Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: Розробка застосунку синтезу вокалу засобами Python

Виконав: студент IV курсу, групи СНс-42
спеціальності 122 Комп’ютерні науки
(шифр і назва спеціальності)

(підпис) Кліщ М.В.
(прізвище та ініціали)

Керівник (підпис) Козбур Г.В.
(прізвище та ініціали)

Нормоконтроль (підпис) Шимчук Г.В.
(прізвище та ініціали)

Завідувач кафедри (підпис) Боднарчук І.О.
(прізвище та ініціали)

Рецензент (підпис) (прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
«___» _червня_ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Кліщ Максим Володимирович
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка застосунку синтезу вокалу засобами Python

Керівник роботи Козбур Галина Володимирівна, канд. техн. наук, доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «*29*» *квітня* 2024 року № *4/7-472*

2. Термін подання студентом завершеної роботи 24 червня 2024р.

3. Вихідні дані до роботи літературні та інтернет джерела щодо синтезу вокалу

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1. Аналіз задачі синтезу вокалу та постановка завдання. 1.1 Предметна область. 1.2 Огляд існуючих застосунків синтезу вокалу. 1.3 Огляд існуючих рішень на основі глибинного навчання. 1.4 Постановка завдання. 1.5 Висновок до першого розділу. 2. Проєктування архітектури моделі та застосунку синтезу вокалу. 2.1 Пайплайн застосунку. 2.2 Архітектура моделі. 2.3 Датасет. 2.4 Попереднє опрацювання датасету. 2.5 Функція втрат. 2.6 Тренувальний процес. 2.7 Постфільтр. 2.8 Проєктування системи класів застосунку. 2.9 Інтерфейс застосунку синтезу вокалу. 2.10 Висновок до другого розділу. 3. Оцінювання якості моделі та тестування застосунку синтезу вокалу. 3.1 Оцінка прогнозованих значень. 3.2 Об'єктивне оцінювання. 3.3 Суб'єктивне оцінювання. 3.4 Тестування функціональності застосунку синтезу вокалу. 3.5 Висновок до третього розділу. 4. Безпека життєдіяльності, основи охорони праці. Висновки. Перелік джерел.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Титульний слайд. 2. Завдання роботи. 3. Мета роботи. 4.Задача синтезу вокалу. 5. Пропонована архітектура моделі. 6. Датасет. 7. Постфільтрація. 8. Діаграма класів застосунку. 9. Інтерфейс застосунку. 10. Функція втрат на валідаційній вибірці. 11. Об'єктивне оцінювання. 12. Суб'єктивне оцінювання. Mean opinion score. 13. Суб'єктивне оцінювання. Preference score. 14. Прогнозовані значення lf0. 15. MGC. 16. BAP. 17. Прогнозоване аудіо. 18. Результати. 19. Завершальний слайд.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В.С., доцент кафедри МТ	12.06.2024	15.06.2024

7. Дата видачі завдання 29 січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	30.01.2024	*Виконано*
2.	Підбір джерел про методи синтезу вокалу	31.01.2024-03.02.2024	*Виконано*
3.	Опрацювання джерел щодо існуючих методів синтезу	04.02.2024-06.02.2024	*Виконано*
	вокалу		
4.	Розробка моделі синтезу вокалу	07.02.2024-08.02.2024	*Виконано*
5.	Розробка застосунку синтезу вокалу	09.02.2024-11.02.2024	*Виконано*
6.	Оформлення розділу «Аналіз задачі синтезу вокалу та	03.06.2024-05.06.2024	*Виконано*
	постановка завдання»		
7.	Оформлення розділу «Проєктування архітектури	06.06.2024-08.06.2024	*Виконано*
	моделі та застосунку синтезу вокалу»		
8.	Оформлення розділу «Оцінювання якості моделі та	09.06.2024-11.06.2024	*Виконано*
	Тестування застосунку синтезу вокалу»		
9.	Виконання завдання до підрозділу «Безпека життєдіяльності»	12.06.2024-13.06.2024	*Виконано*
10.	Виконання завдання до підрозділу «Основи охорони	14.06.2024-15.06.2024	*Виконано*
	праці»		
11.	Оформлення кваліфікаційної роботи	16.06.2024-17.06.2024	*Виконано*
12.	Нормоконтроль	18.06.2024-19.06.2024	*Виконано*
13.	Перевірка на плагіат	20.06.2024	*Виконано*
14.	Попередній захист кваліфікаційної роботи	21.06.2024	*Виконано*
15.	Захист кваліфікаційної роботи	28.06.2024	

Студент _____ Кліщ М.В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Козбур Г.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка застосунку синтезу вокалу засобами Python // Кваліфікаційна робота освітнього рівня «Бакалавр» // Кліщ Максим Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-42 // Тернопіль, 2024 // С. 60, рис. – 22, табл. – 2, слайдів – 19, додат. – 0, бібліогр. – 81.

Ключові слова: синтез вокалу, синтез співочого голосу, нейронні мережі, машинне навчання, глибинне навчання, залишкова мережа, python.

Кваліфікаційна робота присвячена розробці методу синтезу вокалу та розробці на основі нього застосунку.

У першому розділі кваліфікаційної роботи описано існуючі застосунки синтезу вокалу. Розглянуто існуючі методи синтезу вокалу на основі глибинного навчання. Визначено вимоги до застосунку, який розроблено в процесі виконання роботи.

У другому розділі кваліфікаційної роботи запропоновано архітектуру моделі синтезу вокалу. Описано архітектуру застосунку синтезу вокалу. Показано етапи опрацювання датасету. Висвітлено процес тренування моделі.

У третьому розділі кваліфікаційної роботи описано тестування застосунку синтезу вокалу. Оцінено якість отриманої моделі синтезу вокалу. Описано процес суб'єктивного та об'єктивного оцінювань.

У четвертому розділі кваліфікаційної роботи описано фізіогічний та психологічний впливи синтезованого вокалу на життєдіяльність людини. Висвітлено проблеми, які можуть виникати під час роботи зі застосунок. Подано рекомендації щодо безпечної роботи зі застосунком синтезу вокалу.

ANNOTATION

Development of a Vocal Synthesis Application Using Python // Qualification work of the educational level "Bachelor" // Klishch Maksym // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SNs-42 // Ternopil, 2024 // P. 60, fig. – 22, tabl. – 2 , chair. – 19 , annexes. – 0 , references – 81.

Keywords: vocal synthesis, singing voice synthesis, neural networks, machine learning, deep learning, residual network, python.

The qualification work is devoted to the development of a vocal synthesis method and an application based on it.

The first chapter of the qualification work describes existing applications of vocal synthesis. Existing methods of vocal synthesis based on deep learning are considered. The requirements for the application developed in the course of the work are defined.

In the second chapter of the qualification work, the architecture of the vocal synthesis model is proposed. The architecture of the vocal synthesis application is described. The stages of dataset processing are shown. The process of model training is covered.

The third chapter of the qualification work describes the testing of the vocal synthesis application. The quality of the resulting vocal synthesis model is evaluated. The process of subjective and objective evaluation is described.

The fourth chapter of the qualification work describes the physiological and psychological effects of synthesized voice on human. The problems that may arise when working with the application are highlighted. Recommendations for safe work with the vocal synthesis application are given.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

BAP (англ. band aperiodicit) – ступінь неперіодичності в певних частотних смугах сигналу.

CNN (англ. Convolution Neural Network) – згорткова нейронна мережа.

f0 – фундаментальна частота.

GAN (англ. Generative Adversarial Network) – генеретивна змагальна мережа.

$\mathcal{L}$ – функція втрат.

$\mathcal{L}_f$ – функція втрат для lf0.

$\mathcal{L}_{sp}$ – функція втрат для MGC.

$\mathcal{L}_{ap}$ – функція втрат для BAP.

$\mathcal{L}_{vuv}$ – функція втрат для визначення чи озвучено фрагмент.

lf0 – тензор, що значення якого є натуральним логарифмом від відповідних значень f0.

log – натуральний логарифм.

LSTM (англ. Long Short-Term Memory) – довга короткочасна пам’ять.

MDN (англ. Mixture Density Network) – мережа змішаних розподілів.

MGC (англ. mel-generalized cepstrum) – мел-загальні кептральні коефіцієнти.

SVS (англ. Singing Voice Synthesis) – синтез вокалу; синтез співочого голосу.

TTS (англ. Text to Speech) – синтез мовлення.

VUV (англ. voiced/unvoiced) – озвучено/неозвучено.

λ_f – ваговий коефіцієнт функції втрат $\mathcal{L}_f$.

λ_{sp} – ваговий коефіцієнт функції втрат $\mathcal{L}_{sp}$.

λ_{ap} – ваговий коефіцієнт функції втрат $\mathcal{L}_{ap}$.

λ_{vuv} – ваговий коефіцієнт функції втрат $\mathcal{L}_{vuv}$.

дБ – децибел.

дБА – акустичний децибел.

Синтез вокалу – завдання генерування співочого голосу на основі інформації з партитури, що включає інформацію про ноти та текст.

ЗМІСТ

ВСТУП..... 9
РОЗДІЛ 1. АНАЛІЗ ЗАДАЧІ СИНТЕЗУ ВОКАЛУ ТА ПОСТАНОВКА ЗАВДАННЯ..... 10
1.1 Предметна область..... 10
1.2 Огляд існуючих застосунків синтезу вокалу..... 10
1.3 Огляд існуючих рішень на основі глибинного навчання..... 13
1.4 Постановка завдання..... 16
1.5 Висновок до першого розділу..... 17
РОЗДІЛ 2. ПРОЄКТУВАННЯ АРХІТЕКТУРИ МОДЕЛІ ТА ЗАСТОСУНКУ СИНТЕЗУ ВОКАЛУ..... 18
2.1 Пайплайн застосунку..... 18
2.2 Архітектура моделі..... 19
2.3 Датасет..... 21
2.4 Попереднє опрацювання датасету..... 21
2.5 Функція втрат..... 24
2.6 Тренувальний процес..... 25
2.7 Постфільтр..... 26
2.8 Проєктування системи класів застосунку..... 27
2.9 Інтерфейс застосунку синтезу вокалу..... 33
2.10 Висновок до другого розділу..... 33
РОЗДІЛ 3. ОЦІНЮВАННЯ ЯКОСТІ МОДЕЛІ ТА ТЕСТУВАННЯ ЗАСТОСУНКУ СИНТЕЗУ ВОКАЛУ..... 34
3.1 Оцінка прогнозованих значень..... 34
3.2 Об'єктивне оцінювання..... 41
3.3 Суб'єктивне оцінювання..... 42
3.4 Тестування функціональності застосунку синтезу вокалу..... 45
3.5 Висновок до третього розділу..... 46
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ..... 47

4.1 Фізіологічний та психологічний вплив синтезованого вокалу на життєдіяльність людини.....47
4.2 Заходи щодо зниження ризиків для оператора ПК при роботі із застосунком синтезу вокалу.....48
4.3 Висновок до четвертого розділу.....50
ВИСНОВКИ.....52
ПЕРЕЛІК ДЖЕРЕЛ.....53

ВСТУП

Актуальність теми. Синтез вокалу є однією з галузей сучасних технологій обробки звуку, що динамічно розвивається. З розвитком машинного навчання, зокрема методів глибинного навчання, з’являється все більше можливостей для створення високоякісних синтезованих співочих голосів, які можуть застосовуватися в різних сферах.

Цифрова трансформація означає інтеграцію цифрових технологій у всі аспекти бізнесу, що призводить до фундаментальних змін у способі функціонування організацій і надання цінності клієнтам [1-3]. Синтезований вокал трансформує музичну індустрію. Композитори можуть попередньо прослухати свій твір без залучення живих виконавців. Музика може бути створена швидше і з меншими витратами, оскільки немає потреби наймати вокалістів.

Також можуть з'являтися нові жанри та стилі музики, що використовують синтезовані голоси.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є розробка застосунку синтезу вокалу. Для досягнення поставленої мети потрібно виконати наступні завдання:

– Оглянути існуючі методи та застосунки синтезу вокалу.
– Описати вимоги до застосунку синтезу вокалу.
– Запропонувати архітектуру застосунку синтезу вокалу.
– Обґрунтувати вибір методу синтезу вокалу та реалізувати його.
– Розробити застосунок з використанням вибраного методу.
– Оцінити якість реалізованих методу та застосунку синтезу вокалу.

Практичне значення одержаних результатів.

У ході виконання роботи було розроблено застосунок синтезу вокалу, який є агностичним до архітектури моделі глибинного навчання синтезу вокалу, мови вхідного тексту і мови виконання співу.

РОЗДІЛ 1. АНАЛІЗ ЗАДАЧІ СИНТЕЗУ ВОКАЛУ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Предметна область

Синтез вокалу або синтез співочого голосу (SVS) – завдання генерування співочого голосу на основі інформації з партитури, що включає інформацію про ноти та текст [4].

Синтез вокалу тісно пов'язаний із задачею синтезу мовлення (TTS). Багато архітектур для вирішення задачі SVS базується на тих, що використовуються для вирішення задачі TTS (наприклад, XiaoiceSing [4] на основі FastSpeech [5] або Bytesing [6] – Tacotron [7]).

Синтез вокалу відрізняється від синтезу мовлення тим, що не потребує моделювання наголошених голосних. Наголошеність складу зазвичай враховується при написанні мелодії [8].

Важливою для синтезу вокалу є фундаментальна частота (f0). Вона має більший діапазон значень та є більш чутливою до змін, порівнюючи із задачею TTS [4].

1.2 Огляд існуючих застосунків синтезу вокалу

Ряд комерційних застосунків пропонує вирішення задачі синтезу вокалу. Найбільш популярним серед них є Vocaloid [9]. Vocaloid для синтезу вокалу спирається на бібліотеку фонетичних звуків, записаних від професійних співаків. Користувачі можуть керувати висотою і тривалістю кожної фонеми, щоб підібрати потрібну мелодію. Також програма дозволяє додавати вібрато, динаміку, артикуляцію та інші параметри для підвищення реалістичності або для досягнення інших цілей.

Особливості функціонування Vocaloid в контексті конкатенативного синтезу вокалу описано у роботі [10].

UTAU [11], безкоштовне програмне забезпечення для синтезу вокалу, розроблене Ameya/Ayame. OpenUTAU [12] – застосунок з відкритим вихідним кодом, спрямований на покращення та розширення можливостей UTAU. Обидва застосунки є альтернативами Vocaloid. Ці програми дозволяють користувачам створювати синтезований вокал, вводячи тексти і мелодії, подібно до Vocaloid, але з виразними відмінностями в технологіях, спільноті та доступності.

У UTAU/OpenUTAU користувачі можуть створювати власні банки голосів, записуючи і налаштовуючи фонетичні зразки, що дозволяє працювати з широким спектром вокальних стилів і мов.

На рисунку 1.1 подано інтерфейс застосунку OpenUTAU.

