (середнє/стандартне значення)

РОЗДІЛ 2 ОГЛЯД СИСТЕМ ДЛЯ РОЗДІЛЕННЯ ДЖЕРЕЛА ЗВУКУ

2.1 Параметри для оцінки якості розділення джерела звуку

У цьому розділі розглядається оцінку алгоритмів сліпого розділення джерел звуку (BASS). Залежно від конкретного застосування можуть бути дозволені різні спотворення між отриманим внаслідок роботи алгоритму джерелом і бажаним справжнім джерелом. Розглядаються чотири різні набори таких дозволених спотворень, від незмінних у часі посилень до змінних у часі фільтрів. У кожному випадку ми порівнюємо отримане джерело з оригінальним плюс терміни помилки, що відповідають перешкодам, адитивному шуму та алгоритмічним артефактам. Потім отримується глобальний показник продуктивності, використовуючи коефіцієнт енергії, плюс окремий показник продуктивності для кожного члена помилки.

Сліпе розділення джерел звуку (BASS) було темою інтенсивної роботи протягом останніх років. З'явилося кілька успішних моделей, таких як незалежний компонентний аналіз (ICA), розріджені декомпозиції (SD) і обчислювальний аналіз слухових сцен (CASA). Проте все ще важко оцінити алгоритм або порівняти кілька алгоритмів через відсутність відповідних показників продуктивності та загальних тестових звуків, навіть у дуже простому випадку лінійних миттєвих сумішей. Тому було розроблено певні числові критерії продуктивності, які можуть допомогти оцінити та порівняти алгоритми при застосуванні до звичайних задач BASS. Перш ніж представити їх, спочатку будуть описані наявні показники ефективності та їх недоліки.

Проблема BASS виникає, коли один або кілька мікрофонів записують звук, який є сумішшю звуків, що надходять з кількох джерел. Для простоти буде розглянуто лише лінійні інваріантні в часі системи змішування. Якщо через $s_j(t)$ позначити сигнал, що випромінюється j-им джерелом ($1 \leq j \leq n$), $x_i(t)$ – сигнал, записаний i-м мікрофоном ($1 \leq i \leq m$) і $a_{ij}(\tau)$ причинно-наслідкові фільтри від джерела до мікрофона, ми маємо $x_i(t) = \sum_{j=1}^{n} \sum_{\tau=0}^{+\infty} a_{ij}(\tau) s_j(t-\tau) + n_i(t)$, де

$n_i(t)$ є деяким адитивним датчиком шуму. Цей вираз m × n зручніше виразити за допомогою матриці фільтрів формалізму як

$$\mathbf{x} = \mathbf{A} * \mathbf{s} + \mathbf{n} \qquad (8)$$

де операція * позначає згортку. Далі змінні без індексу часу будуть позначати пакетні послідовності, напр. **x** = [x(0), ... , x(T−1)]. Жирні літери будуть використовуватися для багатоканальних змінних, таких як вектор спостережень **x**, вектор джерел **s** або система змішування **A**, а прості літери для одноканалових змінних, таких як j-е джерело s_j.

BASS охоплює багато програм, і критерії, які використовуються для оцінки продуктивності алгоритму, залежать від програми. Іноді метою є вилучення вихідних сигналів, які прослуховуються, відразу після поділу або після певної обробки аудіо. Іноді для опису складних звукових сцен, пов'язаних із людським слухом, потрібно отримати характеристики джерела та/або параметри змішування. Тут розглядається найбільш поширена задача, яка вирішується алгоритмами BASS: вилучення джерела.

Вилучення джерела полягає у виділенні з аудіо композиції одного або кількох моноджерельних сигналів s_j. Приклади включають шуми та деверберацію мовлення для слухових протезів та виділення цікавих звуків у музичних уривках для створення електронної музики. Без конкретної попередньої інформації про джерела або систему змішування. А ця проблема страждає від добре відомих теоретичних невизначеностей. Як правило, джерела можуть бути відновлені лише до перестановки та довільного підсилення, але додаткові невизначеності можуть існувати у згорнутих сумішах (наприклад, до застосування фільтра).

Вилучення джерела може бути розглянуто на різних рівнях складності залежно від структури системи змішування. Першим критерієм складності є відповідна кількість джерел і датчиків. У безшумних визначених миттєвих сумішах (тобто, коли m = n) існує незмінна в часі лінійна система розмішування $\mathbf{W} = \mathbf{A}^{-1}$. Після того як **W** було оцінено, джерела можна просто відновити як **s** = **Wx**. У безшумних недостатньо

визначених сумішах (тобто коли m < n) це вже неможливо, оскільки рівняння **x = As** має афінний (дозвільний або що зберігає паралельні відносини) набір розв'язків. Цю нетривіальну невизначеність можна усунути, використовуючи знання про джерела, такі як розріджені априори. Другим критерієм складності є довжина змішувальних фільтрів. Багато алгоритмів для миттєвих безшумних визначених сумішей забезпечують майже ідеальні результати. Але згорнуті суміші все ще викликають складні теоретичні питання, такі як ідентифікація джерел аж до виграшу, і технічні труднощі, як-от оцінка довготривалих фільтрів змішування в короткотривалих сумішах.

Перший тип міри припускає, що оцінені джерела $\hat{s}$ були відновлені шляхом застосування незмінної в часі лінійної системи змішування **W** до спостережень **x**. Глобальна система **B = W * A** перевіряє $\hat{s}$ = **B * s**. Якість $\hat{s}_j$ потім вимірюється за допомогою міри міжсимвольної перешкоди (Inter-Symbol Interference)

$$ISI_j := \frac{\sum_{j',\tau} \left|B_{jj'}(\tau)\right|^2 - max_{j',\tau}\left|B_{jj'}(\tau)\right|^2}{max_{j',\tau}\left|B_{jj'}(\tau)\right|^2}. \quad (9)$$

ISI_j завжди додатний і дорівнює нулю лише тоді, коли $\hat{s}_j$ дорівнює справжньому джерелу s_j, аж до підсилення та затримки τ з (j', τ) = arg $max\left|B_{jj'}(\tau)\right|^2$. Цей критерій та інші подібні критерії є релевантними, але не можуть бути застосовані до недостатньо визначених проблем BASS, оскільки досконала інваріантна в часі система демікшування **W** взагалі не існує. Щобільше, навіть у визначеному BASS можна використовувати інші схеми поділу, ніж лінійне демікшування, незмінне в часі. Другий вид міри полягає в прямому порівнянні $\hat{s}_j$ й s_j, звертаючи увагу на невизначеність завдання. Невизначеність підсилення можна обробити шляхом порівняння L_2-нормованих версій джерел з відносною квадратною відстанню

$$D := min_{\varepsilon=\pm 1} \left\| \frac{\hat{s}_j}{\|\hat{s}_j\|} - \varepsilon \frac{s_j}{\|s_j\|} \right\|^2. \quad (10)$$

Ця міра також актуальна, оскільки вона завжди додатна і дорівнює нулю лише тоді,

коли $\hat{s}_j$ дорівнює s_j аж до підсилення. Однак D приймає найбільше значення D = 2, навіть у гіршому випадку, коли невизначеність перестановки була погано вирішена і де $\hat{s}_j$ дорівнює іншому джерелу $s_{j'}$, ортогональному до s_j. Тоді можна було б отримати спотворення D = +∞. Загалом D оцінює погані результати досить грубо. Наприклад, $\hat{s}_j = s_{j'}$ і $\hat{s}_j = s_{j'} / \|s_{j'}\| + 0{,}02\, s_j / \|s_j\|$ призводять до подібних показників D = 2 і D ≈ 1,96, але сприймаються зовсім по-різному.

