

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

«На правах рукопису»
УДК 004.852

«До захисту допущено»

Завідувач кафедри

_____ Едуард ЖАРІКОВ

«___» _____ 2022 р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-професійною програмою «Інженерія програмного
забезпечення комп'ютерних систем»**

зі спеціальності 121 «Інженерія програмного забезпечення»

на тему: «Методи та програмне забезпечення

**автоматизованого озвучування іншомовного відео з урахуванням
емоційної**

складової мовлення»

Виконав:

студент II курсу, групи ІТ-04мп
Клярський Кирило Андрійович

Керівник:

Ст. викл, проф.
Стеценко Інна Вячеславівна

Рецензент:

доцент кафедри ІСТ, д.т.н., доц.,
Клименко Ірина Анатоліївна

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2022 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Рівень вищої освіти – другий (магістерський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення комп'ютерних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Едуард ЖАРІКОВ

«___» _____ 2022р.

**ЗАВДАННЯ
на магістерську дисертацію студенту
Клярському Кирилу Андрійовичу**

1. Тема дисертації «Методи та програмне забезпечення автоматизованого озвучування іншомовного відео з урахуванням емоційної складової мовлення», науковий керівник дисертації Стеценко Інна Вячеславівна, ст. викл, проф., затверджені наказом по університету від «24» квітня 2022 р. № 88
2. Термін подання студентом дисертації «6» червня 2022 р.
3. Об'єкт дослідження – іноземні відео
4. Предмет дослідження – методи та засоби для автоматизованого перекладу іноземних відео
5. Перелік завдань, які потрібно розробити – проаналізувати предметну область, привести проектні рішення з автоматизованої адаптації іншомовних відео, розробити програмне забезпечення по атвоматизованому озвучуванню перекладеного тексту з відео, розробити стартап-проект.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу – Схема структурна бізнес-процесу автоматизований переклад відео.

7. Орієнтовний перелік публікацій – «Комп’ютерні інтелектуальні системи та мережі. Матеріали XV Всеукраїнської науково практичної WEB конференції аспірантів, студентів та молодих вчених (22-24 березня 2022 р.). – Кривий Ріг: Криворізький національний університет, 2022. – ст.106-109»

8. Дата видачі завдання «30» вересня 2020 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання	Примітка
1	Аналіз предметної області	01.06.2021	
2	Порівняння проектних рішень з автоматизованої адаптації іншомовних відео	01.08.2021	
3	Розробка програмного забезпечення по атвоматизованому озвучуванню перекладеного тексту з відео	01.10.2021	
4	Розробка стартап-проекту	01.12.2021	
5	Виконання експериментальних досліджень	01.02.2022	
6	Оформлення пояснювальної записки	01.03.2022	
7	Подання дисертації на попередній захист	22.04.2022	
8	Подання дисертації на захист	10.05.2022	

Студент

Кирило КЛЯРСЬКИЙ

Науковий керівник

Інна СТЕЦЕНКО

РЕФЕРАТ

Розмір пояснювальної записки – 90 аркушів, містить 34 ілюстрації, 21 таблицю, 2 додатки.

Актуальність теми. У роботі розглянуто проблему в області обробки природних мов, а саме – проблему з іноземними відео. Показано основні особливості існуючих рішень проблеми, їх переваги та недоліки. А саме, показано, що на сьогодні широко використовується у комерційних цілях тільки автоматизований переклад субтитрів. Недоліки у такому підході у першу чергу пов'язані із увагою прослуховувача та кількістю зрозумілого. Виявлено потребу у розробці нового підходу у сприйнятті іншомовних відео людьми, які цю мову не знають, а саме, додатку для автоматизованого перекладу саме аудіодоріжки оригінального відео. У роботі також робиться акцент на емоційній складовій мовлення, як дуже важливого аспекту мови. Завдяки ньому передаються більше інформації.

Мета дослідження. Основною метою є покращити сприйняття іноземних відео.

Об'єкт дослідження: іноземні відео.

Предмет дослідження: методи та засоби для автоматизованого перекладу іноземних відео

Для реалізації поставленої мети **сформульовані наступні завдання:**

- проаналізувати предметну область;
- привести проектні рішення з автоматизованої адаптації іншомовних відео;
- розробити програмне забезпечення по автоматизованому озвучуванню перекладеного тексту з відео;
- розробити стартап-проект.

Наукова новизна результатів магістерської дисертації полягає в тому, що запропоновано архітектурне рішення для побудови програмного

забезпечення для створення автоматизованого перекладу відео. До цього цілісних рішень із комерційним використанням у цій області немає. Також додано урахування емоційної складової мовлення.