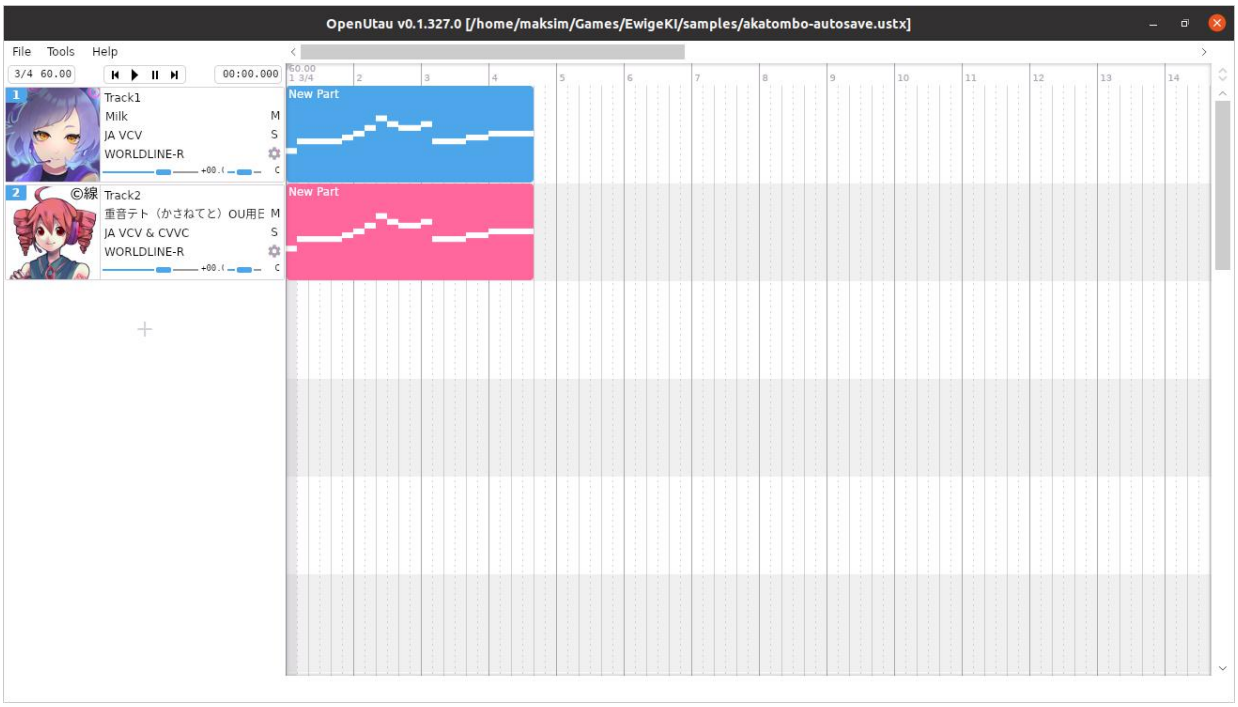

Рисунок 1.1 – Інтерфейс застосунку OpenUTAU

Synthesizer V (або SynthV) [13], представлений компанією Dreamtonics, представляє нове покоління програмного забезпечення для синтезу вокалу, яке робить акцент на високоякісному вокальному синтезі та зручному інтерфейсі.

SynthV швидко завоював визнання завдяки реалістичним результатам, які генеруються, і розширеним можливостям, позиціонуючи себе як сильного конкурента таким відомим програмам, як Vocaloid, UTAU і OpenUTAU.

Як зазначено на сайті Dreamtonics SynthV використовує гібрид методів штучних нейронних мереж та конкатенативного синтезу [13].

NEUTRINO [14] використовує передові архітектури нейронних мереж(наприклад, трансформетри та дифузійні моделі), щоб відтворювати нюанси людського вокалу. Застосунок є вільним для використання та забезпечує високоякісний результат, який можна порівняти з комерційними аналогами. NEUTRINO доступний для Windows, Linux, MacOS, також доступна версія, що підписана як “Online”. Остання дозволяє генерувати результат за допомогою консолі, що подано на рисунку 1.2.

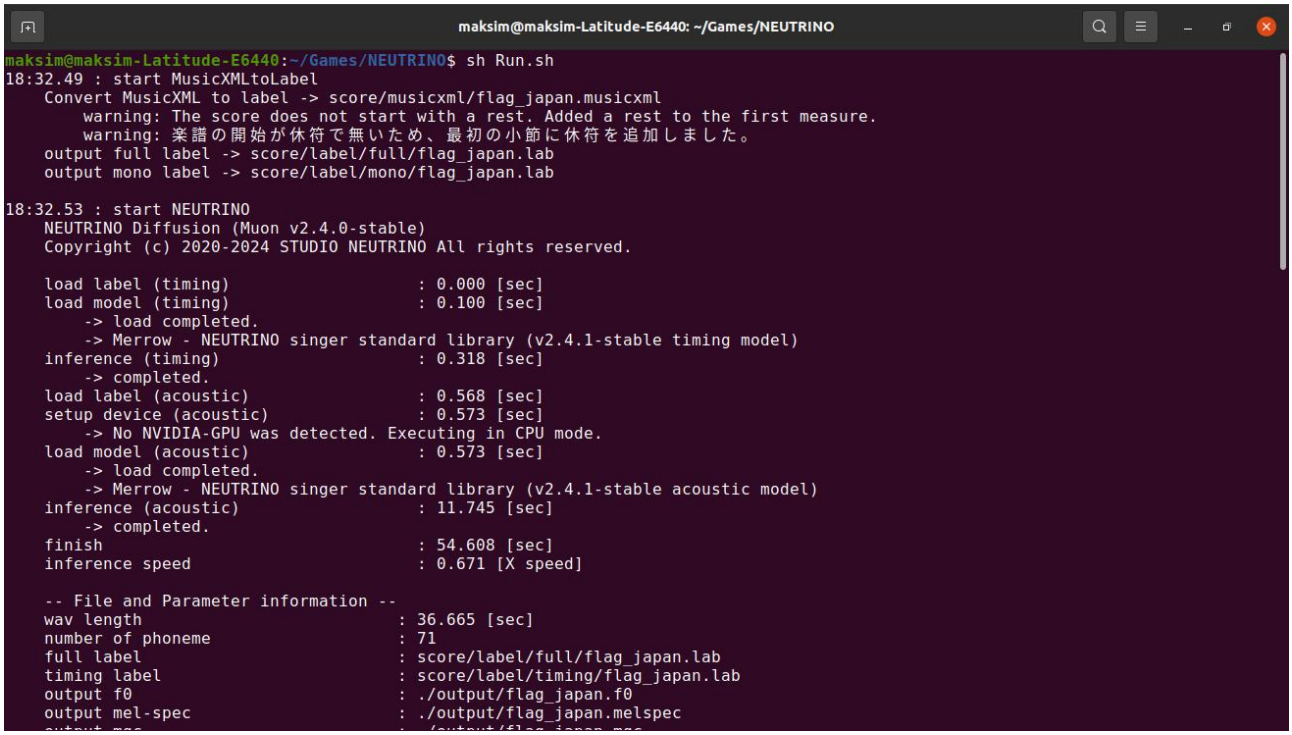

Рисунок 1.2 – Консольний інтерфейс застосунку NEUTRINO

Проєкт Synsinger [15] цікавий тим, що пропонує синтез вокалу на основі формантного синтезу.

Окрім зазначених вище застосунків існує ще ряд аналогів, проте вони представляють менший інтерес, оскільки відтворюють функціонал згаданих раніше.

1.3 Огляд існуючих рішень на основі глибинного навчання

У роботі [16] запропоновано підходити на основі архітектури генеративної змагальної мережі (GAN) [17]. Використання GAN забезпечує високу якість синтезованого звуку, зменшуючи артефакти та спотворення, які часто виникають при традиційних методах синтезу.

У роботі [18] адаптується дифузійна модель [19] для синтезу співу.

У роботі [4] адаптується модель FastSpeech [5] для задачі SVS. Особливістю є використанням блоків Feed-Forward Transformer, які базуються на механізмі самоуваги [20]. Для вирішення проблеми невідповідності кількості фонем та довжини спектограми FastSpeech використовує Length Regulator, який визначає тривалість фонеми. XiaoiceSing, окрім фонем, використовує як вхідні дані довжину та висоту ноти та дозволяє моделювати mel-generalized cepstrum (MGC), band aperiodicit (BAP), voiced/unvoiced (V/UV) та log f0 (lf0). Запроповано моделювання lf0 за допомогою залишкового з'єднання, тобто модель передбачає різницю між цільовим значенням та висотою вхідної ноти. Було використано WORLD [21] (D4C edition [22]) як вокодер.

HiFiSinger [23] розвиває ідею XiaoiceSing, згідно з якою замінюють WORLD на Parallel WaveGAN [24]. Також запропоновано використання multi-length GAN (ML-GAN) для моделювання аудіо з великою довжиною та sub-frequency GAN (SF-GAN) для моделювання аудіо з високою частотою дискретизації.

У роботі [25], що є продовженням роботи [4], було запропоновано замінити блоки з Feed-Forward Transformer на ConvFFT, які включають залишкові згорткові блоки до попередніх.

Основною перевагою XiaoiceSing2 та HiFiSinger є висока якість звуку. Проте, недолік цих моделей полягає у кількості ресурсів необхідних для тренування – XiaoiceSing2 було натреновано з використання чотирьох відеокарт NVIDIA V100, тривалість тренування 24 години та було використано датасет, що включає 5 годин аудіозапису, а HiFiSinger використовує датасет тривалістю 11 годин.

У роботі [26] представлено модель на основі Long Short-Term Memory (LSTM) [27]. Було продемонстровано, що рекурсивні архітектури нейронних мереж потребують меншої кількості параметрів та мають кращу якість прогнозування, порівнюючи з глибинною нейронною мереж.

У роботі [28] описано процес збору даних з мережі Інтернет для синтезу вокалу та їх обробки. Процес обробки включає відокремлення вокалу від акомпанементу. Автори пропонують використання механізму спрямованої уваги [29] для визначення фонем і їх тривалості, і описують яким чином з матриці уваги отримати тривалість фонеми. Модель DeepSinger, яка описується у роботі, базується на Fast-Speech. Відмінність цієї моделі від XiaoiceSing полягає у тому, що остання виконує об'єднання вхідних параметрів(фонеми, висота ноти та інші) до передання їх у енкодер, а DeepSinger – спочатку пропускає ознаки через відповідні енкодери, після чого об'єднює. Їх датасет складався з 12 годин для англійської мов, 19 годин для кантонської, 63 годин.

DeepSinger та XiaoiceSing потребують великої кількості даних для тренування, оскільки базуються на архітектурі трансформер.

Розглянемо моделі SinSy.

У роботі [30] пропонується модель на основі прихованої марковської моделі. Описано вхідні параметри моделі, проте такі як кількість складів у фразі, кількість фонем у складі, як зазначається у [28], є неактуальними з використанням нейронних мереж. Перевагою такої моделі є невеликий розмір та невелика кількість обчислювальних ресурсів, які необхідні для її використання. Претренована модель може легко вивчати інші голоси.

У роботі [31] запропоновано використання нейронних мереж для синтезу вокалу. Запропонована модель визначає різницю між вхідним значенням lf0, яке отримано з висот нот, та бажаним значенням, яке представлене у тренувальній вибірці. Також пропонується інтерполювати lf0 для безголосових участків та обґрунтовується перевага цього рішення.

У подальшому SinSy розвивається у [32]. Автори продемонстрували перевагу використання Mixture Density Network (MDN) [33] для моделювання тривалості фонем та зміщення часу початку та завершення ноти. Було використано моделювання вібрато.

У роботі [34] описується загальний пайплайн синтезу вокалу, який складається з наступних етапів:

1. Отримання необхідних ознак з вхідних даних.
2. Визначення зміщення часу початку та завершення ноти.
3. Визначення тривалості фонем у межах ноти.
4. Моделювання акустичних ознак.
5. Генерація аудіо.

Для другого та третього етапів використовуються раніше згадані нейронні мережі з використання MDN. Для визначення часу початку та завершення фонем, використовується Hidden Semi-Markov Model [35]. Аналогічно як і XiaoiceSing модель Sinsy використовує залишкове з'єднання для моделювання lf0. Також запропоновано моделювання вібрато як різницю між f0 та її згладженим з використанням медіаного фільтра варіантом. Запропоновано автоматичну корекцію f0 на основі:

1. Апріорного розподілу висот.
2. Висоти псевдоноти.

У роботі [36] запропоновано архітектуру моделі на основі згорткових шарів.

У роботах [6] та [37] пропонуються моделі на основі Tacotron2 [38]. Модель Tacotron2 базується на архітектурі encoder-decoder з механізмом уваги [39]. Для задачі SVS ця архітектура дозволяє менш точно визначати

зміщення нот та тривалість фонем, оскільки очікується, що це нівілюється механізмом уваги.

У роботі [40] запропоновано модель VISinger, яка є адаптацією моделі VITS [41] для задачі SVS. У роботі [42] пропонується покращена версія моделі VISinger2. Для генерації з латентного простору періодичного та аперіодичного сигналів, які передаються до HiFiGAN, пропонується використання DDSP [43].

У роботі NNSVS [44] пропонується набір інструментів для тренування моделей для задачі SVS. За мету автори ставили модульність архітектури застосунку, можливість його розширення, незалежність від мови, для якої треба генерувати вокал. NNSVS є опенсорсним проєктом, який доступний на Github. До NNSVS включено набір рецептів для тренування різних моделей на різних датасетах.

Іншим набором інструментів для роботи з моделями для задачі SVS є ESPnet [45]. На відміну від NNSVS, ESPnet дозволяє включати багато мовців та мов до однієї моделі через speaker embedding та language embedding відповідно.

Можемо виокремити наступні методи синтезу:

– конкатенативний синтез [9, 11, 12];
– формантний синтез [18];
– статистичний параметричний синтез [30, 46, 47];
– синтез засобами глибинного навчання [4, 7, 16, 18].

На сьогоднішній день найбільш перпективними виглядають методи глибинного навчання, зокрема архітектури seq2seq, на основі трансформерів та дифузійних моделей.

1.4 Постановка завдання

Основне завдання даної кваліфікаційної роботи – це розробка застосунку синтезу вокалу, як зазначено в темі. Опишемо основні критерії до нього. Ряд критеріїв представлено у NNSVS [44].

Агностицизм до мови. Архітектура застосунку повинна мінімальної залежати від особливостей конкретних мов.

Агностицизм до моделей синтезу вокалу. Застосунок повинен мінімально залежати від архітектури певної моделі.

Модульність та розширюваність. Для досягнення попередніх двох критеріїв розроблений застосунок повинен бути поділеним на модулі, які будуть описані в другому розділі.

На відмінну від NNSVS, нашою задачею є не розробка набору інструментів для створення моделей синтезу вокалу, а створення застосунку, який використовує моделі глибинного навчання для трансформації нотного запису вокальної партії у спів.

Тренування моделі глибинного навчання з урахування обмежених ресурсів. Під ресурсами розуміємо датасет та обчислювальні потужності, зокрема GPU. З урахуванням цього, ставимо перед собою задачу підібрати архітектуру моделі, що буде компромісною між якістю синтезованого вокалу та часом її тренування.

1.5 Висновок до першого розділу

У першому розділі було розглянуто існуючі застосунки синтезу вокалу, зокрема Vocaloid, UTAU, OpenUTAU, SynthV та NEUTRINO. Описано предметну область синтезу вокалу.

Було поставлено вимоги до застосунку.

У розділі було оглянуто різноманітні архітектури моделей для синтезу вокалу, такі як SinSy, VISinger2, VISinger, DeepSinger, XiaoiceSing, XiaoiceSing2, DiffSinger, SingGAN та інші.

РОЗДІЛ 2. ПРОЄКТУВАННЯ АРХІТЕКТУРИ МОДЕЛІ ТА ЗАСТОСУНКУ СИНТЕЗУ ВОКАЛУ

2.1 Пайплайн застосунку

При проєктуванні застосунку візьмемо за основу архітектуру представлену в SinSy [34] та NNSVS [44]. Пропонований застосунок аналогічно включатиме запропоновані у них етапи пайплайну.