Ці показники ефективності страждають від додаткових обмежень. Обидва розглядають лише випадок, коли $\hat{s}_j$ потрібно відновити до перестановки та підсилення. Але в деяких програмах може бути доречним дозволити більші чи менші викривлення, не обов'язково пов'язані з теоретичною невизначеністю проблеми. Наприклад, у музичних програмах hi-fi може бути важливим відновити джерела до простого підсилення, оскільки довільна фільтрація змінює тембр музичних інструментів. І навпаки, у мовних програмах можуть бути допущені деякі фільтраційні спотворення, оскільки мовлення з фільтруванням низьких частот, як правило, все ще зрозуміле. Крім того, обидва показники забезпечують єдиний критерій ефективності, що містить усі помилки оцінки. Але в аудіододатках важливо окремо вимірювати кількість перешкод від непотрібних джерел, кількість або залишковий шум датчика та кількість артефактів «бурління» (також званих «музичним шумом»). Такі артефакти часто вважаються більш дратівливими помилками, ніж перешкоди, які самі по собі дратують більше, ніж шум датчика. Багато методів поділу для недостатньо визначених проблем BASS створюють мало перешкод, але багато артефактів, і це не можна описати одним критерієм.

Тож були розроблені нові критерії продуктивності, які можна застосувати до всіх звичайних проблем BASS і подолати наведені вище обмеження. Єдині припущення, які робляться, це

- справжні сигнали джерела та сигнали шуму (якщо такі є) відомі,
- користувач вибирає сімейство дозволених викривлень F відповідно до програми (але незалежно від типу аудіо суміші або використаного

алгоритму).

Систему змішування та техніку розділення знати не обов’язково.

Окремі показники продуктивності обчислюються для кожного оціненого джерела $\hat{s}_j$ шляхом порівняння його з наданим справжнім джерелом s_j. Потрібно зауважити, що заходи не враховують невизначеність перестановки BASS. Якщо необхідно, $\hat{s}_j$ можна порівняти з усіма джерелами $(s_{j\prime})_{1\leq j\prime\leq n}$ і вибрати «справжнє джерело» як те, що дає найкращі результати.

Розрахунок критеріїв складається з двох послідовних кроків. На першому кроці ми розкладаємо $\hat{s}_j$ як

$$\hat{s}_j = s_{target} + e_{interf} + e_{noise} + e_{artif}, \qquad (11)$$

де $s_{target} = f(s_j)$ є версією s_j, модифікованою дозволеним спотворенням $f \in F$, і де e_{interf}, e_{noise} та e_{artif} відповідно є умовами перешкод, шумів та артефактів. Ці чотири терміни мають представляти частину $\hat{s}_j$, яка сприймається як отримана з об’єднання потрібного джерела s_j, з інших небажаних джерел $(s_{j\prime})_{j\prime\neq j}$, з шумів датчика $(n_i)_{1\leq i\leq m}$ та з інших факторів (наприклад, заборонених спотворень джерел та/або артефактів, що «бурчать»). На другому кроці обчислюються коефіцієнти енергії, щоб оцінити відносну кількість кожного з цих чотирьох членів або на всій тривалості сигналу, або на локальних кадрах.

Починаючи з розкладання $\hat{s}_j$, тепер можна визначити числові критерії продуктивності, обчислюючи коефіцієнти енергії, виражені в децибелах (дБ). Визначаються співвідношення джерела та спотворення

$$SDR := 10log_{10}\frac{\|s_{target}\|^2}{\|e_{interf}+e_{noise}+e_{artif}\|^2}, \qquad (12)$$

відношення джерела до перешкод

$$SIR := 10log_{10}\frac{\|s_{target}\|^2}{\|e_{interf}\|^2}, \qquad (13)$$

співвідношення джерела до шуму

$$SNR := 10log_{10}\frac{\|s_{target}+e_{interf}\|^2}{\|e_{noise}\|^2}, \quad (14)$$

і співвідношення джерел до артефактів

$$SAR := 10log_{10}\frac{\|s_{target}+e_{interf}+e_{noise}\|^2}{\|e_{artif}\|^2}. \quad (15)$$

2.2 Бібліотека для розділення джерела звуку Spleeter

Бібліотека Spleeter була реалізована для того, щоб дослідники в галузі пошуку музичної інформації (MIR) мали можливість використовувати потужність найсучаснішого алгоритму розділення джерела звуку. Реалізована ця бібліотека мовою програмування Python, а її основою є використання відомої бібліотеки для роботи та створення засобів машинного навчання Tensorflow, з попередньо підготовленими моделями для поділу на 2 (вокал та музичний супровід), 4 (вокал, ударні інструменти, бас та інші) і 5 (вокал, ударні інструменти, бас, піаніно та інші) джерел.

В цілях відстеження якості алгоритмів розділення джерела звуку, люди порівнювали свої алгоритми у міжнародних оцінювальних кампаніях. Тому було виявлено, що продуктивність Spleeter відповідає показникам найкращих алгоритмів.

В додаток до цього, алгоритм, що використовує бібліотека Spleeter, є дуже швидким. Якщо використовувати версію GPU, розділення може відбуватись в 100 разів швидше, ніж у режимі реального часу, що робить цю систему хорошим варіантом для обробки великих наборів даних.

Spleeter може використовуватись дослідниками, які працюють над пошуком музичної інформації. Ця бібліотека має ліцензію MIT, тому може вільно використовуватись.

Під капотом Spleeter є досить складним програмним продуктом, але ця бібліотека досить проста у використанні. Фактичного поділу можна досягти за допомогою одного командного рядка, і це повинно працювати на ноутбуці чи комп’ютері незалежно від операційної системи. Для більш просунутих користувачів

є клас API Python під назвою Separator, яким ви можете маніпулювати безпосередньо у своєму коді.