Практичне значення отриманих результатів полягає в тому, що реалізовані методи поєднані в межах одного застосунку і максимально прості у використанні для користувача. Також реалізовано API-інтерфейс, за допомогою якого результати роботи алгоритмів можуть з легкістю отримувати і застосовувати сторонні сервіси. Дана система може бути використана різноманітними сервісами відеохостингу, навчальними платформами та сервісами для кіно і серіалів.

Зв'язок з науковими програмами, планами, темами. Робота виконувалась на кафедрі інформатики та програмної інженерії Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського".

Апробація. Наукові положення дисертації пройшли апробацію на XV Всеукраїнської науково практичної WEB конференції аспірантів, студентів та молодих вчених (22-24 березня 2022 р.). у Кривому Рогу.

Публікації. Наукові положення дисертації опубліковані в:

- 1) «Комп'ютерні інтелектуальні системи та мережі. Матеріали XV Всеукраїнської науково практичної WEB конференції аспірантів, студентів та молодих вчених (22-24 березня 2022 р.). – Кривий Ріг: Криворізький національний університет, 2022. – ст.106-109»

Ключові слова: РОЗПІЗНАВАННЯ МОВЛЕННЯ, ІНОЗЕМНІ МОВИ, ПОРІВНЯННЯ АУДІО, TEXT-TO-SPEECH, SPEECH-TO-TEXT, DEEP LEARNING

ABSTRACT

Explanatory note size – 90 pages, contains 34 illustrations, 21 tables, 2 applications.

Topicality. The paper deals with a problem in the field of natural language learning, namely the problem of foreign languages. It shows the main features of the problem, its advantages and disadvantages. It also shows that only automated subtitling is widely used in commercial purposes today. The drawbacks of this approach are primarily related to the respect of the listener and the amount of intelligibility. The need to develop a new approach to the interpretation of non-native videos by people who do not know the language, and in particular, a supplement for the automated translation of the original video, has been identified. The work also focuses on the emotive structure of speech as a very important aspect of the language. It allows more information to be transmitted.

The aim of the study. The main objective is to increase the take-up of foreign videos.

Object of research: foreign videos

Subject of research:

To achieve this goal, the **following tasks** were formulated:

- first task;
- second task;
- third task.

The scientific novelty of the results of the master's dissertation is the architectural solution for the development of software support for the creation of automated video translation is proposed. Until then, there are no significant commercial projects in this area. Also, an understanding of the emotional structure of the message is added.

The practical value of the obtained results are the implemented methods that are combined within a single tool and are as simple as possible for the user to use. The ARI interface is also implemented, which allows the results of the algorithms to be easily retrieved and used by third-party servers. The system can be used by a variety of video hosting servers, educational platforms and movie and video servers.

Relationship with working with scientific programs, plans, topics. Work was performed at the Department of Informatics and Software Engineering of the National Technical University of Ukraine «Kyiv Polytechnic Institute. Igor Sikorsky».

Approbation. Scientific propositions of the dissertation were approved at the XV All-Ukrainian scientific and practical WEB conference of students, postgraduates and young students (22-24 June 2022). in Krivoy Rog.

Publications. The scientific provisions of the dissertation published in:

1)"Computer Intellectual Systems and Instruments. Proceedings of the XV All-Ukrainian Scientific and Practical WEB Conference of Graduate Students, Students and Young Scientists (22-24 June 2022) - Krivoy Rog: Krivoy Rog National University, 2022 - pp.106-109"

Keywords: SPEECH RECOGNITION, FOREIGN LANGUAGES, AUDIO COMPARISON, TEXT-TO-SPEECH, SPEECH-TO-TEXT, DEEP LEARNING

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	12
ВСТУП	13
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	14
1.1 Історичний опис технологій TTS.....	14
1.1.1 TTS на основі формант[2, 3, 4]	15
1.1.2 Артикуляційні TTS [5,6].....	15
1.1.3 Конкатенативні TTS [7]	15
1.1.4 Параметричні TTS[8]	16
1.1.5 TTS на основі нейронний мереж[9].....	16
1.1.6 Аналіз тексту	17
1.1.7 Акустична модель	18
1.1.8 Вокодер	19
1.1.9 Оцінка синтезованого мовлення.....	19
1.2 Теоретичні основи цифрової обробки сигналів при вирішення задачі TTS.....	19
1.2.2 Перетворення Фур'є.....	21
1.2.3 Спектрограми	21
1.2.4 Фундаментальна частота (F0) [11]	21
1.2.5 Форманти	22
1.2.7 Мелчастотні кепстральні коефіцієнти (MFCCs).....	24
1.3 Опис технологій, використаних у роботі.....	27
1.3.1 Google Colab	27
1.3.2 Jupiter	29
1.3.4 Django	29
1.3.5 Praat.....	30
ВИСНОВКИ ДО РОЗДІЛУ 1	35