Детальніше розглянемо кожен з етапів.

Перший етап – завантаження вхідних даних. Результатом першого етапу є висоти ноти, час їх початку та завершення, і фонеми, які прив'язані до нот.

Фонеми з вхідного тексту отримуємо за допомогою фонемінізатора, який буде описано нижче. Також отримана послідовність фонем може бути розбитою на склади, якщо це не було визначено партитурою.

Висоти, час початку та завершення нот визначається з партитури. Значення висот нот отримується у форматі midi, який буде конвертовано у герци.

На другому етапі відбувається модифікація значень часу початку та завершення нот, які були отримані на попередньому етапі. Як зазначається у [34] це дозволяє отримати більш реалістичний результат.

На третьому етапі визначається тривалість фонем в межах певної ноти (або послідовності нот у випадку ліги).

У Sinsy [34] пропонується використання MDN для прогнозування зміщень часу початку і завершення ноти та тривалості фонем. OpenUtau [12] реалізуються це на основі правил.

Другий та третій етапи у пропонованому застосунку є опційним, оскільки потенційно може бути включеним у модель синтезу вокалу, наприклад, як це зроблено у XiaoiceSing.

Четвертний етап включає не лише безпосереднє моделювання акустичних ознак, а й підготовку вхідних даних та добування додаткових вхідних ознак необхідних для певної моделі.

Генерування аудіо з акустичних ознак відбувається за допомогою вокодера. Для пропонованого застосунку синтезу вокалу використаємо WORLD [21].

2.2 Архітектура моделі

Враховуючи обмеженість обчислювальних ресурсів та часу необхідно для тренування моделі синтезу вокалу, оберемо архітектуру, яка дозволить отримати задовільну якість звуку та непотребувала великої кількості часу та обчислювальних ресурсів для її тренування. Будемо уникати архітектур, які застосовують механізм уваги, оскільки вони часто потребують великих за кількістю розміром датасетів. Таким чином під вимоги не попадають трансформери та дифузійні моделі. Замість них реалізуємо архітектуру на основі класичних видів нейронних мереж.

Пропонована архітектура моделі базується на архітектурі запропонованій у роботі [34] з наступними модифікаціями:

– Блок зі згортковими шарами (CNN) було замінено на залишкові блоки [48]. Очікується, що їх використання дозволить більш ефективно передавати градієнти під час тренування моделі, що сприяє стабільності та прискорить збіжність. У третьому розділі розглянемо вплив цього рішення на якість прогнозування результату моделлю.

– Не проводиться моделювання вібрато.

– Добавлено декілька Feedforward Neural Network (FNN) шарів після LSTM. Очікується, що це дозволить покращити узагальнення моделі та підвищити її здатність до навчання.

Архітектуру моделі подано на рисунку 2.1.

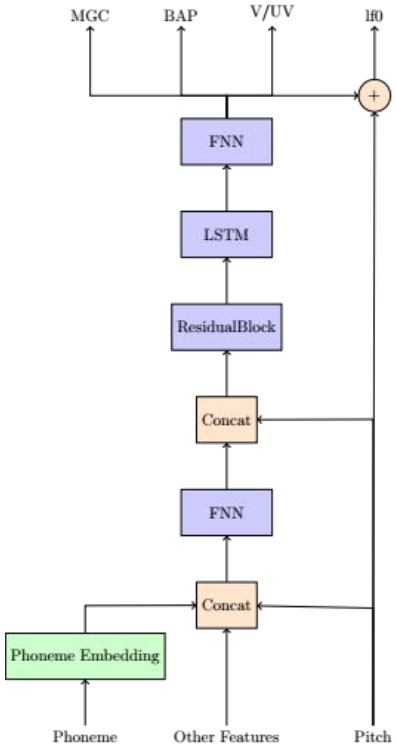

Рисунок 2.1 – Архітектура моделі синтезу вокалу

Опишемо архітектуру Residual1DBlock.

Основна частина складається з одновимірних згорткового шару (Conv1D) з ядром згортки розміром 3 та відступом (padding) – 1 для збереження розмірності вхідного тензору. Для результату застосовується Batch normalization [49] та функцію активації Leaky ReLU (з коефіцієнтом нахилу 0,2). Далі результат передається до одномірного згорткового шару з ядром згортки розміру 1.

Shortcut connection складається з одновимірного згорткового шару з ядром згортки розміром 1.

Використання одновимірного згорткового шару дозволяє послідовні дані.

На рисунку 2.2 подано запропонований Residual1DBlock.

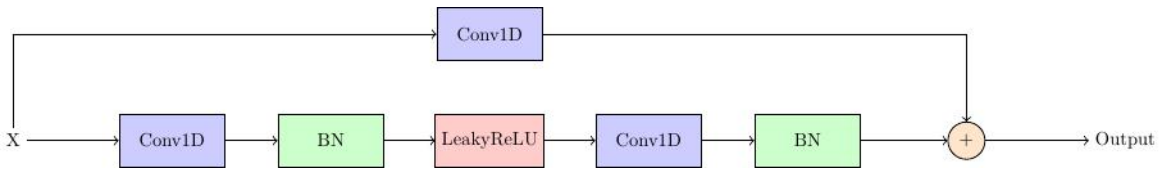

Рисунок 2.2 – Діаграма Residual1DBlock

Цей Residual1DBlock використовується замість згорткових шарів оригінальної моделі.

2.3 Датасет

Було використано датасети Tohoku Kiritan singing database [50] та Children’s Songs Dataset [51].

Tohoku Kiritan singing database включає 50 пісень, загальна тривалість яких становить 3 години 31 хвилину (близько 1 години аудіо з текстом), з темпом від 80 до 190 bpm. Ноти розподілені від соль малої октави до ре другої октави.

Датасет Children’s Songs Dataset містить по 50 пісень та по 50 їх варіантів зі зміненою висотою нот для англійської та корейської мов. Ноти розподілені від фа малої октави до фа другої октави.

Датасет Children’s Songs Dataset містить інформацію про час початку та завершення складів, але не окремо взятої фонеми. Для вирішення цієї проблеми було використано Montreal Forced Aligner [52].

Після опрацювання датасету було отримано 586 семплів для японської мови та 1182 семпли для англійської мови.

2.4 Попереднє опрацювання датасету

Етап 1. Завантаження вхідних даних.

На цьому етапі завантажується інформація про ноти з файлів формату midi та інформація про текст з файлів форматів lab (у випадку Tohoku Kiritan singing database) і csv (у випадку Children’s Songs Dataset).

Етап 2. Уніфікація вхідних даних.

На цьому етапі відбувається приведення датасету до єдиного формату, приводяться одиниці часу до секунд. Для датасету Children’s Songs Dataset

відбувається розділення складів на окремі фонеми та визначення тривалості їх звучання.

Етап 3. Токенізація та фреймізація.

Під токенами вважатемо множину, що складається з символів, які бієктивно представляють фонеми та інших допоміжних символи (наприклад, символи, що позначають дихання або паузу). Кожному токену присвоюється унікальний числовий ідентифікатор.

Знаючи інформацію про те, коли починається та закінчується виконуватися токен, визначаємо фрейми, що відповідатимуть токену за наступними формулами:

$$fr_b = \left\lfloor \frac{t_b}{fr_p} \right\rfloor \tag{2.1}$$

$$fr_e = \left\lceil \frac{t_e}{fr_p} \right\rceil \tag{2.2}$$

де fr_b – номер початкового фрейму, fr_e – номер кінцевого фрейму, t_b – час початку фонеми, t_e – час завершення фонеми, fr_p – тривалість фрейму в секундах.

Етап 4. Опрацювання midi.

Дані про ноти представлені у midi-форматі, для їх завантаження використовується бібліотека pretty-midi [53]. Отримується значення висот нот у герци. У результаті отримуємо вхідну послідовність f0.

Етап 5. Завантаження та уніфікація цільових даних.

На цьому етапі завантажуються цільові дані для моделі. Двохканальний сигнал приводиться до одноканального. При необхідності виконується ресамплінг засобами soxr. Нормалізується гучність до -26 дБ LUFS, за допомогою pyloudnorm [54].

Етап 6. Виділення f0-контуру, гармонійної спектральної оболонка та аперіодичності з цільового аудіофайлу. Це відбується засобами pyworld [55], який є обгорткою WORLD для python.

Етап 7. Зниження розмірності спектральної оболонки та аперіодичності.

Для цього використовує функції spectral_encode та aperidiocity_encode [56].

У результаті отримуємо тензор розмірності (B, T, 60) для MGC та (B, T, $\lceil \frac{min(15000,\ fs-3000)}{3000} \rceil$) для BAP, де B – розмір батчу, T – кількість фреймів, fs – частота дискретизації.

Цей етап дозволяє зменшити кількість параметрів, які необхідно генерувати моделі. Внаслідок цього зменшується час тренування та розмір моделі.

Етап 8. Нормалізація гармонійної спектральної оболонка та аперіодичності.

Для нормалізації MGC та BAP використовується z-score стандартизація для кожної траєкторії.

Етап 9. Інтерполяція f0-контурів.

Виконується лінійна інтерполяція f0-контурів на неозвучених фреймах, аналогічно як у [32].

Етап 10. Поділ семплу на франгменти.

Кожен семпл ділить на менші франгменти з використання наступних параметрів:

1. Максимальна довжина фрагменту. Обмеження довжини фрагменту дозволить уникнити проблеми з нестачею оперативної пам'яті та відеопам'яті. Було встановлено 30 секунд.

2. Тривалість паузи між озвученими областями, при перевищенні якої відбувається поділ областей на фрагменти. Було встановлено 5 секунд.

3. Мінімальна довжина фрагменту. Було встановлено 5 секунд.

Вхідні дані не включають динамічні ознаки, які були запроповані у [27, 31]. Як зазначають автори роботи [44], вони вважають їх не достатньо корисним. У ході тренування моделі у цьому також переконалися, тому для зменшення кількості обчислень та розміру вхідного тензору моделі, було вирішено відмовитися від динамічних ознак.

2.5 Функція втрат

Функцію втрат визначимо як зважену суму функцій втрат окремих компонентів вихідних даних:

$$\mathcal{L} = \lambda_f \mathcal{L}_f + \lambda_{sp} \mathcal{L}_{sp} + \lambda_{ap} \mathcal{L}_{ap} + \lambda_{vuv} \mathcal{L}_{vuv} \tag{2.3}$$

де $\mathcal{L}$ – функція втрат, $\mathcal{L}_f$ – функція втрат для lf0, $\mathcal{L}_{sp}$ – функція втрат для MGC, $\mathcal{L}_{ap}$ – функція втрат для BAP, $\mathcal{L}_{vuv}$ – функція втрат для визначення чи озвучено фрагмент, λ_f, λ_{sp}, λ_{ap}, λ_{vuv} – вагові коефіцієнти функцій втрат відповідних компонентів.

Значення для λ_f, λ_{sp}, λ_{ap}, λ_{vuv} встановлено $\frac{1}{65}$, $\frac{60}{65}$, $\frac{3}{65}$, $\frac{1}{65}$ відповідно.

$\mathcal{L}_f$ визначено як root mean squared error (RMSE):

$$\mathcal{L}_f = \sqrt{\frac{1}{T}\sum_{i=1}^{T} (\log\hat{f}_0(i) - \log f_0(i))^2} \tag{2.4}$$

де $\hat{f}_0(i)$ – цільове і-те значення f0, $f_0(i)$ – прогнозоване і-те значення f0, T – кількість фреймів.

$\mathcal{L}_{sp}$ та $\mathcal{L}_{ap}$ визначено як mean squared error (MSE).

Обчислимо $\mathcal{L}_{sp}$ за формулою:

$$\mathcal{L}_{sp} = \sum_{i=0}^{N_s-1} \frac{1}{T} \sum_{i=1}^{T} (\hat{s}_{it} - s_{it})^2 \tag{2.5}$$

де $\hat{s}_{it}$ – цільове t-те значення і-ої траєкторії MGC, s_{it} – прогнозоване t-те значення і-ої траєкторії MGC, N_s – кількість траєкторій MGC.

Аналогічну формулу використаємо для обчислення $\mathcal{L}_{ap}$:

$$\mathcal{L}_{ap} = \sum_{i=0}^{N_a-1} \frac{1}{T} \sum_{i=1}^{T} (\hat{a}_{it} - a_{it})^2 \tag{2.6}$$

де $\hat{a}_{it}$ – цільове t-те значення і-ої траєкторії BAP, a_{it} – прогнозоване t-те значення і-ої траєкторії BAP, N_a – кількість траєкторій BAP.

$\mathcal{L}_{vuv}$ – це бінарна перехресна ентропія VUV, що обчислюється за формулою:

$$\mathcal{L}_{vuv} = -\frac{1}{T} \sum_{i=1}^{T} \hat{v}_i \log v_i + (1 - \hat{v}_i) \log(1 - v_i) \tag{2.7}$$

де $\hat{v}_i$ – цільове і-те значення VUV, v_i – прогнозована ймовірність і-того значення VUV бути озвученим.

2.6 Тренувальний процес

Для тренування моделей було використано сервіс Colab [57].

Для тренування моделі було використано оптимізатор AdamW [58] з параметрами β_1=0.9, β_2=0.999 та learning rate 10^{-4}.

Було виконання тренування для англійської та японської мов.

Датасет було розділено на тренувальну, валідаційну та тестувальну вибірки у співвідношенні 73,6%, 18,4% та 8% відповідно.

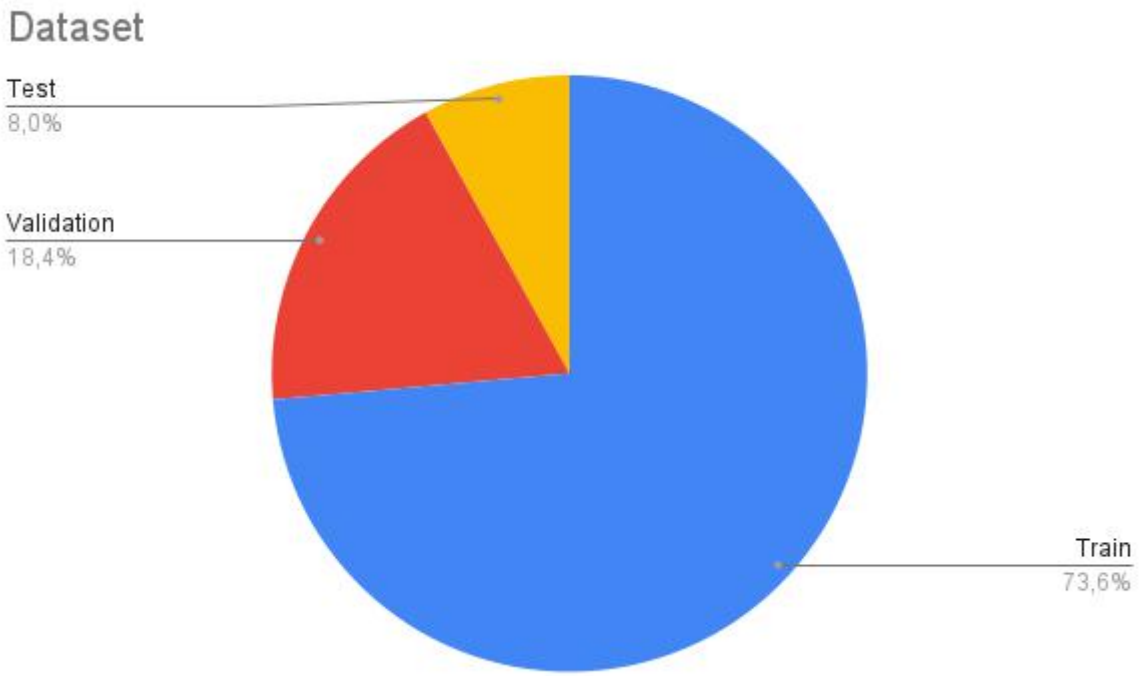

Рисунок 2.3 – Співвідношення між тренувальною, валідаційною та тестувальною вибірками

Розклад змін learning rate було налаштовано таким чином, щоб кожні 20 епох learning rate зменшується у два рази.