Отже, саме ця бібліотека була використана у подальшій роботі як інструмент для поділу джерел музики з попередньо підготовленими моделями під назвою, оскільки була розроблена з урахуванням простоти використання, продуктивності поділу та швидкості. Spleeter заснований на Tensorflow і дає можливість:

- розділяти аудіофайли на 2, 4 або 5 джерел, використовуючи попередньо навчені моделі,
- тренувати моделі поділу джерел або точно налаштувати попередньо навчені за допомогою Tensorflow (за умови, що у вас є набір даних ізольованих джерел)

Ціллю створення бібліотеки Spleeter з попередньо підготовленими найсучаснішими моделями є використання сили поділу джерела в різних завданнях, таких як, аналіз вокальних текстів із аудіо (вирівнювання аудіо/текстів, транскрипція тексту...), транскрипція музики (транскрипція акордів, транскрипція барабанів, транскрипція баса, оцінка акордів, відстеження ударів), ідентифікація співака, будь-який тип класифікації кількох міток (настрій/жанр) або виділення вокальної мелодії. На сьогодні розділення джерел досягло рівня зрілості, що робить його гідним уваги для цих завдань, і що конкретні функції, розраховані з ізольованого вокалу, ударних або басу, можуть допомогти підвищити продуктивність, особливо в сценаріях низької доступності даних (невеликі набори даних, обмежена доступність анотацій), для яких навчання маючи попередньо правильні дані для перевірки може бути складним. Spleeter також дає можливість точно налаштувати надані найсучасніші моделі, щоб адаптувати систему до конкретного випадку використання. Нарешті, наявність доступного інструменту поділу джерел, такого як Spleeter, дозволить дослідникам порівнювати продуктивність своїх нових моделей з найсучаснішими у своїх власних приватних наборах даних замість musdb18, який зазвичай є єдиним набором даних для звітів про розділення.

Попередньо навченими моделями нейронні мережі типу U-Net. Використовувались 12-шарові мережі U-Net (6 шарів для кодера і 6 для декодера). U-Net використовується для оцінки м'якої маски для кожного джерела (стебла). Втрата при навчанні є L_1-нормою між замаскованими спектрограмами вхідного змішування та вихідними цільовими спектрограмами. Моделі навчалися на внутрішніх наборах даних інтернет-сервісу Deezer, що є розробником бібліотеки Spleeter (зокрема, на наборі даних Bean) за допомогою оптимізатора Adam. Час навчання зайняв приблизно цілий тиждень на одному GPU. Потім виконується поділ із розрахункових спектрограм джерела за допомогою м'якого маскування або багатоканальної фільтрації Вінера.

Навчання та висновки реалізовані в Tensorflow, що дає можливість запускати код на центральному процесорному блоці (ЦП) або графічному процесорі.

Оскільки весь процес поділу можна запустити на графічному процесорі, а модель заснована на CNN (що робить розпаралелювання обчислень дуже ефективним), моделі працює дуже швидко. Наприклад, Spleeter може розділити весь набір тестових даних musdb18 (приблизно 3 години 27 хвилин аудіо) на 4 джерела менш ніж за 2 хвилини, включаючи час завантаження моделі (близько 15 секунд) і експорт аудіофайлів wav, використовуючи один графічний процесор GeForce RTX 2080 і подвійний процесор Intel Xeon Gold 6134 на частоті 3,20 ГГц (ЦП використовується лише для завантаження нерозділених файлів і експорту основних файлів). Також Spleeter може розділити на 4 джерела 100 секунд стереоаудіо менш ніж за 1 секунду, що робить його дуже корисним для ефективної обробки великих наборів даних.

Моделі конкурують із найсучаснішими стандартними наборами даних musdb18, хоча вони не були навчені, перевірені чи оптимізовані будь-яким чином з даними musdb18. Повідомлено результати з точки зору стандартних показників поділу джерела, а саме відношення сигналу до спотворення (SDR), відношення сигналу до артефактів (SAR), відношення сигналу до перешкод (SIR) і відношення

вихідного зображення та просторового спотворення (ISR), представлені в таблиці 9 у порівнянні з Open-Unmix, яка виконує майже одні з найсучасніших результатів відтворення розділення джерела звуку. Представлено результати для м'якого маскування та багатоканальної фільтрації Вінера. Як бачимо, за більшістю показників Spleeter є конкурентоспроможним з Open-Unmix і особливо з SDR для всіх інструментів.

	vocals				bass				drums				other			
	SDR	SIR	SAR	ISR	SDR	SIR	SAR	ISR	SDR	SIR	SAR	ISR	SDR	SIR	SAR	ISR
Spleeter Mask	6.55	15.19	6.44	**12.01**	5.10	10.01	5.15	9.18	5.93	12.24	5.78	10.50	4.24	7.86	4.63	9.83
Spleeter MWF	**6.86**	**15.86**	**6.99**	11.95	**5.51**	10.30	5.96	**9.61**	**6.71**	**13.67**	**6.54**	**10.69**	**4.55**	**8.16**	**4.88**	**9.87**
Open-Unmix	6.32	13.33	6.52	11.93	5.23	**10.93**	**6.34**	9.23	5.73	11.12	6.02	10.51	4.02	6.59	4.74	9.31

Таблиця 9. Результати поділу на 4 джерела (жирний шрифт позначає найвищі значення показників).

Spleeter доступний на github з дозвільною ліцензією. Це сховище, можливо, використовуватиметься для випуску інших моделей з покращеною продуктивністю або моделей, що розділятимуться на більше ніж 5 джерел у майбутньому.

РОЗДІЛ 3 АВТОМАТИЧНА ТРАНСКРИПЦІЯ МУЗИКИ

Можливість транскрибувати музичне аудіо в нотні записи є захопливим прикладом людського інтелекту. Воно включає сприйняття (аналіз складних слухових сцен), пізнання (розпізнавання музичних об’єктів), представлення знань (формування музичних структур) та умовивід (перевірка альтернативних гіпотез). Автоматична транскрипція музики (АТМ), тобто розробка обчислювальних алгоритмів для перетворення акустичних музичних сигналів у певну форму нотних записів, є складним завданням в обробці сигналів і штучному інтелекті. Воно містить кілька підзадач, включаючи (багато)ступеневу оцінку звуку, виявлення початку та зсуву, розпізнавання інструментів, відстеження ритму та темпу, інтерпретацію виразного часу та динаміки, а також набір партитур. Враховуючи кількість підзадач, які воно включає, і широкий діапазон його застосування, це вважається фундаментальною проблемою в області обробки музичних сигналів і пошуку музичної інформації (MIR). Через саму природу музичних сигналів, які часто містять кілька джерел звуку (наприклад, музичні інструменти, голос), які створюють одну або кілька одночасних звукових подій (наприклад, ноти, ударні звуки), які мають високу кореляцію як у часі, так і частоти, АТМ досі вважається складною та відкритою проблемою в літературі, особливо для музики, що містить кілька одночасних нот і кілька інструментів.