ПРОЕКТНІ РІШЕННЯ З АВТОМАТИЗОВАНОЇ АДАПТАЦІЇ

ІНШОМОВНИХ ВІДЕО.....	36
2.1 Amazon Polly.....	36
2.2 Google Cloud Speech.....	36
2.3 IBM Watson	36
2.4 Descript	36
2.5 TTS Рішення із відкритим кодом.....	37
2.5.1 Автоенкодер [18].....	39
2.5.2 Варіаційний Автоенкодер (Variational Autoencoders)	40
2.5.3 Як працює VITS.....	42
2.5.3.1 Постеріорний енкодер	42
2.5.3.2 Апріорний енкодер.....	42
2.5.3.3 Декодер.....	43
2.5.3.4 Дискримінатор.....	43
2.5.3.5 Стохастичний предиктор тривалості	43
2.5.3.6 Варіативний інференс	44
2.5.3.7 Оцінка вирівнювання отримана з варіативного інференсу	45
2.5.3.8 Змагальне тренування для поліпшення якості синтезування	46
2.5.3.9 Функція витрат	46
ВИСНОВКИ ДО РОЗДІЛУ 2	47
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПО	
АВТОМАТИЗОВАНОМУ ОЗВУЧУВАННЮ ПЕРЕКЛАДЕНОГО	
ТЕКСТУ З ВІДЕО	48
3.1 Датасети	48
3.1.1 Ukrainian Open Speech To Text Dataset 4.2.....	48
3.1.2 M-AILABS Ukrainian dataset	49
3.1.3 Mozilla Common Voice Ukrainian model.....	50
3.1.4 VoxForge Repository	50

3.2 Процес навчання TTS	51
3.2.1 Конфігурація.....	51
3.2.2 Препроцесінг	52
3.2.2.1 Приведення до однакової дискретизації.....	52
3.2.2.2 Приведення до однакової файлової структури	52
3.2.2.3 Приведення до однакового формату	52
3.2.2.4 Приведення до однакової структури метаданих.....	52
3.2.3 Тренування.....	53
3.3.1 Налаштування серверу Django.....	54
3.3.3 Екстрагування аудіо із відео	55
3.3.4 Зміна дискретизації	55
3.3.5 Виконання частини STT	56
3.3.6 Автоматичний переклад	56
3.3.7 Виконання частини TTS	56
3.3.8 Виконання адаптування емоційної частини.....	56
3.3.9 Перевірка якості зміни емоцій	60
3.3.10 Накладання зміненого аудіо на відео.....	62
3.3.11 Повернення зміненого відео клієнту.....	62
ВИСНОВКИ ДО РОЗДІЛУ 3	63
РОЗРОБКА СТАРТАП-ПРОЕКТУ	64
Принципова ідея проекту	64
4.2 Технологічний аудит ідеї проекту.....	67
4.3 Аналіз ринкових можливостей запуску стартап-проекту.....	68
4.4 Розроблення ринкової стратегії	80
4.5 Розроблення маркетингової програми стартап-проекту.....	84
ВИСНОВКИ ДО РОЗДІЛУ 4	90
ВИСНОВКИ.....	91
ПЕРЕЛІК ПОСИЛАНЬ	93

ДОДАТКИ.....	96
--------------	----

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

MOS (Mean Opinion Score) – метод середньої оцінки

Фонема – Найменша звукова одиниця мови

Графема – Найменша смислорозрізнявальна одиниця писемної мови

POS (Part of speech) – частина мови

MFCCs - Мелчастотні кепстральні коефіцієнти

ВСТУП

В останні часи глобалізація дуже сильно впливає на наше життя. Кожний, в кого є доступ в Інтернет, може спілкуватися з іншими людьми із усієї Землі; читати, слухати, дивитися, сприймати найрізноманітніший контент із усього глобального павутиння.

Тенденція останні роки розвивається таким чином, що більшість інформації та контенту розповсюджується англійською мовою. Також можна відмітити, що сама форма контенту останній час все більше і більше передається у форматі, який зручно сприймати, роблячи щось паралельно, а саме: подкасти, аудіо, фільми та ін. аудіо-візуальні різновиди контенту.

Треба зазнати, що, не зважаючи на те, що зараз доволі багато людей, особливо молодого віку, які можуть якось висловлюватись, спілкуватись, та сприймати інформацію англійською мовою, цього рівня не усім достатньо, щоб можна було абстрагуватися від сприйняття мови – іншомовного звучання, аналізу нових слів і т.п. та сприймати безпосередньо інформацію із відео/аудіо.