2.7 Постфільтр

У роботі [59] запропоновано та обґрунтовано наступну формулу для покращення синтезованих голосів:

$$s'_{ij} = \frac{\sigma_i^{gv}}{\sigma_i}(s_{ij} - \mu_i) + \mu_i \tag{2.8}$$

де s'_{ij} – j-те значення i-ої траєкторії MGC після постфільтрації, s_{ij} – j-те значення i-ої траєкторії MGC до постфільтрації, μ_i – середнє значення i-ої траєкторії MGC, $(\sigma_i)^2$ – дисперсія i-ої траєкторії, $(\sigma_i^{gv})^2$ – глобальна дисперсія i-ої траєкторії вивчена з тренувальних даних.

Вказаний метод постфільтрації є не складним в обчисленнях та не потребує попереднього тренування, достатньо обчислити параметру σ_i^{gv} з датасету.

Цей метод також використовується у [44].

2.8 Проєктування системи класів застосунку

Можемо виділити декілька модулів у цьому застосунку синтезу вокалу:

1. Предметна область. Тут знаходяться класи Note, Phoneme, Part. Їх завдання описати сутності, з якими застосунок працюватиме.

2. Алгоритми машинного та глибинного навчання. Цей модуль включає класи, які описують моделі машинного та глибинного навчання.

3. Взаємодія з вводом та виводом. Модуль включає класи для завантаження вхідних даних з файлової системи та вивід їх користувачеві або збереження у файлову систему.

4. Реалізація функціональної логіки системи. Цей модуль об'єднює інші модулі та описує процес генерування від отримання вхідних даних до виводу їх користувачеві.

5. Фонемінізатори. Модуль, що включає класи та методи, які необхідні для фонемінізації вхідного тексту, тобто для отримання фонем.

Опишемо з точки зору об'єктно-орієнтованого програмування сутності предметної області, які будуть використовуватися. Такими сутностями є фонема, нота, пісня. Вони не реалізують безпосередню логіку застосунку, але дозволять описати терміни, якими необхідно буде оперувати.

Клас Note включає у себе такі параметри як час початку (start_time) її звучання, час завершення (end_time), висота ноти (pitch) та склад (lyric), який озвучується у момент виконання ноти.

Фонема також характеризується часом початку (start_time) та завершення (end_time) її звучання. Окрім цього фонема визначається символом, що дозволить її відокремити від інших. Фонему опишемо класом Phoneme.

Вважаємо, що фонема прив'язана до однієї ноти. Винятком є ліга, коли на один склад припадає декілька нот. У цьому випадку фонеми прив'яжемо лише до першої ноти ліги, а інші позначимо параметром is_slurred=True.

Партія – послідовність нот, яку потрібно виконати. Для її реалізації опишемо клас Part, який містить атрибути notes та phonemes, які відповідають за відображення партії на рівні нот та фонем відповідно.

Таким чином, ці класи дозволять зручно працювати з основними сутностями предметної області, необхідними для подальшої реалізації логіки застосунку. На рисунку 2.4 подано описані вище класи Note, Phoneme, Part.

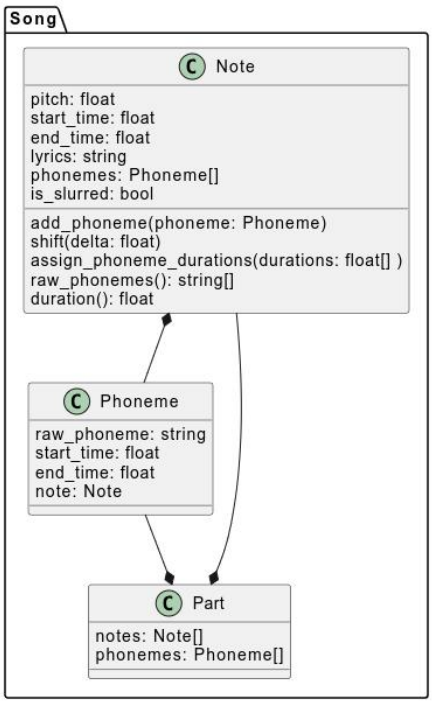

Рисунок 2.4 – Діаграма класів предметної області

Опишемо класи, що відподають за моделі машинного навчання. Ці класи не будуть безпосередньо взаємодіяти з вище описаними.

Для глибинного навчання застосуємо фреймворки Pytorch [60] та pytorch-lightning [61].

Клас TimeLagModel містить метод get_time_lag, який приймає послідовність нот та повертає послідовність, яка вказує на скільки часових одиниць потрібно змістити ноти.

Клас MDNTimeLagModel успадковується від TimeLagModel та torch.nn.Module. Він необхідний для визначення часового зміщення ноти на основі моделі MDN.

Клас DurationModel містить метод get_phoneme_duration, який визначає тривалість кожної фонеми у межах заданого проміжку часу. Цей проміжок часу не залежить від тривалості конкретної ноти, що робить цей метод більш гнучним у застосувані.

Клас MDNDurationModel визначає тривалість фонем на основі MDN. Успадковується від DurationModel та torch.nn.Module.

Клас ResidualBlock1D інкапсулює структуру та функціональність одного залишкового блоку, пристосованого для обробки одновимірних даних у нейронних мережах. Атрибут block визначає послідовність шарів нейронної мережі, які складають основну частину залишкового блоку. Атрибут shortcut використовується для скороченого з'єднання в залишковому блоці.

Клас ResLSTMFNN описує запропоновану нижче модель. Атрибутами цього класу є шари нейронної мережі(ph_embedding, fnn_in, residual, lstm, fc_f0, fc_sp, fc_ap, fc_vuv) та мінімальне, максимальне, середнє значення lf0 та середньоквадратичне відхилення lf0. ResLSTMFNN успадковується від torch.nn.Module.

Клас GlobalVariancePostfilter реалізує постфільтрацію за формулою (2.8). Містить атрибут scale_gv, який зберігає значення σ_i^{gv} . Клас є нащадком torch.nn.Module.

Для забезпечення незалежності від реалізації моделі введемо класи SVSModel та Postfilter, які узагальнюють класи моделей синтезу вокалу та постфільтрів відповідно.

На рисунку 2.5 подано діаграму класів даного модуля.

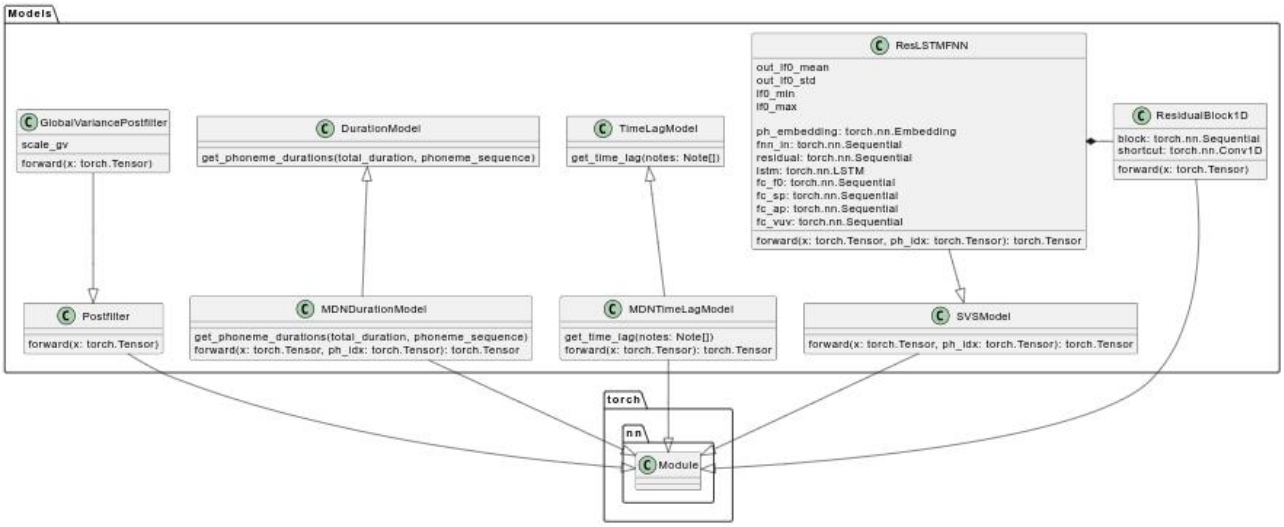

Рисунок 2.5 – Діаграма класів моделей машинного навчання

Фонемінізатор, як було зазначено вище, перетворює текст у фонеми, тому для цієї задачі створимо інтерфейс Phonemizer, який вказуватиме, що його реалізація повинна містити метод get_phonemes, що на вхід приймати об'єкти класу Note та повертати список фонем.

Для англійської мови реалізуємо підтримку epitran [62] та espeak-ng [63], для чого введемо інтерфейс G2PEnglishEngine з методом g2p, який конвертує графеми у фонеми. Класи G2PEpitran та G2PEspeakNG містять конкретну реалізацію зазначеного методу та підтримують epitran і espeak-ng, відповідно.

Також оскільки нот прив'язуються до складів, створимо клас EnglishSyllabificator, який дозволяє поділити текст на склади методом syllabify.

Оскільки японською мовою текст пісень на нотному стані позначається за допомогою хіраґани, для якої існує однозначне відображення у фонеми, то фонемінізатор для цієї мови можемо реалізувати у вигляді словника. Також хіраґана є силабічним алфавітом, відповідно не потрібно реалізовувати поділ на склади як для англійської мови.

Для обох мов додамо класи JapanesePhonemizer та EnglishPhonemizer. Останній включає атритуби типу G2PEnglishEngine та EnglishSyllabificator,

На рисунку 2.6 подано діаграму класів модуля фонемінізації.

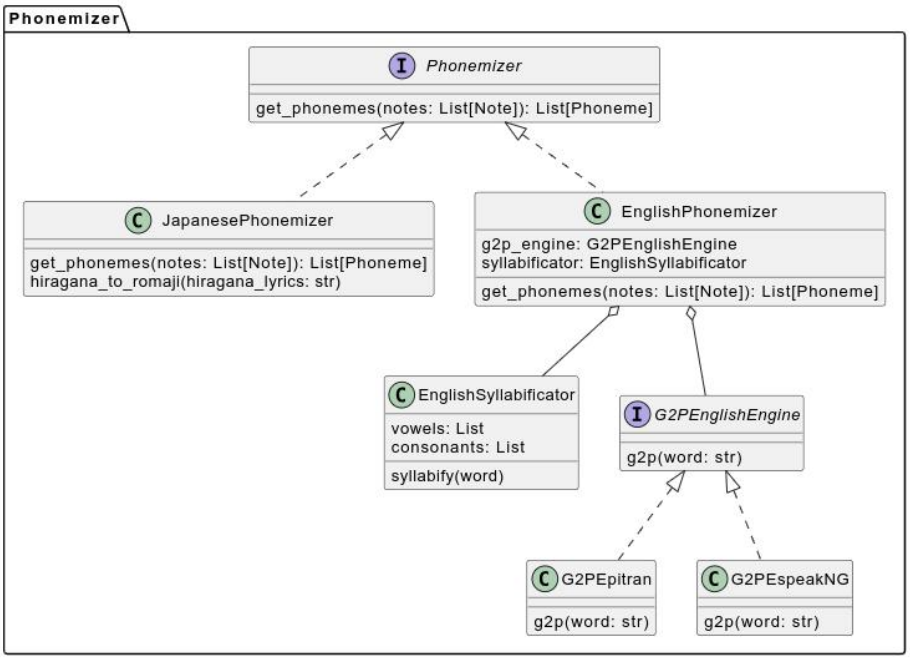

Рисунок 2.6 – Діаграма класів модуля фонемінізації

Клас PartLoader інтерфейс, що визначає контракт для класів, які можуть завантажувати музичні партії. У ньому оголошено метод load, який повертає об'єкт типу Part. Метод може приймати будь-яку кількість аргументів.

Клас MusicXMLLoader, який успадковує PartLoader. Цей клас перевизначає метод load для завантаження файлів формату musicxml [64].

Клас LabLoader успадковує клас PartLoader, дозволяє завантажувати файли формату .midi та .lab.

Клас PartBuilder призначений для побудови об'єкту Part з використанням класів TimeLagModel та DurationModel та фабрики для завантаження компонентів.

Інтерфейс AudioExport описує контракт для експорту аудіо. Він містить метод export, який приймає масив y та частоту дискретизації sr. Клас FileAudioExport реалізує даний інтерфейс та дозволяє зберегти аудіо у файл.

На рисунку 2.7 подано діаграму класів модуля для завантаження та вивантаження даних.

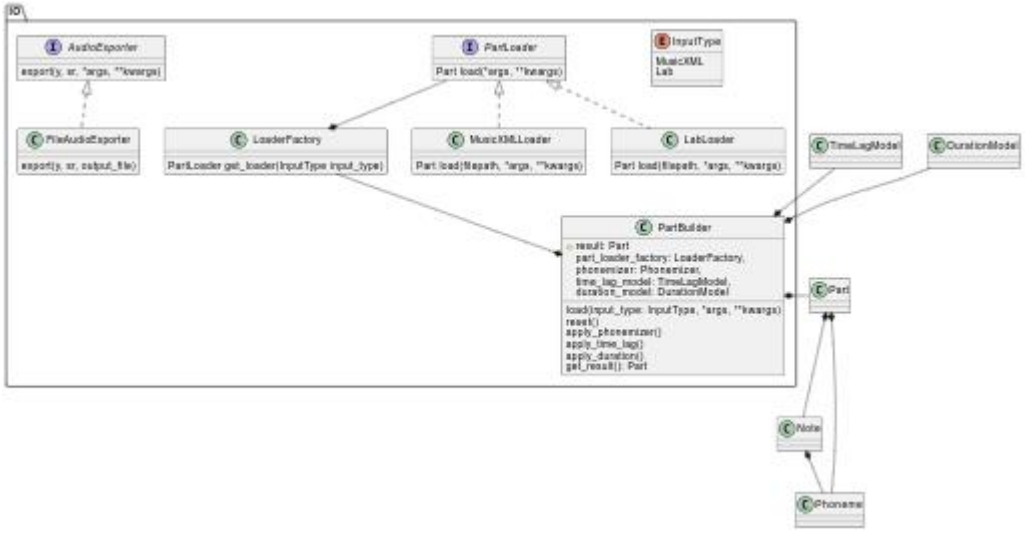

Рисунок 2.7 – Діаграма класів модуля для завантаження та вивантаження даних

Клас SVSEngine є частиною застосунку синтезу вокалу, що об'єднує в себе моделі глибокого навчання та різні етапи попередньої обробки для створення синтезованих співочих голосів. Цей клас створений для опису процесу генерування та об'єднання окремих компонентів у єдиний об'єкт.

На рисунку 2.8 подано діаграму запропонованих класів.