Типові представлення даних, що використовуються в системі АМТ, проілюстровані на рис. 9. Зазвичай система АТМ приймає форму звукового сигналу як вхідний сигнал (рис. 9а), обчислює частотно-часове представлення (рис. 9б) і виводить подання звукових тонів з плином часу (фортепіанний рулон) (рис. 9в) або набірну партитуру (рис. 9г).

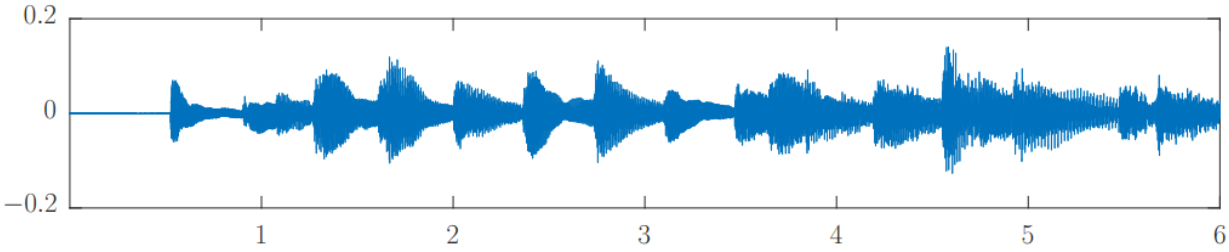


Рисунок 9а. Амплітуда

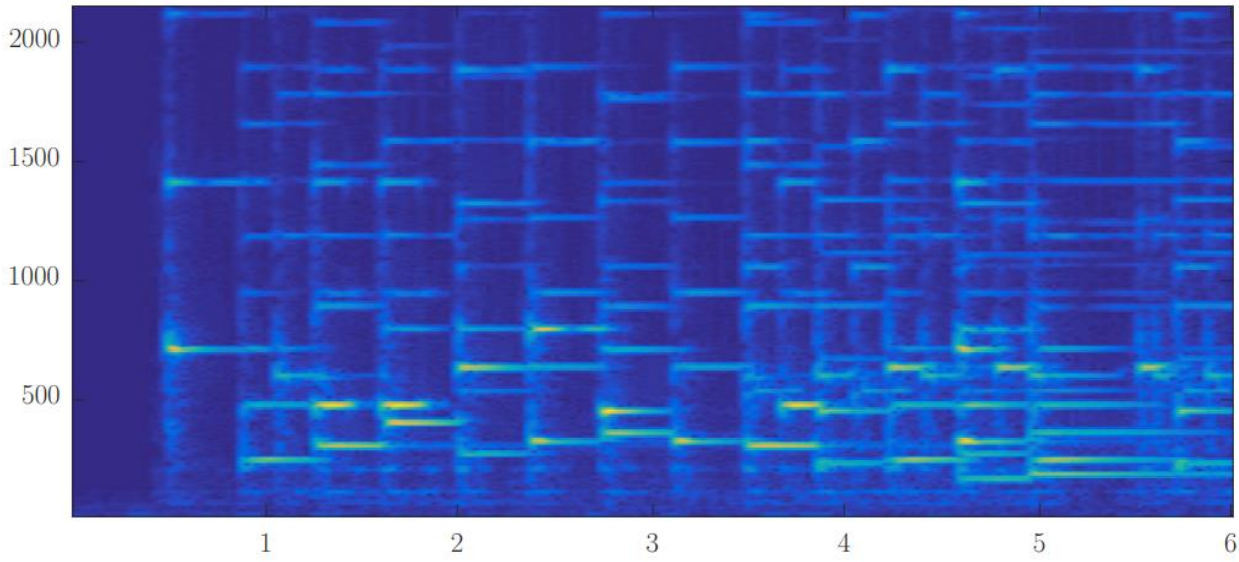


Рисунок 9б. Частота

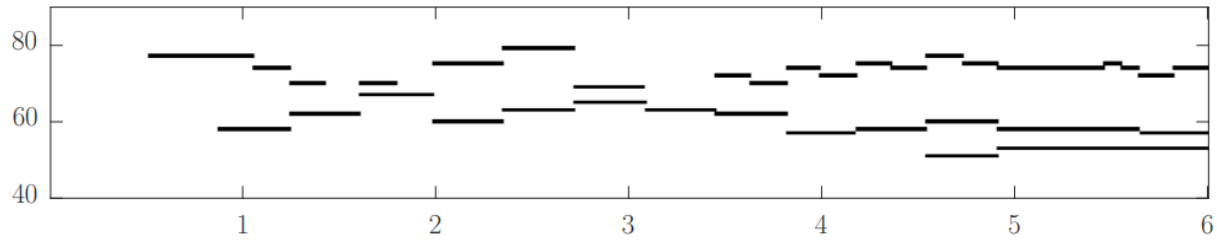


Рисунок 9в. Подання звукових тонів з плином часу

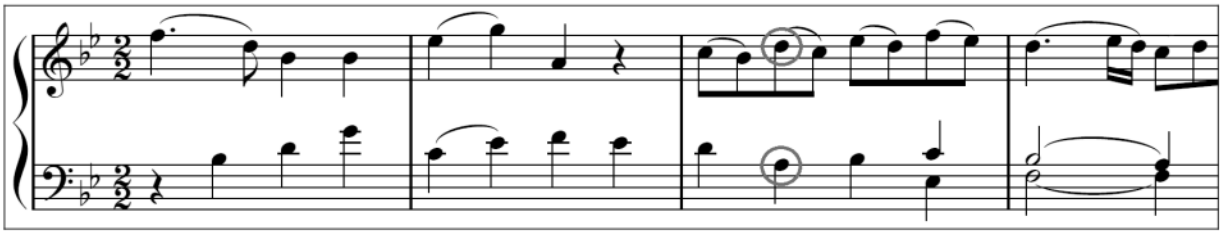


Рисунок 9г. Партитура

У цій роботі розглядається транскрибування поліфонічної музики, створеної звуковими інструментами та голосом.

АТМ тісно пов'язане з іншими завданнями обробки музичних сигналів, такими як розділення джерела звуку, яке також включає оцінку та висновок вихідних сигналів зі спостережень за сумішшю. Це також корисно для багатьох завдань високого рівня в MIR, таких як структурна сегментація, виявлення кавер-пісень та оцінка подібності музики, оскільки ці завдання набагато легше вирішувати, коли музичні ноти відомі. Таким чином, АТМ забезпечує основний зв’язок між сферами обробки музичних сигналів і символьної обробки музики (тобто обробка нотних записів і моделювання музичної мови).