Тому можна зазначити, що актуальність цієї проблеми доволі велика. Постає питання розробки відповідного програмного забезпечення, яке б дозволяло людям, які не володіють англійською мовою на вільному рівні, сприймати контент у першу чергу із іномовних відео (у даному випадку буде розглянуто випадок із англійської в українську). Приймаються умовності, що оригінальне відео може мати субтитри.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Задачу, яку ставить перед собою дана робота, можна розділити грубо на декілька етапів, а саме:

1. STT – Визначення тексту, який треба перекласти/розпізнавання тексту із аудіо оригінального відео (крок може бути пропущено, якщо відео містить субтитри із оригінальним текстом).

2. TTS – Безпосередньо синтез мовлення українською мовою.

3. Збереження звуку оточення оригінального аудіо та накладання перекладаного аудіо на оригінальне відео.

Основний акцент у роботі надано саме етапу синтезу мовлення. Тому доцільно зробити історичний нарис, як вирішувалась ця задача і які методи використовуються сьогодні.

1.1 Історичний опис технологій TTS

Задача перетворення тексту в аудіо сьогодні розглядається із використанням наступних галузей:

- акустика
- лінгвістика
- цифрова обробка сигналів
- статистика
- глибоке навчання.

На рисунку. 1.1 зображено загальну схему перетворення тексту в аудіо. На вхід поступає текст, який треба синтезувати. Цей текст подається до системи TTS. Виконуються де-які перетворення і на вихід йде сигнал у формі хвилі. Цей сигнал вже можна конвертувати у аудіофайл.

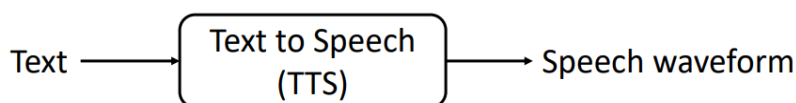


Рисунок 1.1 – Загальна схема задачі TTS

1.1.1 TTS на основі формант[2, 3, 4]

Такі системи працюють за рахунок того, що виробляються сегменти мовлення за рахунок генерування штучних сигналів, побудованих на специфічних правилах, які імітують структуру формантів та інші властивості мовлення.

Переваги:

- згенероване мовлено дуже зрозуміле
- підходить для слабких ПК та вбудованих систем

Недоліки:

- не натуральне звучання, подібне до роботів
- важко зпроектувати правила, які визначають параметри моделі

1.1.2 Артикуляційні TTS [5,6]

Такі системи працюють за рахунок імітування органів людини, які задіяні в процесі мовлення(губи, язик, голосову зв'язку та рухомий голосовий тракт).

Переваги:

- теоретично, найзрозуміліший і найнатуральніший метод

Недоліки:

- на практиці дуже важко побудувати модель артикуляційного апарату (важко збирати дані).

1.1.3 Конкатенативні TTS [7]

Дуже велика база даних містить аудіофрагменти коротких і високоякісних уривків мовлення, які записано однією людиною. Задля того, що відтворити відповідний текст, ці уривки перемішується у порядку тексту.

Переваги:

- зрозуміле мовлення

Недоліки:

- не підтримує емоції
- вимагає дуже велику базу даних для збереження уривків мовлення
- важко змінити голос.

1.1.4 Параметричні TTS[8]

Використовуються приховані Марківські моделі. Вся необхідна інформація для синтезування зберігається у параметрах моделі.

Переваги:

- гнучкі, не вимагають багато даних

Недоліки:

- менш зрозуміле мовлення ніж при конкатенативному підході.

1.1.5 TTS на основі нейронних мереж[9]

Окремий випадок параметричної моделі. Функціонують за рахунок глибоких нейронних мереж.

Переваги:

- дуже великий стрибок у якості мовлення як для зрозумілості, так і для натуральності
- треба проводити менше роботи для попередньої обробки даних.

Відповідно до порівняння вище наданих підходів до синтезу мовлення, у роботів використовується саме підхід на основі нейронних мереж. Загальна схема[10] цього підходу надана на рисунку. 1.2.

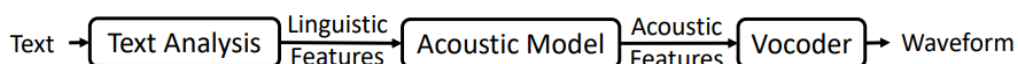


Рисунок 1.2 – Загальна схема нейронних TTS

Як видно із схеми, можна виділити 3 загальні блоки, які виконують задачу

1.1.6 Аналіз тексту

До цього блоку на вхід подається текст і на вихід – лінгвістичні фічі. У таблиці 1.1 наведено де-які функції даного блоку.