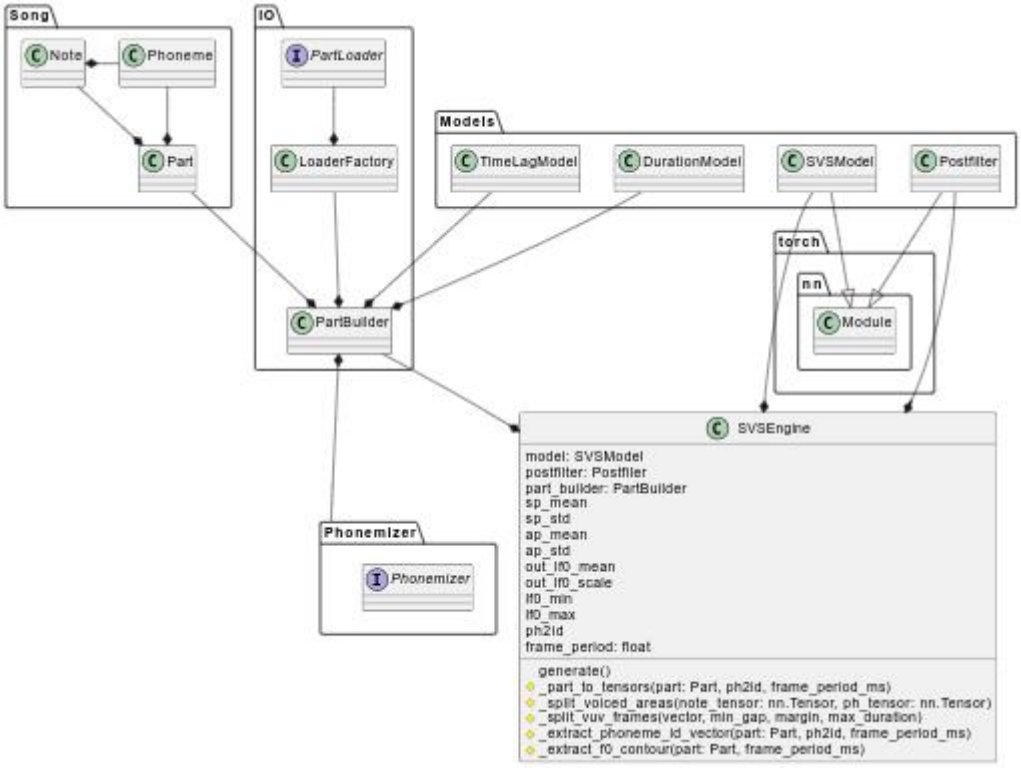

Рисунок 2.8 – Діаграма класів застосунку

SVSEngine реалізує функціональну логіку пропонованого застосунку синтезу вокалу та, як видно на рисунку 2.8, об'єднює інші модулі. Цей клас не є залежним від конкретним моделей глибинного навчання.

2.9 Інтерфейс застосунку синтезу вокалу

Для застосунку пропонується два інтерфейси: консольний та графічний.

Консольний інтерфейс було створено за допомогою argparse, яка є стандартною бібліотекою Python. Користувач може вказати шляхи до папки з вхідними файлами та до папки, куди бажає зберегти отримані результати. Також необхідно вказати формат вхідних даних (musicxml або lab) та мову вхідного тексту (англійську або японську).

Графічний інтерфейс було створено за допомогою gradio [65]. Він пропонує користувачу завантажити файл формату musicxml та вказати мову вхідного тексту. Після натискання кнопки Submit, буде згенеровано аудіофайл з вокалом, який можна прослухати та завантажити.

2.10 Висновок до другого розділу

В другому розділі було описано архітектуру моделі та застосунку синтезу вокалу. Запропонований застосунок відповідає вимогам визначеним у першому розділі.

Описано основні етапи підготовки даних та процес тренування моделі.

РОЗДІЛ 3. ОЦІНЮВАННЯ ЯКОСТІ МОДЕЛІ ТА ТЕСТУВАННЯ ЗАСТОСУНКУ СИНТЕЗУ ВОКАЛУ

3.1 Оцінка прогнозованих значень

На рисунку 3.1 подано згенеровану послідовність lf0, отриману з цільових даних та вхідне значення, яке отримане з нотної партії.

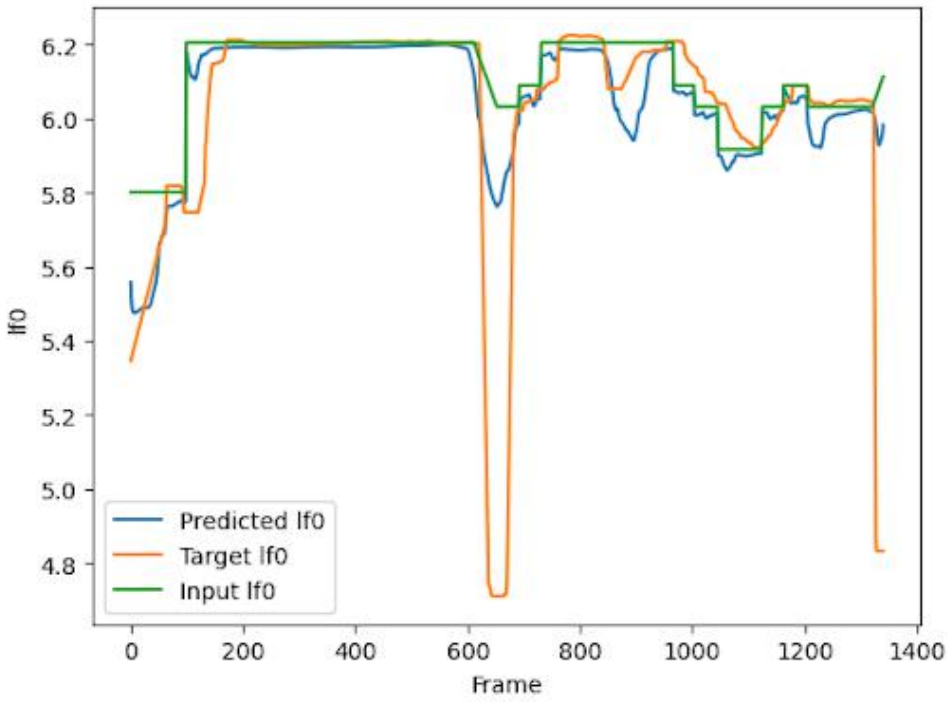

Рисунок 3.1 – Візуалізація значень lf0: згенероване(синя лінія), цільове(помаранчева лінія) та вхідне(зелена лінія)

З рисунка видно, що є декілька ділянок(від 200 до 600 фреймів), де прогнозоване значення lf0 знаходить близько до цільового lf0 (помаранчева лінія), що свідчить про хорошу точність прогнозу на цих ділянках.

Проте існують ділянки(від 1000 до 1200), коли прогнозоване значення lf0 значно відхиляється від цільового lf0 та від вхідного lf0, що демонструє не задовільну точність на цих ділянках.

На рисунку 3.2 подано прогнозовані MGC.

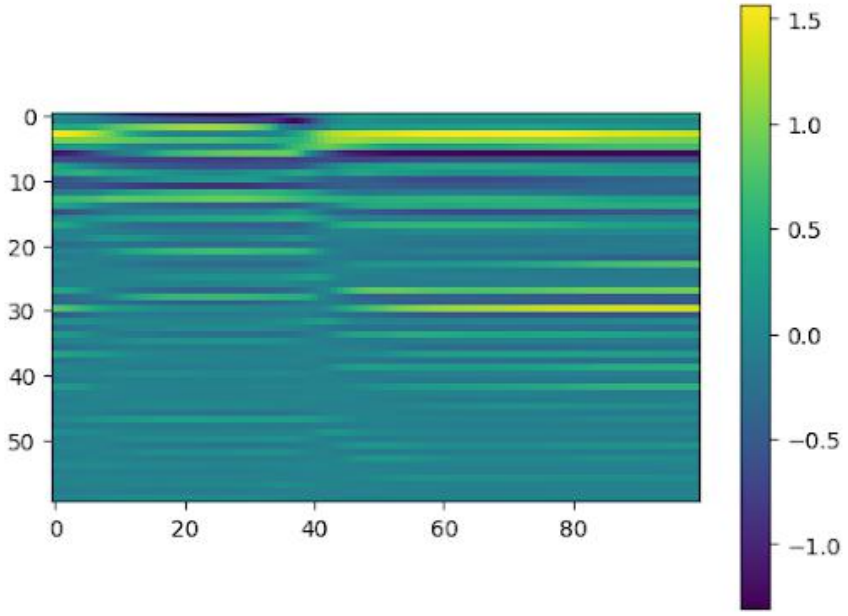

Рисунок 3.2 – Візуалізація прогнозованих MGC

Можемо спостерігати, що прогнозовані MGC є надто гладкими. Для порівняння на рисунку 3.3 подано MGC отриману з цільового аудіо.

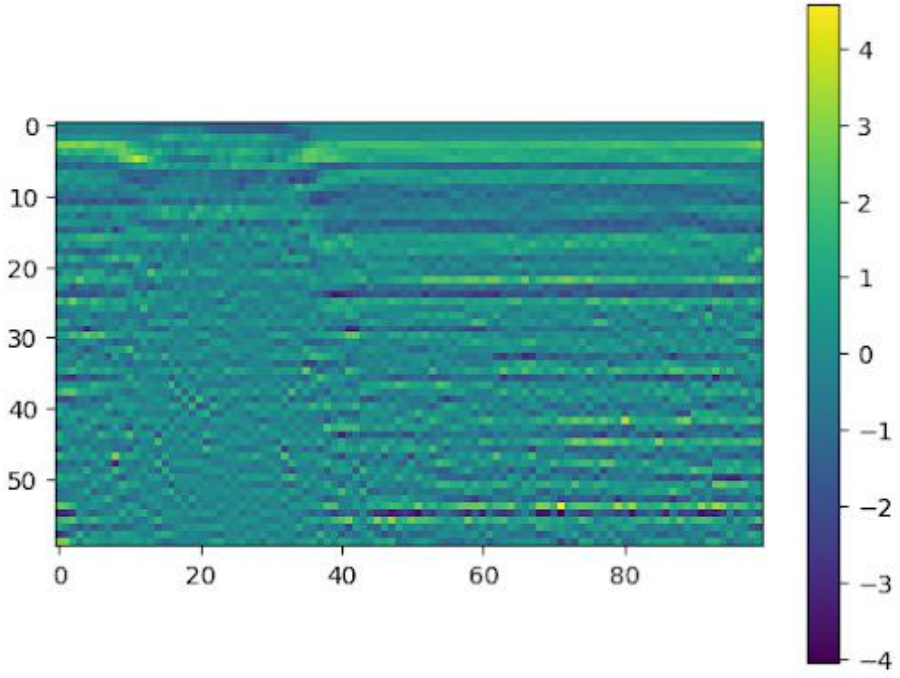

Рисунок 3.3 – Візуалізація цільового значення MGC

Можемо спостерігати, що існує проблема надто гладкого згенерованого MGC. Особливо це помітно для траєкторій від 40 до 59. Цю проблему можна вирішити при застосувані постфільтрів або тренуванні моделі з архітектурою GAN чи на основі дифузійної моделі.

Розглянемо деякі з траєкторій MGC. На рисунку 3.4 зображено прогнозовані та цільові значення нульової траєкторії.

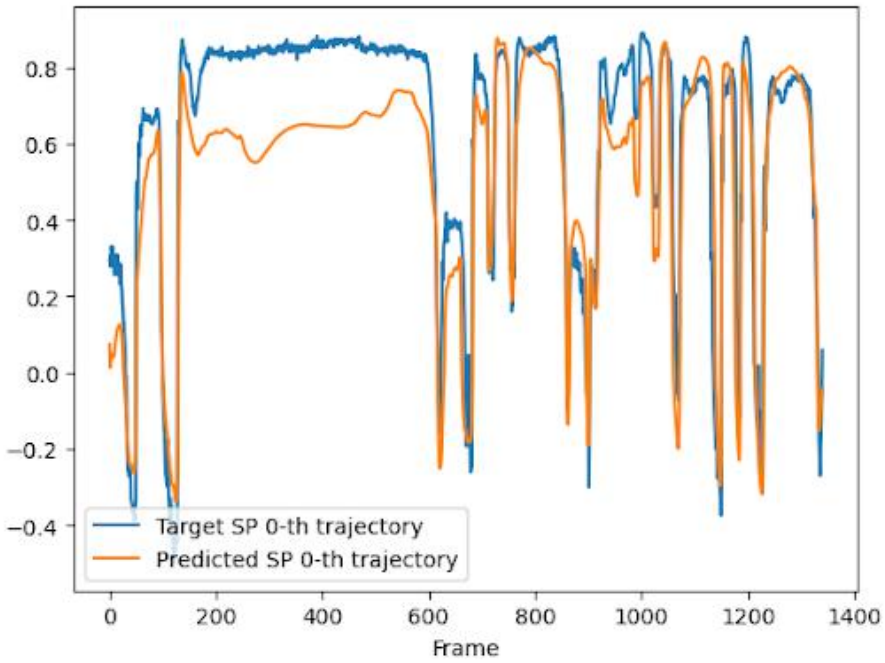

Рисунок 3.4 – Значення нульової траєкторії MGC: цільові (синя лінія) та прогнозовані (помаранчева лінія)

Можемо спостерігати, що прогнозований результат не є ідентичним до цільового, проте має подібні тенденції, що цільові значення. На деяких ділянках (наприклад, на між 200 і 600 фреймами) прогнозована траєкторія демонструє значне відхилення порівняно з цільовою траєкторією.

На рисунку 3.5 подано прогнозовані та цільові значення для першої траєкторії.

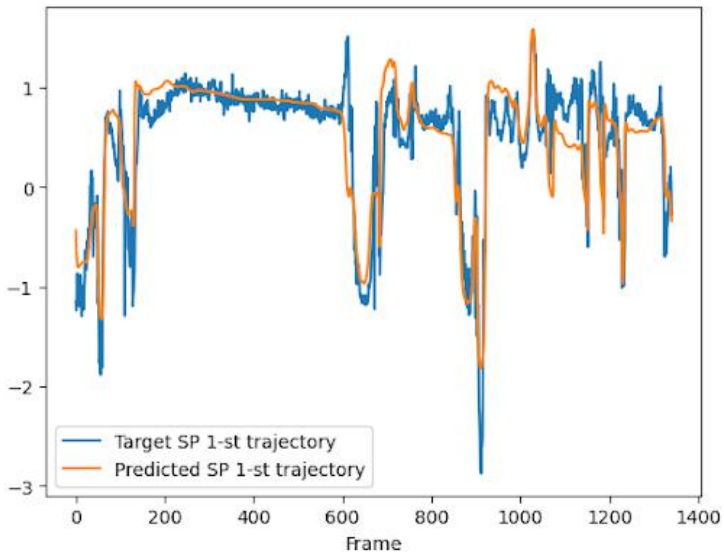

Рисунок 3.5 – Значення першої траєкторії MGC: цільові(синя лінія) та прогнозовані(помаранчева лінія)

З рисунка видно, що для першої траєкторії результат кращий, проте існують ділянки(фрейми від 1100 до 1150, від 1250 до 1300), де прогнозований результат помітно відрізняється від цільового.

На рисунку 3.6 подано прогнозовані та цільові значення для другої траєкторії.