АТМ має тісні зв'язки з іншими проблемами обробки сигналів. Стосовно області обробки мовлення, АТМ широко вважається музичним еквівалентом автоматичного розпізнавання мовлення (АРМ), у тому сенсі, що обидва завдання передбачають перетворення акустичних сигналів у символічні послідовності. Як і проблема «коктейльної вечірки» в мовленні, музика зазвичай включає кілька одночасних голосів, але, на відміну від мовлення, ці голоси сильно корелюють за часом і частотою. Крім того, системи АТМ і АРМ отримують переваги від компонентів мовного моделювання, які поєднуються з акустичними компонентами для отримання правдоподібних результатів. Таким чином, існують чіткі зв’язки між АТМ та ширшою сферою обробки природної мови (ОПМ), причому музика має свої власні граматичні правила або статистичні закономірності, подібно до природної мови.

У новітній галузі аналізу звукової сцени існує пряма аналогія між АТМ та виявленням звукових подій (SED), зокрема з поліфонічним SED, який передбачає виявлення та класифікацію множинних подій, що перекриваються, зі звуку. Хоча повсякденні та природні звуки не демонструють такого ж ступеню тимчасової закономірності та залежності від частоти між джерелами, як у музичних сигналах, існує тісна взаємодія між двома проблемами з точки зору використовуваних методологій.

Крім того, АТМ пов'язаний з обробкою зображень і комп'ютерним баченням,

оскільки музичні об'єкти, такі як ноти, можуть бути розпізнані як двовимірні моделі в частотно-часових представленнях. У порівнянні з обробкою зображень і комп’ютерним баченням, де оклюзія є поширеною проблемою, на системи АТМ часто впливають музичні об’єкти, що займають однакові частотно-часові області.

У порівнянні з іншими проблемами в області обробки музичних сигналів або ширшої дисципліни обробки сигналів, існує кілька факторів, які роблять АТМ особливо складним:

- Поліфонічна музика містить поєднання кількох одночасних джерел (наприклад, інструментів, вокалу) з різною висотою, гучністю та тембром (якістю звуку), причому кожне джерело створює один або кілька музичних голосів. Визначення музичних атрибутів (наприклад, висоти) із сигналу суміші є надзвичайно недостатньо визначеною проблемою.
- Накладаючись звукові події часто демонструють гармонійні відносини один з одним; для будь-якого приголосного музичного інтервалу основні частоти утворюють невеликі цілі співвідношення, так що їх гармоніки перекриваються за частотою, що робить розділення голосів ще складнішим. Взявши як приклад акорд до-мажор, співвідношення основних частот його трьох нот C:E:G становить 4:5:6, а відсоток гармонійних позицій, які перекриваються іншими нотами, становить 46,7%, 33,3% і 60 % для C, E та G відповідно.
- Хронометраж музичних голосів регулюється регулярною метричною структурою музики. Зокрема, музиканти приділяють пильну увагу синхронізації початку та зміщення між різними голосами, що порушує загальне припущення про статистичну незалежність між джерелами, що в іншому випадку полегшує розділення.
- Анотація транскрипцій на основі правди для поліфонічної музики займає дуже багато часу і вимагає високого досвіду. Відсутність таких анотацій

обмежила використання потужних методів навчання під керівництвом певних підпроблем АТМ, таких як транскрипція піаніно, де анотація може бути автоматизована завдяки певним моделям піаніно, які можуть автоматично фіксувати дані про виконання. Був запропонований підхід, який вимагає професійних музичних виконавців і ретельної попередньої та постобробки партитур. Ноти зазвичай не забезпечують хорошої анотації правди для АТМ; вона не вирівняна за часом до аудіосигналу і зазвичай не забезпечує точного представлення виконання. Навіть якщо існують точні транскрипції, непросто визначити відповідні пари аудіофайлів і партитур, оскільки існує безліч версій будь-якого даного музичного твору, доступних у розповсюджувачів музики. У кращому випадку ноти можна розглядати як слабкі етикетки.

Наведені вище ключові проблеми часто не вирішуються повністю в сучасних системах АТМ, що призводить до поширених проблем на виходах АТМ, таких як октавні помилки, помилки півтону, пропущені ноти (зокрема при наявності щільних акордів), додаткові ноти (часто проявляються як гармонічні помилки за наявності невидимих тембрів), об’єднані чи фрагментовані ноти, неправильне визначення початку/зміщення або неправильно призначені потоки.

3.1 Методи АТМ

За останні чотири десятиліття було розроблено багато підходів до АТМ для поліфонічної музики. Хоча кінцевою метою АТМ є перетворення акустичного музичного запису в певну форму нотного запису, більшість підходів були розроблені для досягнення певної проміжної мети. Залежно від рівня абстракції та структур, які необхідно моделювати для досягнення таких цілей, підходи АТМ загалом можна розділити на чотири категорії: на рівні кадру, на рівні ноти, на рівні потоку та на рівні нотації.

Транскрипція на рівні кадру, або оцінка на основі декількох тонів (MPE), — це

оцінка кількості та висоти нот, які одночасно присутні в кожному часовому кадрі (10 мс). Зазвичай це виконується в кожному кадрі незалежно, хоча контекстна інформація іноді розглядається за допомогою фільтрації оцінок висоти кадру на етапі постобробки. На рис. 10 (верхній графік) показано приклад транскрипції на рівні кадру, де кожна чорна точка є оцінкою висоти. Методи цієї категорії не формують поняття музичних нот і рідко моделюють будь-які музичні структури високого рівня. Значна частина наявних підходів АТМ працює на цьому рівні. Останні підходи включають традиційні методи обробки сигналів, імовірнісне моделювання, баєсівські підходи, факторизацію невід'ємної матриці (NMF), і нейронні мережі. Усі ці методи мають як плюси, так і мінуси, і дослідження не звелися до єдиного підходу. Наприклад, традиційні методи обробки сигналів прості та швидкі та краще узагальнюються для різних інструментів, тоді як методи глибокої нейронної мережі зазвичай досягають вищої точності на конкретних інструментах (наприклад, фортепіано). Баєсівські підходи забезпечують комплексне моделювання процесу генерації звуку, однак моделі можуть бути дуже складними та повільними.