Таблиця 1.1 – Функції блоку аналізу тексту

Функція	Приклад
Нормалізація	125 → «сто двадцять п'ять»
Мінімізація багатозначностей	гóрод (місто), горóд (сад)
Сегментація фраз/слів/слогів	мураха → му-ра-ха
Визначення частин мови (POS)	Петр йде додому. → Noun, verb, adverb.
Індикація наголосів	Хіба (ві́н) просив (тебе́)?

До об'єктів лінгвістичних фічей відносяться:

- фонема
- слог
- слово
- фраза
- висловлювання

Відповідні деякі лінгвістичні фічі наведено у таблиці 1.2

Таблиця 1.2 – Приклади фічей об'єктів блоку аналізу тексту

Об'єкт фічей	Приклади фічей
Фонема	- поточна фонема - попередня і наступна фонери

Слог	- кількість фонем у слогі - схожість голосних у слогі
Слово	- передбачання частини мови попередніх та наступних слів - позиція слова у контексті фрази
Фраза	-кількість слогів у попередніх і наступних фразах
Висловлювання	-кількість слогів, слів та фраз у висловлюванні.

1.1.7 Акустична модель

Акустична модель виконує функцію перетворення лінгвістичних фічей в акустичні, які потім перетворюються в форму сигналу за допомогою вокодерів. Вибір акустичних характеристик значною мірою залежить від типу TTS.

Приклади фічей акустичної моделі:

1. Фундаментальна частота. (F0)
2. Голосні/приголосні

Мовлення у заданому відрізку часу вважається голосним, якщо середня потужність переходить деяку планку, яку встановив користувач. В іншому випадку мовлення вважається приголосним.

Деякі голосні звуки – [a], [o], [y]

Деякі приголосні звуки – [б], [п], [м]

3. Енергія мовлення
4. Мелчастотні кепстральні коефіцієнти(MFCCs).
5. Віконна функція/віконне перетворення Фур'є.

1.1.8 Вокодер

Вокодер перетворює спектрограми в аудіо сигнал.

1.1.9 Оцінка синтезованого мовлення

Якість синтезованого мовлення оцінюється за наступними критеріями:

- розбірливість
- натуральність
- емоційна складова
- інші критерії, які важливі для конкретної задачі.

Хоча можна робити перевірку на кожен критерій синтезованого мовлення, однак зазвичай користуються так званим MOS [1] (Mean Opinion Score).

MOS використовується зазвичай тоді, коли об'єкт, який треба оцінити, має дуже багато об'єктивних параметрів і оцінювати кожен із них стає надто важко. Тому оцінюється безпосередньо суб'єктивне відчуття мовлення. Це спрощує процедуру оцінки. Виражається оцінка числовим значенням від 1 до 5, 1 – низька якість та 5 – найкраща якість. MOS є досить суб'єктивною оцінкою, оскільки він заснований на сприйнятті якості голосу людьми. Однак є програми, які вміють вимірювати MOS і такі дані об'єктивніші.

1.2 Теоретичні основи цифрової обробки сигналів при вирішення задачі TTS.

1.2.1 Загальні терміни

Енергія

Під енергією сигналу розуміється наступне рівняння:

$$Energy(x) = var(x) = E[(x - \mu)^2], \quad (1.1)$$

де $\mu = E[x]$ – середнє значення сигналу x .

Оскільки амплітуда варіюється, немає сенсу рахувати миттєву енергію, а тільки у рамках вікна (заданого інтервалу). Рахується енергія у децибелах у логарифмічній шкалі, що робить обробку сигналу більш зручною (рис. 1.3, рис. 1.4).