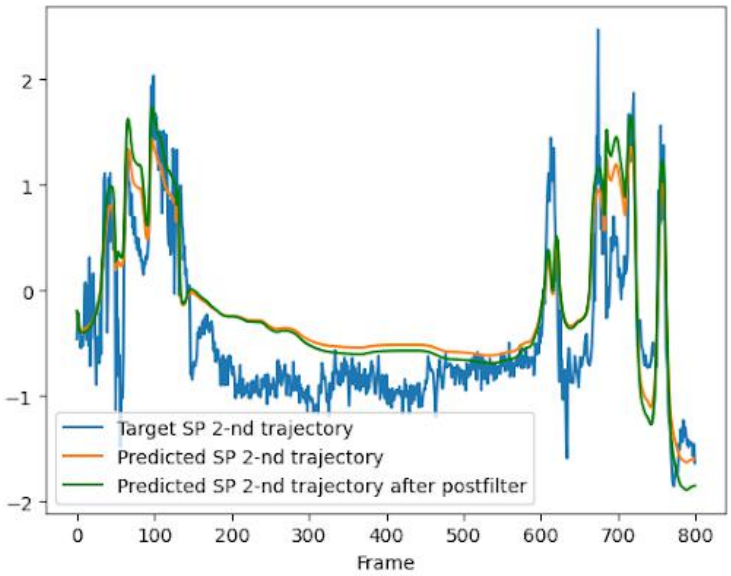

Рисунок 3.6 – Значення другої траєкторії MGC: цільові(синя лінія), прогнозовані до постфільтрації(помаранчева лінія) та після(зелена лінія)

Для першої та другої траєкторій помітно, що цільова траєкторія частіше демонструє швидкі коливання і різкі зміни, тоді як прогнозована траєкторія виглядає більш гладкою і менш мінливою.

Отже, як цільова, так і прогнозована траєкторії, як правило, слідують одній загальній тенденції, що вказує на те, що модель відображає загальну форму і поведінку цільової траєкторії. Однак між ними є помітні розбіжності.

На рисунку 3.7 зображено цільові та прогнозовані значення нульової траєкторію BAP.

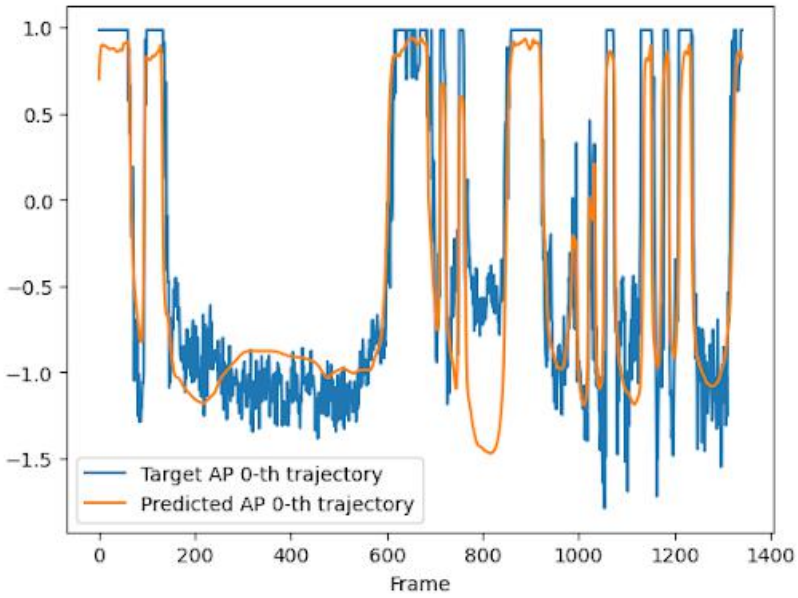

Рисунок 3.7 – Значення нульової траєкторії BAP: Цільові(синя лінія) та згенеровані(помаранчева лінія)

Можемо спостерігати, що прогнозовані значення траєкторії загалом слідують тенденціям цільової траєкторії. Як і для MGC, для BAP спостерігається аналогічна проблема того, що згенерована траєкторія більш гладка, ніж цільова.

На рисунку 3.7 зображено цільові та прогнозовані значення першої траєкторію BAP.

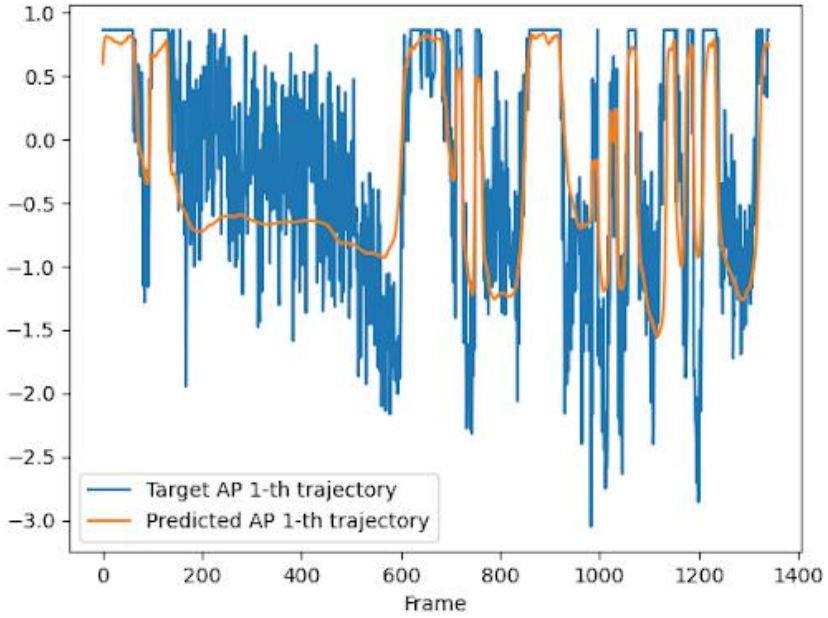

Рисунок 3.8 – Значення першої траєкторії BAP: Цільові(синя лінія) та згенеровані(помаранчева лінія)

Відхилення виглядають більшими за величиною порівняно з нульовою траєкторією. Порівнюючи рисунки 3.7 та 3.8, можемо помітити, що флуктуації цільових значень першої траєкторії більші, ніж нульової.

На рисунку 3.9 подано візуалізацію аудіосигналів для цільового вокалу та прогнозованого.

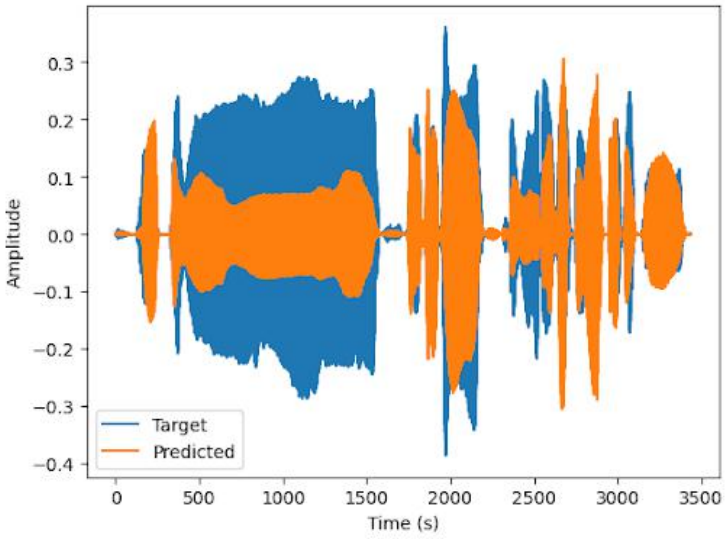

Рисунок 3.9 – Візуалізація аудіосигналу: цільового (синя лінія) та прогнозованого(помаранчева лінія)

З рисунку бачимо, що прогнозований аудіосигнал відтворює голосові атаки, проте там, де треба виконати довшу ноту, наприклад, з 500 до 1500 секунди, можемо спостерігати, що прогнозований аудіосигнал має нижчу амплітуду, ніж цільовий.

Описані вище спостереження щодо прогнозованих значень пропонованою моделлю на основі Residual1DBlock, справедливі також для моделі на основі згорткових шарів.

Розглянемо зміну функції втрат у ході тренування.

На рисунку 3.10 зображено графік зміни функції втрат на валідаційному датасеті залежно від ітерації.

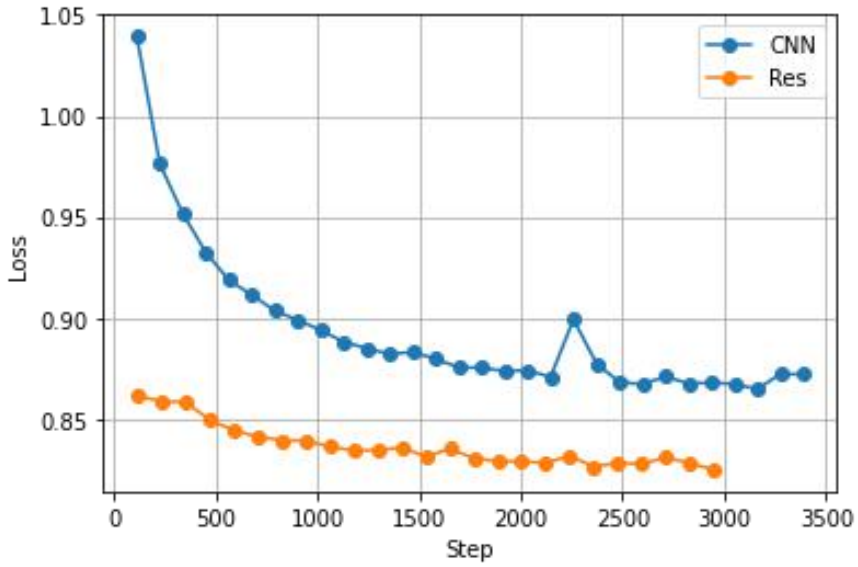

Рисунок 3.10 – Функція втрат на валідаційному датасеті Tohoku Kiritan singing database для моделі на основі Residual1DBlocks (оранжева лінія) та зготкових шарів (синя лінія)

З рисунку видно, що модель на основі Residual1DBlocks збігається швидше, ніж модель на основі згорткових шарів. Кількість тренувальних епох становила 25 для першої моделі та 30 для другої. При цьому перша модель досягнула значення функції втрат 0,82 на валідаційній вибірці, тоді як друга – 0,87.

3.2 Об'єктивне оцінювання

Об'єктивні метрики формуються на основі функції втрат (2.3) визначеної у попередньому розділі, її компонентів (2.4-2.7), кореляції Пірсона для lf0 (lf0 corr), accuracy, precision, recall та f1-score для V/UV (vuv accuracy, vuv precision, vuv recall та vuv f1-score відповідно).

Результат оцінювання подано у таблиці 3.1. Символом "↑" позначено метрики, для яких більше значення є більш бажаним, символом "↓" – менший результат є більш бажаним. Результат подано для обох мов для моделі на основі Residual1DBlock та CNN.

Таблиця 3.1 – Метрики моделі синтезу вокалу

Метрика	**Англійська мова**		**Японська мова**	
	CNN	Residual1D Block	CNN	Residual1D Block
lf0 corr ↑	0,8993	0,8814	0,6488	0,6598
lf0 loss ($\mathcal{L}_f$) ↓	0,2916	0,3092	0,4313	0,4261
sp loss ($\mathcal{L}_{sp}$) ↓	53,7573	53,1168	55,7414	56,1524
ap loss ($\mathcal{L}_{ap}$) ↓	3,5305	3,7429	1,3507	1,5088
vuv loss ($\mathcal{L}_{vuv}$) ↓	0,2667	0,2272	0,2279	0,2141
vuv accuracy ↑	0,9159	0,9304	0,9439	0,9441
vuv precision ↑	0,9577	0,9627	0,9667	0,9741
vuv recall ↑	0,9331	0,9458	0,9637	0,9558
vuv f1-score ↑	0,9449	0,9538	0,9648	0,9646
loss ($\mathcal{L}$) ↓	0,8899	0,8830	0,8885	0,8969

Порівнюючи моделі на основі CNN та Residual1DBlock, можемо помітити, що результат не є однозначним. Для японської мови модель на основі Residual1DBlock є кращою при прогнозуванні lf0, VUV, проте уступає при

прогнозуванні MGC та BAP. Для англійської модель на основі демонструє кращей результат – MGC, VUV.

Якщо порівнювати моделі для англійської та японської мов на основі Residual1DBlock, то можемо помітити кореляція lf0 значно вища для англійської мови (0,8814) порівняно з японською мовою (0,6598) та значення функції втрат lf0 на тестувальній вибірці є меншою для англійської мови (0,3092) порівняно з японською мовою (0,4261). Отже, модель краще прогнозує фундаментальну частоту для англійської мови. По значенню кореляції модель уступає XiaoiceSing2 (0,99) [25] та SinSy (0,97) [34].

При порівнянні $\mathcal{L}_{sp}$ можемо помітити, що для англійської мови результат є кращим на 2-3 одиниці.

$\mathcal{L}_{ap}$ є не порівнянним між мовами, оскільки для англійською BAP представлено вектором розмірністю 5, а для японською – 3.

Метрики для VUV є кращими для японської мови.

Отже, не було помічено суттєвої різниці між моделями на основі CNN та Residual1DBlock. Проте, необхідно враховувати, що модель на основі Residual1DBlock швидше збігається(див. рис. 3.10).

3.3 Суб'єктивне оцінювання

Було проведено 2 опитування.

У першому опитувані 5 респодентам (не носіям японської чи англійської мов) пропонувалося оцінити аудіозапис згенерованого вокалу від 1 до 5. Запропонові приклади не були представлені у тренувальній вибірці. Було запропоновано по 10 записів (по 5 на кожну мову) згенерованих у кожному з наступних застосунків:

1. У застосунку розробленому підчас виконання даної кваліфікаційної роботи, генерація відбувалася без використання постфільтра. Позначимо як ewigeki.

2. У OpenUtau з використання войсбанків Milk [66] та Kasane Teto [67]. Позначимо як milk та kasane відповідно.

3. У Sinsy [68] з використанням голосів “f00001j : Yoko” (позначимо як sinsy1) та “f00002j : Xiang-Ling” (sinsy2).

4. У застосунку Neutrino [14] версії 2.4(позначимо як neutrino).

Учасникам було невідомо якою моделлю згенеровано аудіофайл.

Результати опитування були обробленими за формулами:

$$\tilde{\mu}_m = \frac{1}{\tilde{n}_m} \sum_{i=1}^{\tilde{n}_m} \tilde{s}_{m.i} \tag{3.1}$$

де $\tilde{s}_{m,i}$ – і-та оцінка для m-тої моделі, $\tilde{n}_m$ – кількість оцінок для m-тої моделі, $\tilde{\mu}_m$ – середня оцінка для m-тої моделі;

$$CI_m = \tilde{\mu}_m \pm 1{,}96 \frac{\tilde{\sigma}_m}{\sqrt{\tilde{n}_m}} \tag{3.2}$$

де CI_m – довірчий інтервал для m-тої моделі, $(\tilde{\sigma}_m)^2$ – дисперсія оцінок для m-тої моделі.

Таблиця 3.2 – Результат опитування

Модель	**MOS**
ewigeki (пропонована)	2,54±0,34
neutrino	4,52±0,30
kasane	2,52±0,23
milk	2,96±0,37
sinsy1	4,32±0,25
sinsy2	3,64±0,27

На рисунку 3.10 зображено результати опитування.