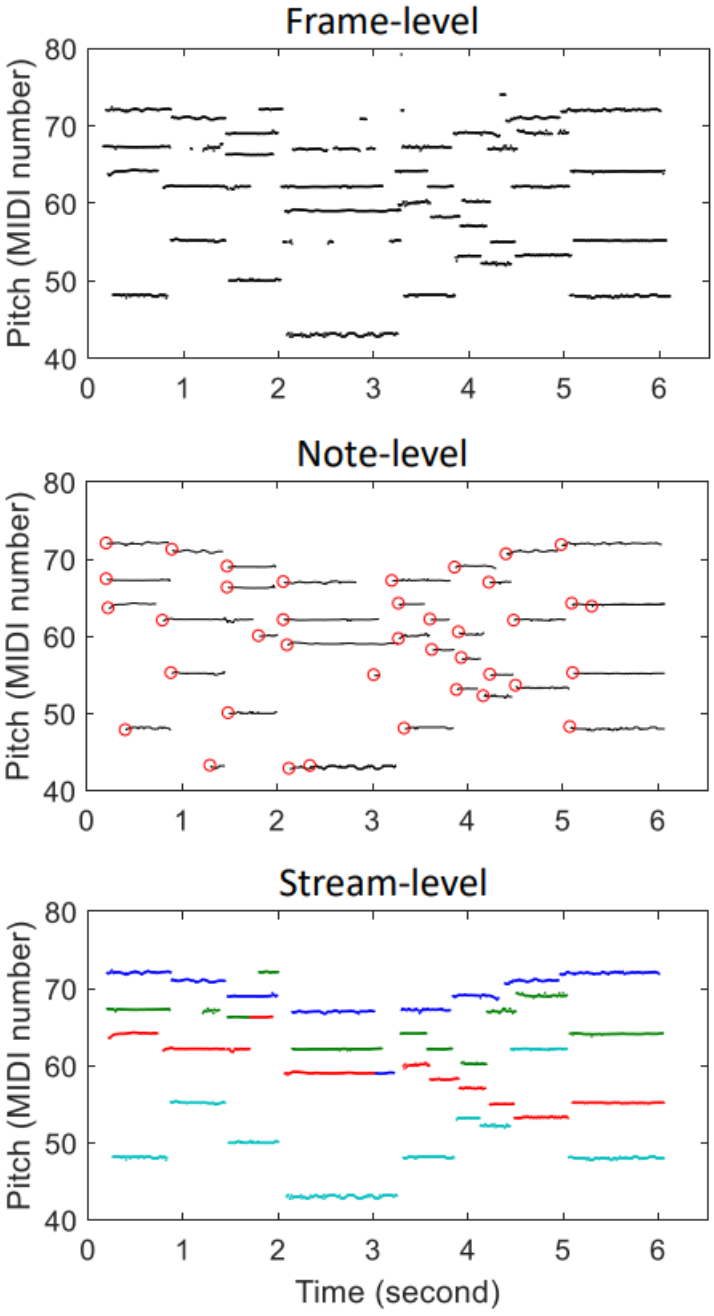


Рисунок 10. Приклади транскрипцій на рівні кадру, ноти та потоку відповідно

Транскрипція на рівні ноти, або відстеження нот, на один рівень вище, ніж MPE, з точки зору великої кількості структур оцінок. Він не тільки оцінює висоту звуку в кожному часовому кадрі, але також з'єднує оцінки висоти звуку з часом у ноти. У літературі про АТМ музична нота часто характеризується трьома елементами: висота, час початку та час зсуву. Оскільки зміщення нот може бути

неоднозначним, ними іноді нехтують при оцінці підходів відстеження нот, і тому деякі підходи відстеження нот лише оцінюють висоту звуку та час початку нот. Рис. 10 (посередині) показує приклад транскрипції на рівні ноти, де кожна нота показана у вигляді червоного кола (початок), за яким слідує чорна лінія (контур висоти). Багато підходів відстеження нот формують ноти шляхом постобробки вихідних даних MPE (тобто оцінки висоти звуку в окремих кадрах). Методики, які використовувалися в цьому контексті, включають медіанну фільтрацію, приховані моделі Маркова та нейронні мережі. Ця постобробка часто виконується для кожного MIDI-тону незалежно без урахування взаємодії між одночасними нотами. Це часто призводить до помилкових або відсутніх нот, які мають спільні гармоніки з правильно оціненими нотами. Було запропоновано деякі підходи для розгляду нотної взаємодії через спектральну модель правдоподібності або модель мови музики. Інша підгрупа підходів оцінює ноти безпосередньо з аудіосигналу, а не на основі виходів MPE. Деякі підходи спочатку визначають початок, а потім оцінюють висоту звуку в межах кожного інтервалу між настанням, тоді як інші оцінюють тон, початок і іноді зміщення в тій самій структурі.

Транскрипція на рівні потоку, яку також називають Multi-Pitch Streaming (MPS), націлена на групування оцінених тонів або нот у потоки, де кожен потік зазвичай відповідає одному інструменту або музичному голосу і тісно пов’язаний з розділенням джерела інструменту. На рис. 10 (нижній графік) показано приклад транскрипції на рівні потоку, де потоки тону звуку різних інструментів мають різні кольори. У порівнянні з транскрипцією на рівні ноти, контур висоти кожного потоку набагато довший, ніж одна нота, і містить численні розриви, спричинені тишею, невисотними звуками та різкими змінами частоти. Тому прийомів, які часто використовуються в транскрипції на нотному рівні, як правило, недостатньо, щоб згрупувати висоту звуків у довгий і неперервний контур. Однією важливою ознакою для MPS, яка не досліджується в MPE та відстеженні нот, є тембр: ноти одного потоку (джерела) зазвичай демонструють подібні тембральні характеристики у

порівнянні з нотами в різних потоках. Тому транскрипцію на рівні потоку в літературі також називають відстеженням тембру або інструментом. Наявних робіт на цьому рівні небагато.

З кожним рівнем завдання транскрипції стає складнішим, оскільки потрібно моделювати більше музичних структур і сигналів. Проте всі вихідні дані транскрипції на цих трьох рівнях є параметричними транскрипціями, які є параметричними описами аудіоконтенту. «Фортепіанний рулон» MIDI, показаний на рис. 9(в), є хорошим прикладом такої транскрипції. Це справді абстракція музичного аудіо, однак вона ще не досягла рівня абстракції нотних записів: час все ще вимірюється в одиницях секунд замість ударів; висота звуку вимірюється в номерах MIDI замість написаних імен нот, які сумісні з клавішею; а поняття такт, метр, тональність, гармонія та потік відсутні.