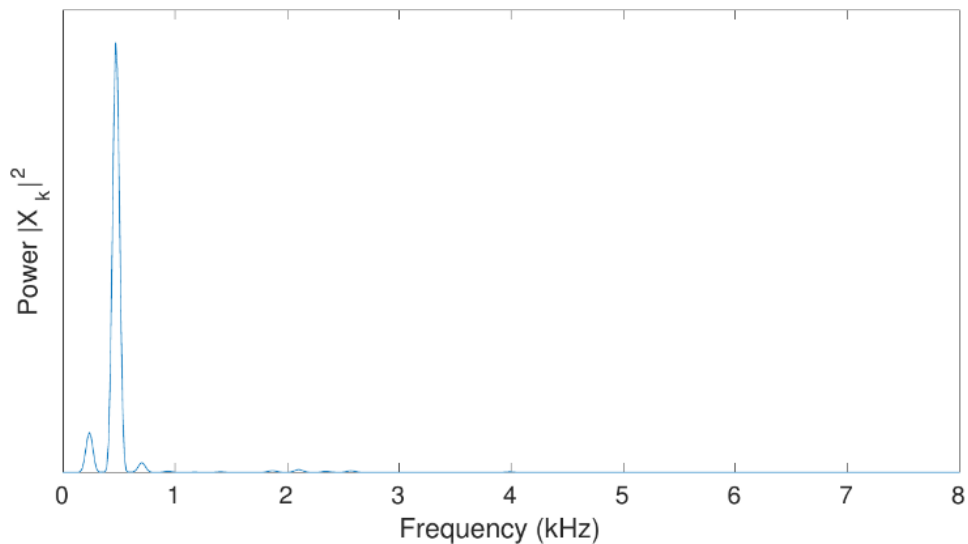


Рисунок 1.3 – Порівняння лінійної та логарифмічної шкали

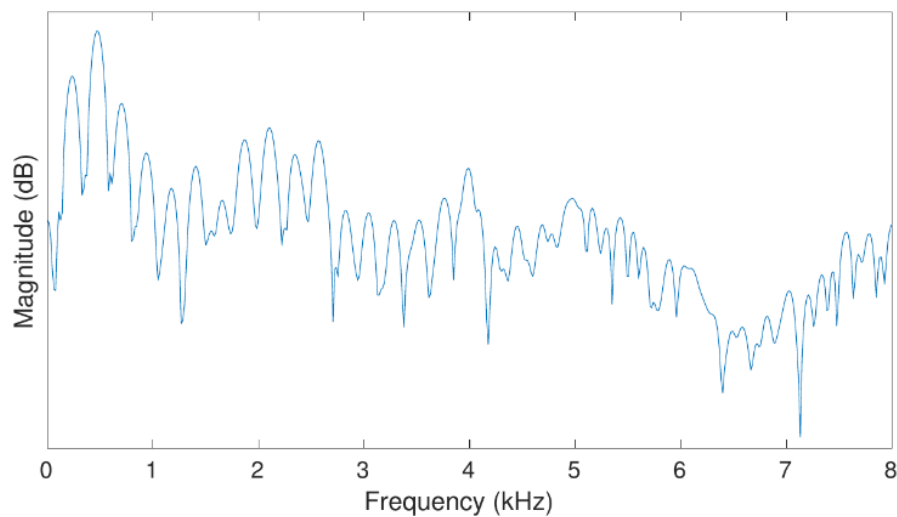


Рисунок 1.4 – Порівняння лінійної та логарифмічної шкали

1.2.2 Перетворення Фур'є

Дана операція є найважливішою і найпопулярнішою в обробці сигналів. Вона переносить сигнал із області часу в область частоти. Це надає багато важливої інформації щодо характеристики сигналу.

1.2.3 Спектрограми

Спектрограма – це тримірне відображення сигналу у області частот-амплітуд-часу. Таке перетворення особливо цінне для обробки мовленевих сигналів.

1.2.4 Фундаментальна частота (F0) [11]

F0 відповідає особливій характеристиці, яка показує індивідуальну квазі-періодичну наближену частоту мовлення. Це не чітко виражений періодичний сигнал, бо містить флуктуації. Однак все ж це є досить загальною характеристикою мовлення. У жінок та дітей цей показник вище ніж у чоловіків. Зазвичай рангується у діапазоні від 80 до 450 Гц.

На рисунку 1.5 показано, як із фундаментальною частотою повторюються високі області частоти. У даному випадку F0 становить 1/7 від 1000 Гц.

Може варіюватися від речення до речення. Культурні та стилістичні аспекти також впливають на цей показник. Попри того, варіюючи цю величину, можна задавати інтонацію висловлювання, таку як запитальну чи окликову.

Цей показник дуже схожий із характеристикою висоти звуку (pitch). Але F0 виражає об'єктивну міру, а висота – людське сприйняття.