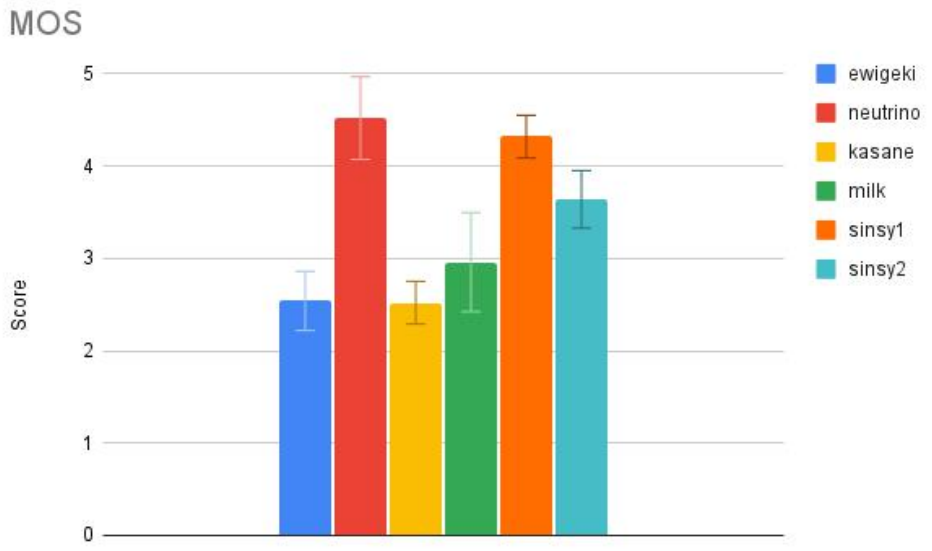

Рисунок 3.10 – Результат оцінювання синтезованого вокалу

У результаті, моделі “neutrino” та “sinsy1” мають найвищі середні оцінки. Моделі “ewigeki” та “kasane” найнижчі. Значення “milk” дещо краще, ніж у “ewigeki”.

Пропонована модель очікувано поступається найкращим моделям. Однак, видно, що її результат є на прийнятному рівні для практичного використання.

Друге опитування полягало в тому, щоб обрати більш привабливий з суб'єктивної точки зору варіант з двох пропонованих з можливістю вказати про відсутність переваги. Метою опитування було порівняння результатів генерації моделей з Residual1DBlock та зі згортковими шарами (обидва варіанти з постфільтрацією).

У опитуванні було 6 учасників, яким не було відомо, якою моделлю згенеровано запропоновані аудіофайли, та які не брали участі у першому опитуванні. Опитування складалося з 5 запитань.

На рисунку 3.11 подано результат другого опитування.

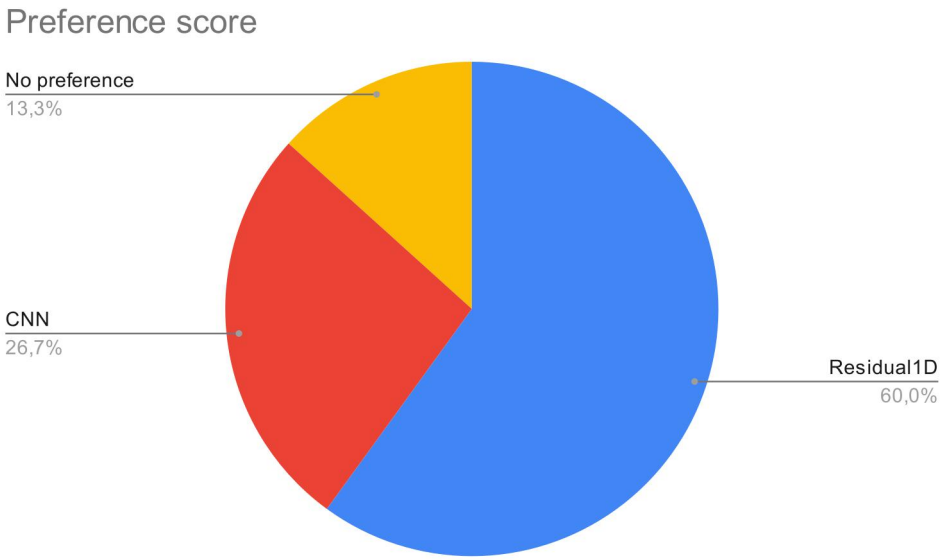

Рисунок 3.11 – Результат опитування щодо вибору більш привабливого варіанту

Можемо спостерігати, що модель на основі Residual1DBlock переважає модель на основі згорткових шарів.

Вважаємо за доцільне в подальшому розширити кількість учасників обох опитувань для отримання більш надійних результатів.

3.4 Тестування функціональності застосунку синтезу вокалу

Переходимо до тестування функціональності застосунку.

Перейдемо на вебсторінку застосунку. Сторінка поділена на дві частини. Зліва розміщено елементи для вводу даних, кнопку “Clear” для очищення даних та кнопку “Submit” для підтвердження генерації. Справа після початку генерації буде показано скільки часу необхідно очікувати до отримання результату.

Попередньо підготуємо файл формату musicxml та завантажимо його на сторінку. Оберемо мову. Після натискання кнопки “Submit” отримаємо згенероване аудіо. Результат подано на рисунку 3.12.

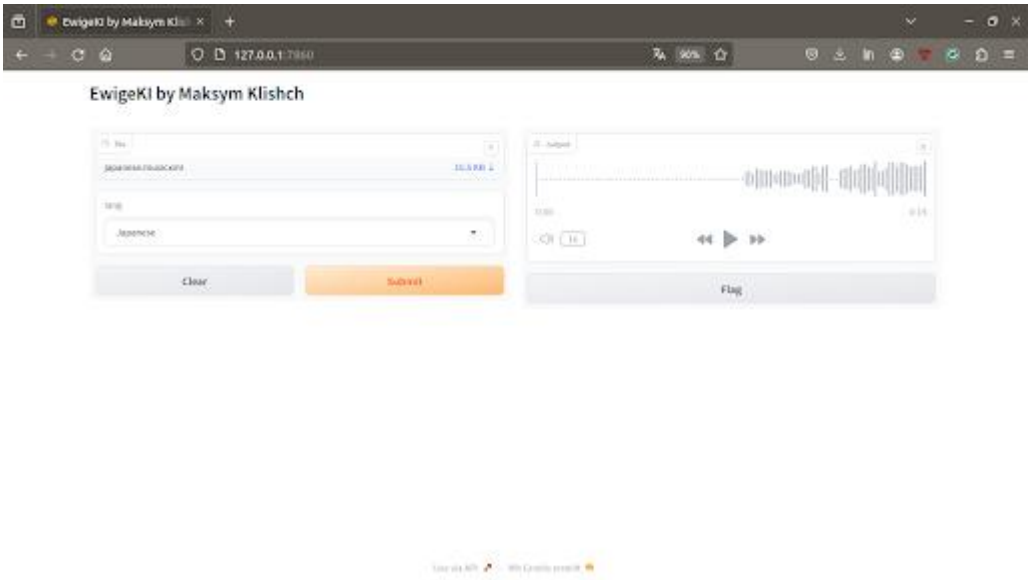

Рисунок 3.12 – Інтерфейс застосунку синтезу вокалу

За допомогою пакета unittest було написано юніт-тести для застосунку. Ці тести перевіряють детерміновані частини застосунку. Для стохастичних функцій вони застосовуються з метою перевірки коректності формату, типу відповідь та інших детермінованих елементів, але не змісту результату.

3.5 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи було описано результати тестування якості пропонованої моделі. Здійснено порівняння з моделлю на основі CNN. На тестувальній вибірці між моделлями не було помічено суттєвої різниця, проте на валідаційній вибірці можемо спостерігати, що модель на основі Residual1DBlock швидше збігається.

Описано процес та результати суб'єктивного оцінювання якості синтезу вокалу. Отриманий застосунок по якості синтезу значно уступає передовим рішенням, проте з урахуванням обмежених ресурсів для тренування отриманий результат є в деякій мірі очікуваним. З цього погляду, відкривається потенціал для подальших покращень і оптимізацій, що можуть сприяти підвищенню якості та ефективності розробленого рішення.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Фізіологічний та психологічний вплив синтезованого вокалу на життєдіяльність людини

Вплив синтезованого вокалу на життєдіяльність людини можна розглядати як з фізіологічного, так і з психологічного боку.

Оскільки відсутні дослідження, що розглядають безпосередньо фізіологічний та психологічний вплив синтезованого вокалу на життєдіяльність людини, розглянемо вплив шуму, музики та реального вокалу.

Синтезований вокал, особливо якщо він звучить неприродно або монотонно, може викликати напругу слухового апарату. Це може призводити до втоми та перевантаження слуху, особливо при тривалому прослуховуванні. Важливо враховувати характеристики синтезованого вокалу, такі як тональність, частота та інтенсивність звуку, оскільки вони можуть впливати на комфорт слухового сприйняття.

Деякі дослідження показують, що звуки можуть впливати на фізіологічні параметри, такі як частота серцевих скорочень, артеріальний тиск та рівень стресу. Синтезований вокал, як і природні звуки, може мати подібний вплив, залежно від його характеристик (тональність, інтенсивність тощо).

Зокрема у наукових публікаціях продемонстровано вплив музики на:

– рівень стресу [69];
– впливати на частоту дихання [70];
– артеріальний тиск [70];
– частота серцевих скорочень [70].

Отже, важливо враховувати характеристики синтезованого вокалу, такі як тональність, частота, інтенсивність та природність звучання, оскільки вони можуть впливати на комфорт слухового сприйняття та фізіологічні процеси в організмі людини. Тривале прослуховування неприродного або монотонного синтезованого вокалу може призводити до негативних наслідків, таких як втома,

перевантаження слуху, зміни в частоті серцевих скорочень, артеріальному тиску та рівні стресу.

У роботі [71] було запропоновано гіпотезу про ефект моторошної долини (англ. Uncanny Valley Effect). Запропонований ефект виникає, коли штучні об'єкти (у тому числі аудіо [72]) стають дуже схожими на людські, але все ж мають невеликі відмінності, що викликають дискомфорт або відчуття тривожності у людей. Це може знижувати рівень прийняття застосунку синтезу вокалу.

Проте, достовірність цього ефекту для синтезованого вокалу та мовлення в загальному є предметом дискусій у науковій спільноті.

Наприклад, у роботі [73] продемонстровано, що учасники опитування надавали перевагу більш реалістичному синтезованому голосу. У роботі [74] зазначають, що сучасні системи синтезу голосу успішно нівелюють ефект моторошної долини.

Отже, вплив синтезованого вокалу на людину є неоднозначним і залежить від особливостей використання, особистого сприйняття та потреб кожного індивіда.

4.2 Заходи щодо зниження ризиків для оператора ПК при роботі із застосунком синтезу вокалу

Робота зі застосунком синтезу вокалу пов'язана з тривалим прослуховування аудіофайлів, що потенційно може привести до погіршення слуху [75]. Це впливатиме на безпеку здоров'я користувачів та тестувальників.

У [76] зазначає Всесвітня організація охорони здоров'я, що безпечність прослуховування залежить від 3 факторів:

1. Гучності звуку. Висока гучність може бути шкідливою для слуху, особливо якщо вона триває тривалий час. Прослуховування музики (у тому числі синтезованого вокалу) або інших звуків на високому рівні гучності через навушники може призвести до пошкодження слуху.

2. Тривалості. Навіть якщо гучність звуку не надто висока, тривале прослуховування може бути також шкідливим для слуху.

3. Частоти відтворення.

Гучність звуку, яку користувачі можуть безпечно слухати, залежить від тривалості впливу.

У [77] вказують на необхідність вставнолення максимально допустимого рівня звуку на пристрої до значення 80 дБА для дорослих та 75 дБА для дітей при використані пристрою 40 годин на тиждень.

Міністерством охорони здоров'я України визначено у [78] санітарні норми, що регуюють питання допустимого рівня шуму. Для творчої діяльності, наукової діяльності допустимий рівень шуму становить 50 дБА.

Також у [79] зазначено, що допустимий рівень шуму, який проникає в житлові приміщення, та рівень шум на території житлової забудови для музичних класів становить 35 дБА, а для залів театрів та концертних залів – 30 дБА.

Проте, часто прослуховування відбування з використанням персональний пристроїв на потокових платформах як Spotify або Youtube Music. Ці платформи використовують нормалізацію гучності до певного рівня. Наприклад, Spotify пропонує значення -14 дБ LUFS [80].

У [81] Всесвітня організація охорони здоров'я наводить наступні рекомендації для заходів та подій:

1. Встановлення обмеження на рівень шуму. У роботі пропонується верхнє обмеження на рівні 100 дБА. Це дозволить знизити ризик пошкодження слуху при тривалому впливі гучних звуків. Встановлення такого обмеження може бути реалізовано через регулювання звукового обладнання або програмне забезпечення.

2. Спостереження за рівнем шуму. Постійний моніторинг рівня шуму під час роботи із застосунком дозволяє виявляти перевищення допустимих рівнів і вживати необхідних заходів для їх зниження.

3. Системи виведення звуку повинні забезпечувати безпеку прослуховування. Використання сучасних звукових систем, які мають вбудовані обмежувачі рівня гучності та можливість точно налаштовувати звукові параметри, допоможе забезпечити безпечне прослуховування. Такі системи дозволяють уникнути раптових піків гучності і забезпечують рівномірний розподіл звуку.

4. Доступ до тихих зон. Вони дозволяють відпочити від гучних звуків, що сприяє зниженню загального навантаження на слуховий апарат.

5. Проведення навчань та інформування щодо безпечного прослуховування. Надання попередження про можливі ризики. Інформування учасників про небезпеки гучного звуку та способи захисту слуху є важливим кроком для запобігання пошкодженню слуху. Це може включати розповсюдження інформаційних матеріалів, проведення навчальних семінарів або показ відеороликів. Також це може бути частиною інструктажів з питань охорони праці.

Зазначені рекомендації вважаємо також доречними і для користувачів застосунку синтезу вокалу.

Для забезпечення здоров'я користувачів та тестувальників рекомендується регулярні перерви – наприклад, через кожні 30 хвилин роботи робити 5 хвилин перерви. Це допоможе знизити ризик накопичення навантаження на слуховий апарат і дозволить уникнути пошкодження слуху.

Врахування зазначених рекомендацій дозволить забезпечити безпечне використання застосунків синтезу вокалу, зменшуючи ризики для здоров'я користувачів і тестувальників.

4.3 Висновок до четвертого розділу

В четвертому розділі кваліфікаційної роботи розглянуто вплив синтезованого вокалу на фізіологію та психологію людини, рекомендації щодо безпечного прослуховування під час роботи із застосунком.

Робота зі застосунком синтезу вокалу вимагає особливої уваги до питань збереження слуху, адже тривале прослуховування аудіофайлів може призвести до його погіршення. Важливими аспектами для забезпечення безпеки користувачів є контроль гучності, тривалості та частоти відтворення звуку. Встановлення обмежень на рівень шуму та використання пристроїв з функціями автоматичного зниження гучності сприятиме збереженню слуху. Застосування рекомендацій Всесвітньої організації охорони здоров'я, таких як спостереження за рівнем шуму, забезпечення доступу до тихих зон та проведення навчань щодо безпечного прослуховування, є також актуальним для користувачів застосунку синтезу вокалу.

Організація робочого місця, яка включає тихі зони для відпочинку та використання якісних навушників з функцією шумозаглушення, може значно знизити негативний вплив на слуховий апарат. При роботі із застосунком синтезу вокалу слід уважно ставитися до питань збереження слуху, дотримуватися санітарних норм та впроваджувати рекомендації щодо безпечного прослуховування, що дозволить знизити ризики для здоров'я та підвищити комфорт роботи з аудіоматеріалами.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено застосунок синтезу вокалу, вибрано архітектуру моделі.