Транскрипція на рівні нотації має на меті транскрибувати музичне аудіо в доступну для читання музичну партитуру, як-от нотний запис, який широко використовується в західній класичній музиці. Транскрипція на цьому рівні вимагає глибшого розуміння музичних структур, включаючи гармонійні, ритмічні та потокові структури. Гармонійні структури, такі як клавіші та акорди, впливають на написання нот кожного тону MIDI; ритмічні структури, такі як удари та такти, допомагають квантувати довжину нот; і потокові структури допомагають призначати ноти. Була деяка робота з оцінки музичних структур за аудіо або MIDI-уявленнями виконання. Наприклад, були запропоновані методи для написання тону, квантування часу та голосового поділу із виконуваних MIDI-файлів. Однак мало роботи було зроблено щодо інтеграції цих структур у повну транскрипцію нотних записів, особливо для поліфонічної музики. Кілька програмних пакетів, включаючи Finale, GarageBand і MuseScore, забезпечують функціональність перетворення MIDI-файлу в музичну нотацію, однак результати часто не задовольняють, і незрозуміло, які музичні структури були оцінені та інтегровані під час процесу транскрипції. Когліаті та ін. запропонував метод перетворення виконання MIDI в музичну нотацію

із систематичним порівнянням виконання транскрипції із вищезгаданим програмним забезпеченням. З точки зору транскрипції аудіонотації, Карвальо та Смарагдіс запропонували доведення концепції з використанням наскрізних нейронних мереж для безпосереднього відбиття музичного аудіо в нотному записі без явного моделювання музичних структур.

3.2 Створення алгоритму АТМ

Оскільки, той факт що поліфонічна музика містить поєднання кількох одночасних джерел (наприклад, інструментів, вокалу) з різною висотою, гучністю та тембром (якістю звуку), причому кожне джерело створює один або кілька музичних голосів, стає дуже складно виконати музичне транскрибування.

Тому було вирішено виокремити з джерела музики тональні та перкусійні компоненти, використовуючи розглянуту вище систему для розділення джерел звуку Spleeter, яка може розділити аудіофайл на 4 окремі файли vocals, drums, bass і other. Оскільки на меті отримати саме тональні та перкусійні (ударні) компоненти, то для подальшого алгоритму нам необхідні тільки vocals та drums файли.

Файл vocals буде містити голос виконавців, що є тональною компонентою. Саме по цій компоненті надалі будуть визначатись ноти. Але для цього все ще необхідно знати, коли нота починається. Тож, щоб полегшити задачу вирішено знаходити темп музики з допомогою файлу drums, що містить перкусійну компоненту.

Файл bass містить баси, що заважають при виконанні задачі транскрибування, тому їх необхідно було вилучити.

Інші інструменти та шуми, що знаходяться у файлі other, також потрібно виокремити.

Отже, для розв'язання подальшої задачі нам достатньо тільки двох файлів: vocals та drums.

3.2.1 Знаходження темпу музичної композиції

Темп – це крок або швидкість музики. Вищий темп означає швидшу пісню, тоді як нижчий темп означає повільнішу пісню.

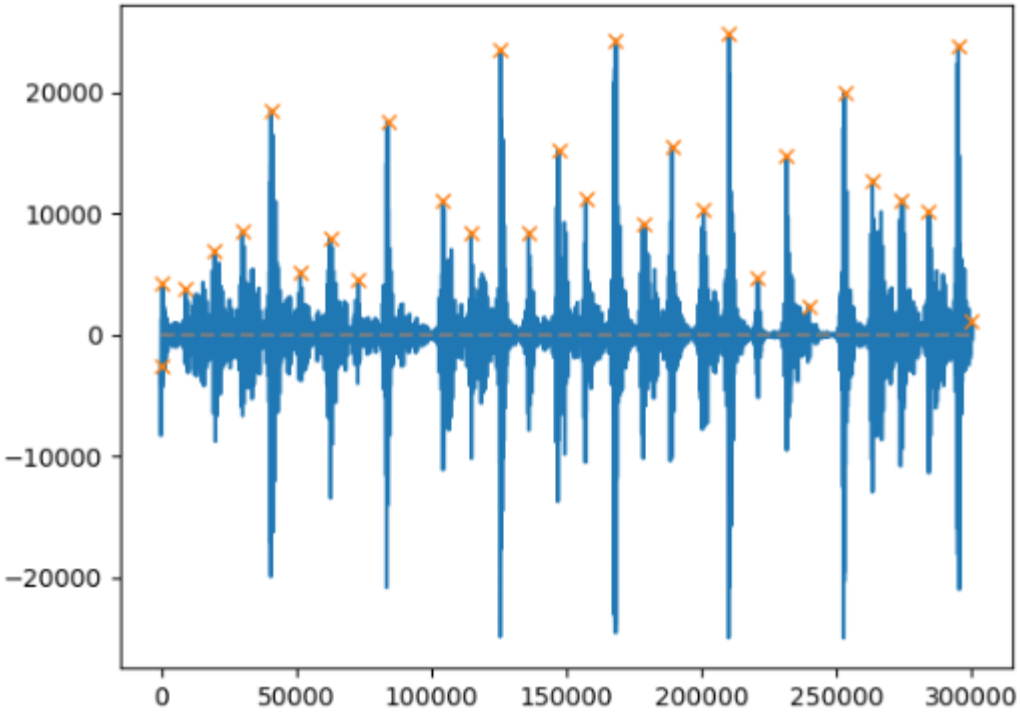


Рисунок 11. Знаходження «піків» музичної композиції

Визначення темпу музичної композиції робиться для того, щоб мати мапу розташування нот. Для знаходження темпу виконується згладжування сигналу, що знаходиться у файлі drums, та пошук точок максимумів сигналу, так званих “піків”.

Згладжування виконується так, щоб відстань між піками була більш менш рівномірною. Але це не завжди може бути так, оскільки у музичній композиції можуть бути частини, де темп прискорений або, навпаки, сповільнений.

Відстані між “піками” і є темпом музичної композиції. На цих часових ділянках виконуватиметься розпізнавання нот.

3.2.2 Розпізнавання нот

На тих часових ділянках, що ми отримали на попередньому кроці, застосовується алгоритм швидкого перетворення Фур’є (FFT).

FFT є важливим методом вимірювання в науці для вимірювання звуку та

акустики. Він перетворює сигнал в окремі спектральні компоненти й тим самим надає частотну інформацію про сигнал. FFT використовуються для аналізу несправностей, контролю якості та моніторингу стану машин або систем.

Строго кажучи, FFT є оптимізованим алгоритмом для реалізації «Дискретного перетворення Фур'є» (DFT). Сигнал відбирається протягом певного проміжку часу і розділяється на його частотні компоненти. Ці компоненти являють собою поодинокі синусоїдальні коливання на різних частотах, кожне зі своєю амплітудою і фазою. Це перетворення показано на рис. 12.