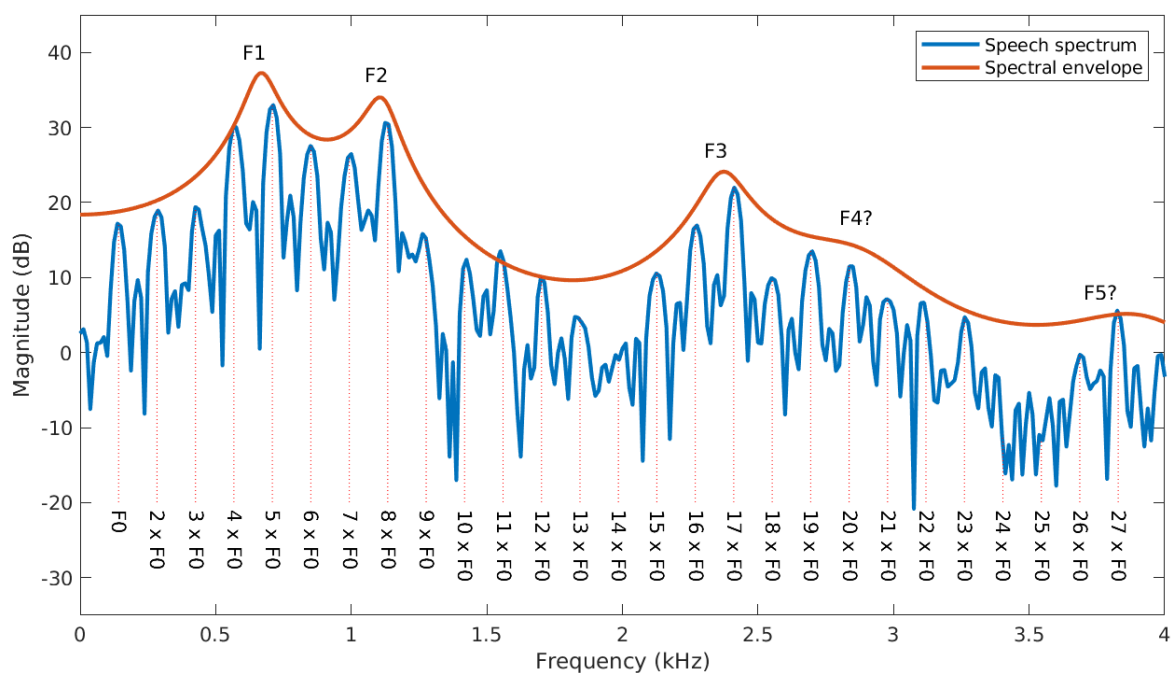


Рисунок 1.5 – Фундаментальна частота. (F1, F2.. – форманти)

1.2.5 Форманти

Форманти надають інформації про частоти, які більш за все превалюють у спектрограмі. На рисунку 1.5 зображено F1, F2 частоти деяких звуків.

Також за допомогою співвідношення формант одна до одної особливо зручно отримувати інформацію щодо конкретно висловлювальних звуків(рис. 1.6)

1.2.6 Мел-частота

Мел – психоакустична характеристика відчуття частоти. Базується на припущенні, що людині легше відчувати різницю у низьких частотах та важче у високих. На рисунку 1.8 показано співвідношення мел до частоти.