Отримана архітектура застосунку мінімально залежить від особливостей конкретної мови співу та вхідного тексту. Добавлення нових мов може відбуватися через розширення системи класів через успадкування існуючих інтерфейсів. Аналогічним чином було реалізовано агностицизм до моделей глибинного навчання.

Запропоновано модель на основі [34] та продемонстровано, що вона збігається швидше, ніж аналогічна зі згортковими шарами замість залишкових, при цьому не поступається в якості прогнозування.

В першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

– Описано існуючі застосунки синтезу вокалу.

– Розглянуто існуючі методи синтезу вокалу на основі глибинного навчання.

– Визначено вимоги до застосунку.

В другому розділі кваліфікаційної роботи:

– Обґрунтовано архітектуру моделі синтезу вокалу.

– Описано архітектуру застосунку синтезу вокалу.

В третьому розділі кваліфікаційної роботи:

– Описано тестування застосунку синтезу вокалу.

– Оцінено якість отриманої моделі синтезу вокалу.

У розділі «Безпека життєдіяльності, основи охорони праці» обговорено вплив синтезованого вокалу на фізіологію та психологію людини, надаються рекомендації щодо безпечного прослуховування при його використанні.

ПЕРЕЛІК ДЖЕРЕЛ

1 I. Strutynska, H. Kozbur, L. Dmytrotsa, O. Sorokivska, L. Melnyk and R. Grytseliak, "Regarding to the Concept of Small and Medium-Sized Enterprises Digitalization in Ukraine: Problems and Solutions," *2021 11th International Conference on Advanced Computer Information Technologies (ACIT)*, Deggendorf, Germany, 2021, pp. 276-279, doi: 10.1109/ACIT52158.2021.9548382.

2 I. Strutynska, L. Dmytrotsa, H. Kozbur, L. Melnyk, and R. Sherstiuk, “The unification of approaches to measuring the digital maturity of business structures (international and domestic approaches).,” in *ICTERI*, pp. 10–23, 2021.

3 Л. Мосій, І. Струтинська та Г. Козбур, “Роль комп'ютерно-інформаційних технологій у цифровій трансформації економіки,” *ФОП Паляниця ВА*, pp. 432-434, 2023.

4 P. Lu, J. Wu, J. Luan, X. Tan, and L. Zhou, “Xiaoicesing: A high-quality and integrated singing voice synthesis system,” *arXiv preprint arXiv:2006.06261*, 2020.

5 Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, “FastSpeech: Fast, Robust and Controllable Text to Speech,” *arXiv preprint arXiv:1905.09263*, 2019.

6 Y. Gu, X. Yin, Y. Rao, Y. Wan, B. Tang, Y. Zhang, J. Chen, Y. Wang, and Z. Ma, “ByteSing: A Chinese Singing Voice Synthesis System Using Duration Allocated Encoder-Decoder Acoustic Models and WaveRNN Vocoders,” *arXiv preprint arXiv:2004.11012*, 2020.

7 Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, “Tacotron: Towards end-to-end speech synthesis,” *arXiv preprint arXiv:1703.10135*, 2017.

8 V. Lanzrein and R. Cross, The Singer’s Guide to German Diction. Oxford University Press, 2018.

9 “VOCALOID – the modern singing synthesizer – vocaloid.com.” https://www.vocaloid.com/en/. Дата звернення: 6 черв. 2024. [Онлайн]

10 H. Kenmochi and H. Ohshita, “Vocaloid-commercial singing synthesizer based on sample concatenation.,” in *Interspeech*, vol. 2007, pp. 4009–4010, 2007.

11 “Singing voice synthesis tool UTAU download page – utau2008.xrea.jp.” https://utau2008.xrea.jp/. Дата звернення: 3 черв. 2024. [Онлайн]

12 “Home – openutau.com.” https://www.openutau.com/. Дата звернення: 3 черв. 2024. [Онлайн].

13 “Synthesizer V – Dreamtonics – dreamtonics.com.” https://dreamtonics.com/synthesizerv/. Дата звернення: 4 черв. 2024. [Онлайн].

14 “NEUTRINO – Neural singing synthesizer – studio-neutrino.com.” https://studio-neutrino.com/. Дата звернення: 4 черв. 2024. [Онлайн4].

15 “synsinger – synthetic singing for the masses.” https://synsinger.wordpress.com/. Дата звернення: 4 черв. 2024. [Онлайн].

16 R. Huang, C. Cui, F. Chen, Y. Ren, J. Liu, Z. Zhao, B. Huai, and Z. Wang, “SingGAN: Generative adversarial network for high-fidelity singing voice generation,” *arXiv preprint arXiv:2110.07468*, 2021.

17 I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *arXiv preprint arXiv:1406.2661*, 2014.

18 J. Liu, C. Li, Y. Ren, F. Chen, and Z. Zhao, “Diffsinger: Singing voice synthesis via shallow diffusion mechanism,” *arXiv preprint arXiv:2105.02446*, 2021.

19 J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *arXiv preprint arXiv:2006.11239*, 2020.

20 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *arXiv preprint arXiv:1706.03762*, 2017.

21 M. Morise, F. Yokomori, and K. Ozawa, “World: a vocoder-based high-quality speech synthesis system for real-time applications,” *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

22 M. Morise, “D4c, a band-aperiodicity estimator for high-quality speech synthesis,” *Speech Communication*, vol. 84, pp. 57–65, 2016.

23 J. Chen, X. Tan, J. Luan, T. Qin, and T.-Y. Liu, “Hifisinger: Towards high-fidelity neural singing voice synthesis,” *arXiv preprint arXiv:2009.01776*, 2020.

24 R. Yamamoto, E. Song, and J.-M. Kim, “Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram,” *arXiv preprint arXiv:1910.11480*, 2019.

25 C. Wang, C. Zeng, and X. He, “Xiaoicesing 2: A high-fidelity singing voice synthesizer based on generative adversarial network,” *arXiv preprint arXiv:2210.14666*, 2022.

26 J. Kim, H. Choi, J. Park, S. Kim, J. Kim, and M. Hahn, “Korean singing voice synthesis system based on an lstm recurrent neural network,” in *Proc. Interspeech*, pp. 1551–1555, 2018.

27 S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

28 Y. Ren, X. Tan, T. Qin, J. Luan, Z. Zhao, and T.-Y. Liu, “Deepsinger: Singing voice synthesis with data mined from the web,” *arXiv preprint arXiv:2007.04590*, 2020.

29 H. Tachibana, K. Uenoyama, and S. Aihara, “Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Apr. 2018.

30 K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, “Speech synthesis based on hidden markov models,” *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

31 M. Nishimura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “Singing voice synthesis based on deep neural networks.,” in *Interspeech*, pp. 2478–2482, 2016.

32 Y. Hono, S. Murata, K. Nakamura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “Recent development of the dnn-based singing voice synthesis

system – sinsy," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1003–1009, IEEE, 2018.

33 C. M. Bishop, "Mixture density networks," *Aston University*, 1994.

34 Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Sinsy: A deep neural network-based singing voice synthesis system," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, p. 2803–2815, 2021.

35 S.-Z. Yu, "Hidden semi-markov models," *Artificial intelligence*, vol. 174, no. 2, pp. 215–243, 2010.

36 K. Nakamura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Singing voice synthesis based on convolutional neural networks," *arXiv preprint arXiv:1904.06868*, 2019.

37 M. Nishihara, Y. Hono, K. Hashimoto, Y. Nankaku, and K. Tokuda, "Singing voice synthesis based on frame-level sequence-to-sequence models considering vocal timing deviation," *arXiv preprint arXiv:2301.02262*, 2023.

38 J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," *arXiv preprint arXiv:1712.05884*, 2017.

39 J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *arXiv preprint arXiv:1506.07503*, 2015.

40 Y. Zhang, J. Cong, H. Xue, L. Xie, P. Zhu, and M. Bi, "Visinger: Variational inference with adversarial learning for end-to-end singing voice synthesis," *arXiv preprint arXiv:2110.08813*, 2021.

41 J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," *arXiv preprint arXiv:2106.06103*, 2021.

42 Y. Zhang, H. Xue, H. Li, L. Xie, T. Guo, R. Zhang, and C. Gong, "Visinger 2: High-fidelity end-to-end singing voice synthesis enhanced by digital signal processing synthesizer," *arXiv preprint arXiv:2211.02903*, 2022.

43 J. Engel, L. Hantrakul, C. Gu, and A. Roberts, “DDSP: Differentiable digital signal processing,” *arXiv preprint arXiv:2001.04643*, 2020.

44 R. Yamamoto, R. Yoneyama, and T. Toda, “Nnsvs: A neural network-based singing voice synthesis toolkit,” *arXiv preprint arXiv:2210.15987*, 2022.

45 J. Shi, S. Guo, T. Qian, N. Huo, T. Hayashi, Y. Wu, F. Xu, X. Chang, H. Li, P. Wu, S. Watanabe, and Q. Jin, “Muskits: an end-to-end music processing toolkit for singing voice synthesis,” in *Proceedings of Interspeech*, pp. 4277– 4281, 2022.

46 H. Zen, K. Tokuda, and A. W. Black, “Statistical parametric speech synthesis,” *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

47 K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, “An hmm-based singing voice synthesis system,” in *Ninth International Conference on Spoken Language Processing*, 2006.

48 K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *arXiv preprint arXiv:1512.03385*, 2015.

49 S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015

50 I. Ogawa and M. Morise, “Tohoku kiritan singing database: A singing database for statistical parametric singing synthesis using japanese pop songs,” *Acoustical Science and Technology*, vol. 42, no. 3, pp. 140–145, 2021.

51 S. Choi, W. Kim, S. Park, S. Yong, and J. Nam, “Children’s song dataset for singing voice research,” in *International Society for Music Information Retrieval Conference (ISMIR)*, vol. 4, 2020.

52 M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, “Montreal forced aligner: Trainable text-speech alignment using kaldi.,” in *Interspeech*, vol. 2017, pp. 498–502, 2017.

53 C. Raffel and D. P. Ellis, “Intuitive analysis, creation and manipulation of midi data with pretty midi,” in *15th International society for music information retrieval conference late breaking and demo papers*, pp. 84–93, 2014.

54 C. J. Steinmetz and J. Reiss, “pyloudnorm: A simple yet flexible loudness meter in python,” in *Audio Engineering Society Convention 150*, Audio Engineering Society, 2021.

55 J. Hsu, “GitHub - JeremyCCHsu/Python-Wrapper-for-World-Vocoder: A Python wrapper for the high-quality vocoder ”World” – github.com.” https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder. Дата звернення: 5 черв. 2024. [Онлайн].

56 M. Morise, G. Miyashita, and K. Ozawa, “Low-dimensional representation of spectral envelope without deterioration for full-band speech analysis/synthesis system.,” in *INTERSPEECH*, pp. 409–413, 2017.

57 “Google Colab – colab.research.google.com.” https://colab.research.google.com/. Дата звернення: 5 черв. 2024. [Онлайн].

58 I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.

59 H. Silén, E. Helander, J. Nurminen, and M. Gabbouj, “Ways to implement global variance in statistical speech synthesis.,” in *Interspeech*, pp. 1436–1439, 2012.

60 J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. Luk, B. Maher, Y. Pan, C. Puhrsch, M. Reso, M. Saroufim, M. Y. Siraichi, H. Suk, M. Suo, P. Tillet, E. Wang, X. Wang, W. Wen, S. Zhang, X. Zhao, K. Zhou, R. Zou, A. Mathews, G. Chanan, P. Wu, and S. Chintala, “PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation,” in 29th *ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, Volume 2 (ASPLOS ’24), ACM, Apr. 2024.

61 W. Falcon and The PyTorch Lightning team, “PyTorch Lightning.” https://github.com/Lightning-AI/lightning, Mar. 2019. Дата звернення: 5 черв. 2024. [Онлайн].

62 D. R. Mortensen, S. Dalmia, and P. Littell, “Epitran: Precision G2P for many languages,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (N. C. C. chair), K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, eds.), (Paris, France), European Language Resources Association (ELRA), May 2018.

63 “Espeak NG text-to-speech.” https://github.com/espeak-ng/espeak-ng. Дата звернення: 6 черв. 2024. [Онлайн].

64 “MusicXML for Exchanging Digital Sheet Music – musicxml.com.” https://www.musicxml.com/. Дата звернення: 6 черв. 2024. [Онлайн].

65 “Gradio.” https://www.gradio.app/. Дата звернення: 6 черв. 2024. [Онлайн].

66 “Home – Milk – xepheris.wixsite.com.” https://xepheris.wixsite.com/milk. Дата звернення: 6 черв. 2024. [Онлайн].

67 “Kasane Teto Official website – kasaneteto.jp.” https://kasaneteto.jp/utau/. Дата звернення: 6 черв. 2024. [Онлайн].

68 “Sinsy – HMM/DNN-based Singing Voice Synthesis System – sinsy.jp.” https://www.sinsy.jp/. Дата звернення: 6 черв. 2024. [Онлайн].

69 M. V. Thoma, R. La Marca, R. Bronnimann, L. Finkel, U. Ehlert, and U. M. Nater, “The effect of music on the human stress response,” *PloS one*, vol. 8, no. 8, p. e70156, 2013.

70 L. Bernardi, C. Porta, and P. Sleight, “Cardiovascular, cerebrovascular, and respiratory changes induced by different types of music in musicians and non-musicians: the importance of silence,” *Heart*, vol. 92, no. 4, pp. 445–452, 2006.

71 M. Masahiro, “The uncanny valley,” Energy, vol. 7, p. 33, 1970.

72 M. Avdeeff, “Artificial intelligence & popular music: Skygge, flow machines, and the audio uncanny valley,” in *Arts*, vol. 8, p. 130, MDPI, 2019.

73 J. Romportl, “Speech synthesis and uncanny valley,” in *International conference on text, speech, and dialogue*, pp. 595–602, Springer, 2014.

74 A. Diel and M. Lewis, “The vocal uncanny valley: Deviation from typicalorganic voices best explains uncanniness.,” 2023.

75 Грибан Г. В. Охорона праці / Г. В. Грибан, О. В. Негодченко. – Київ: Центр учбової літератури, 2009. – 280 с.

76 World Health Organization, “Make listening safe,” World Health Organization, 2021. Доступно: https://cdn.who.int/media/docs/default-source/documents/health-topics/deafness-and-hearing-loss/mls-brochure-english-2021.pdf

77 World Health Organization, “Safe listening devices and systems: a WHO-ITU standard,” 2019. Доступно: https://www.who.int/publications/i/item/9789241515276

78 Україна, МОЗ України. (1999, 1 груд.). Постанова, Норми № 37, *Санітарні норми виробничого шуму, ультразвуку та інфразвуку ДСН 3.3.6.037-99*. Дата звернення: 7 черв. 2024. [Онлайн]. Доступно: https://zakon.rada.gov.ua/rada/show/va037282-99

79 Україна, Міністерство охорони здоров'я України. (2019, 22 лют.). *Наказ Міністерства охорони здоров'я України № 463, Про затвердження Державних санітарних норм допустимих рівнів шуму в приміщеннях житлових та громадських будинків і на території житлової забудови*. Дата звернення: 7 черв. 2024. [Онлайн]. Доступно: https://zakon.rada.gov.ua/laws/show/z0281-19#Text

80 “Loudness normalization - Spotify”. Spotify. Дата звернення: 7 черв. 2024. [Онлайн]. Доступно: https://support.spotify.com/us/artists/article/loudness-normalization/

81 World Health Organization, WHO global standard for safe listening venues and events. World Health Organization, 2022. Доступно: https://www.who.int/publications/i/item/9789240043114