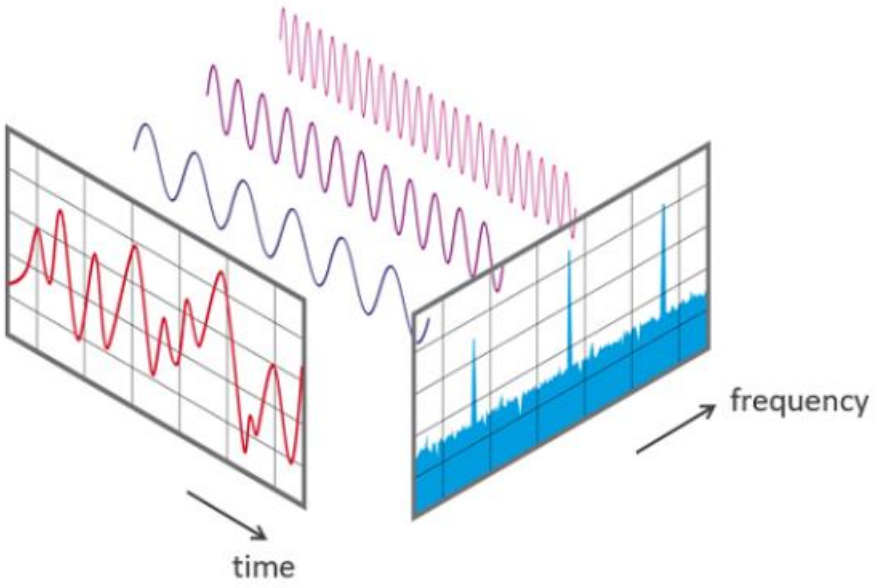


Рисунок 12. Протягом виміряного проміжку часу сигнал містить 3 різні домінантні частоти

З цих компонент обирається домінантна, тобто та що має найвищу частоту. Маючи всі домінуючи частоти музичної композиції можна отримати ноти та їх октави.

ВИСНОВКИ

В роботі було розглянуто проблему автоматичної транскрипції музики з використанням методів машинного навчання для полегшення транскрибування аудіокомпозицій. А саме використання нейронної мережі для розділення джерела звуку й отримання необхідних компонент для виконання задачі транскрибування. Було проаналізовано різні підходи та види мереж, що виконують задачу поділу джерела звуку. В результаті цього аналізу було обрано відповідну систему розділення джерела звуку, показники продуктивності та якості якої є одними з найкращих на даний час.

Завдання полегшити задачу транскрибування музичної композиції й створення відповідного програмного застосунку «Система для автоматичної транскрипції музичної композиції» були виконані. Однак результати все ще не є ідеальними, тому надалі планується зробити певні оптимізації та покращення алгоритму. Оскільки на цей час алгоритм працює тільки з музичними композиціями, в яких присутній вокал, то планується розробити можливість транскрибувати тільки музичний супровід.

Успішна система автоматичної транскрипції музики може забезпечити широкий спектр взаємодій між людьми та музикою, включаючи музичну освіту (наприклад, через системи автоматичного навчання грі на музичних інструментах), створення музики (наприклад, диктування імпровізованих музичних ідей та автоматичний музичний супровід), музичний реліз (наприклад, візуалізація музичного вмісту та інтелектуальне редагування на основі вмісту), пошук музики (наприклад, індексація та рекомендація музики за мелодією, басом, ритмом чи акордом), музикознавство (наприклад, аналіз джазових імпровізацій та іншої музики без нот), а також можливість перевірки музичної композиції на плагіат. Таким чином, АТМ — це спроможна технологія з очевидним потенціалом як для економічного, так і для соціального впливу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Han Li, Kean Chen, Lei Wang, Jianben Liu, Baoquan Wan, Bing Zhou. 2022. Sound Source Separation Mechanisms of Different Deep Networks Explained from the Perspective of Auditory Perception.
2. Stefan Uhlich, Marcello Porcu, Franck Giron, Michael Enenkl, Thomas Kemp, Naoya Takahashi and Yuki Mitsufuji. 2017. Improving music source separation based on deep neural networks through data augmentation and network blending.
3. Andreas Jansson, Eric Humphrey, Nicola Montecchio, Rachel Bittner, Aparna Kumar, Tillman Weyde. 2017. Singing voice separation with deep U-Net Convolutional Networks.
4. Releasing Spleeter: Deezer Research source separation engine. 2019.
5. https://deezer.io/releasing-spleeter-deezer-r-d-source-separation-engine-2b88985e797e
6. Romain Hennequin, Anis Khlif, Felix Voituret, Manuel Moussallam. 2020. Spleeter: a fast and efficient music source separation tool with pre-trained models. Automatic Music Transcription: An Overview.
7. Emmanuel Vincent, Rémi Gribonval, Cédric Févotte. 2010. Performance measurement in blind audio source separation.
8. A. Klapuri and M. Davy, Eds., Signal Processing Methods for Music Transcription. New York: Springer, 2006.
9. E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, “Automatic music transcription: challenges and future directions,” Journal of Intelligent Information Systems, vol. 41, no. 3, pp. 407–434, Dec. 2013.
10. M. Müller, D. P. Ellis, A. Klapuri, and G. Richard, “Signal processing for music analysis,” IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 6, pp. 1088–1110, Oct. 2011.
11. M. Schedl, E. Gómez, and J. Urbano, “Music information retrieval: Recent developments and applications,” Foundations and Trends in Information Retrieval,

vol. 8, pp. 127–261, 2014.

12. N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, “Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription,” in Proc. International Conference on Machine Learning (ICML), 2012.
13. T. Virtanen, M. D. Plumbley, and D. P. W. Ellis, Eds., Computational Analysis of Sound Scenes and Events. Springer, 2018.
14. L. Su and Y.-H. Yang, “Escaping from the abyss of manual annotation: New methodology of building polyphonic datasets for automatic music transcription,” in Proc. International Symposium on Computer Music Multidisciplinary Research (CMMR), 2015, pp. 309–321.
15. Z. Duan, B. Pardo, and C. Zhang, “Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 18, no. 8, pp. 2121–2133, 2010.
16. Z. Duan and D. Temperley, “Note-level music transcription by maximum likelihood sampling.” in ISMIR, 2014, pp. 181–186.
17. Z. Duan, J. Han, and B. Pardo, “Multi-pitch streaming of harmonic sound mixtures,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, no. 1, pp. 138–150, Jan 2014.
18. V. Emiya, R. Badeau, and B. David, “Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 18, no. 6, pp. 1643–1654, 2010.
19. L. Su and Y.-H. Yang, “Combining spectral and temporal representations for multipitch estimation of polyphonic music,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 10, pp. 1600– 1612, Oct 2015.
20. P. H. Peeling, A. T. Cemgil, and S. J. Godsill, “Generative spectrogram factorization models for polyphonic piano transcription,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 18, no. 3, pp. 519–527, March 2010.