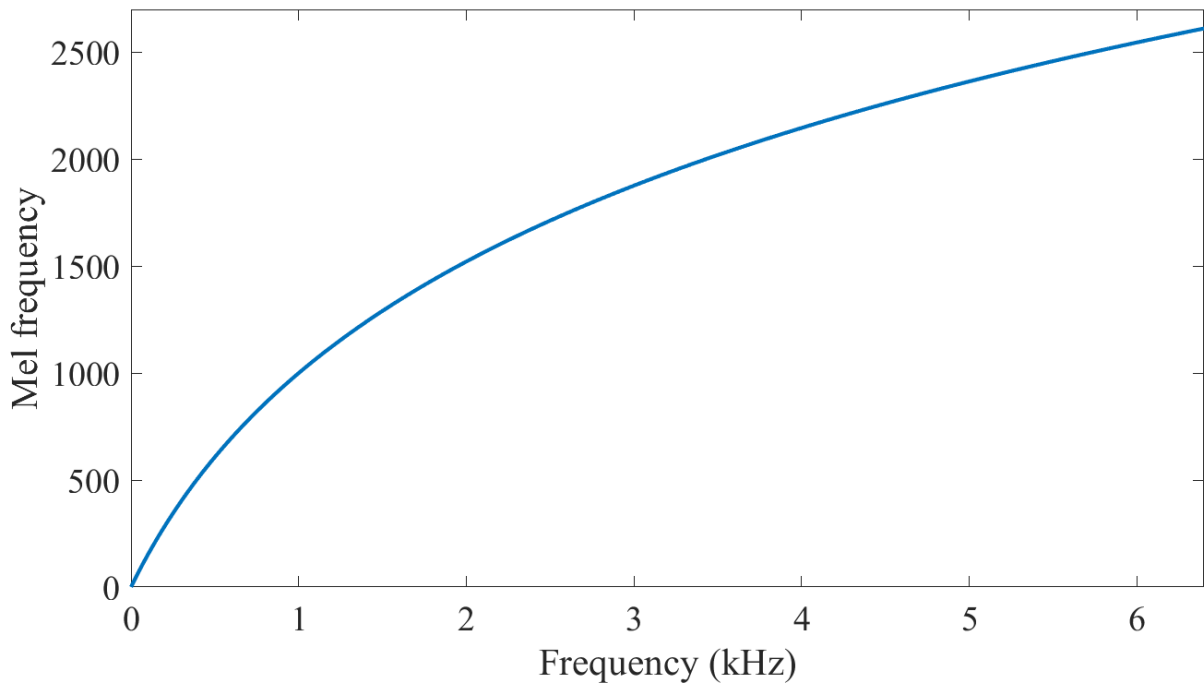


Рисунок 1.8 – Співвідношення мел до частоти

Загальна формула розрахунку мел:

$$m = 2595 \log_{10} \left(1 + \frac{f}{700} \right), \quad (1.2)$$

де f – частота.

1.2.7 Мелчастотні кепстральні коефіцієнти (MFCCs)

Треба зазначити, що представлення сигналу в області амплітуди сигналу не дає жодної інформації щодо макро-структур саме мовлення. Такі фічі, як голосні звуки, дуже важливі для аналізу, бо вони надають інформацію щодо внутрішньої структури мовлення. Тому доцільно

використовувати таке відображення на такій області, що можна вилучити найбільше інформації щодо саме мовлення, як окремого випадку сигналу.

Кепструм – операція, схожа на трансформацію Фур'є, але виконується за рахунок функції косинуса [12]. Такий підхід має деякі переваги:

- отримаємо більш деталізовану інформацію щодо макро-структур сигналу (кожна форманта відповідає деякому звуку);
- нижні частоти містять інформацію про фічі спектра, які змінюються повільно (це допомагаю точно визначати форманти);
- кепструм має гармонічну структуру;
- визначення фундаментальної частоти проходить більш робасно та легко.

Для отримання MFCCs на кепструм накладається так званий триангулярний фільтр(рис. 1.9). За його допомогою сигнал деякої частоти буде підсилюватися, а іншої – послаблюватися.

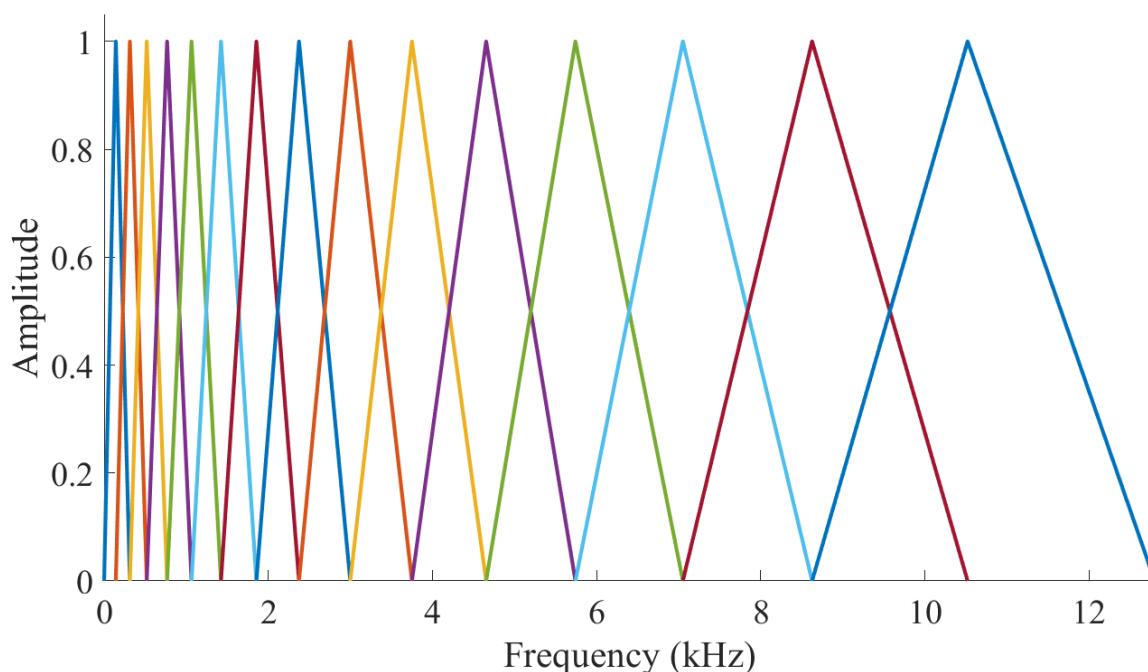


Рисунок 1.9 – Приклад триангулярного фільтру

Накладання такого фільтру на кепструм породжує наступну структуру (рис. 1.10).

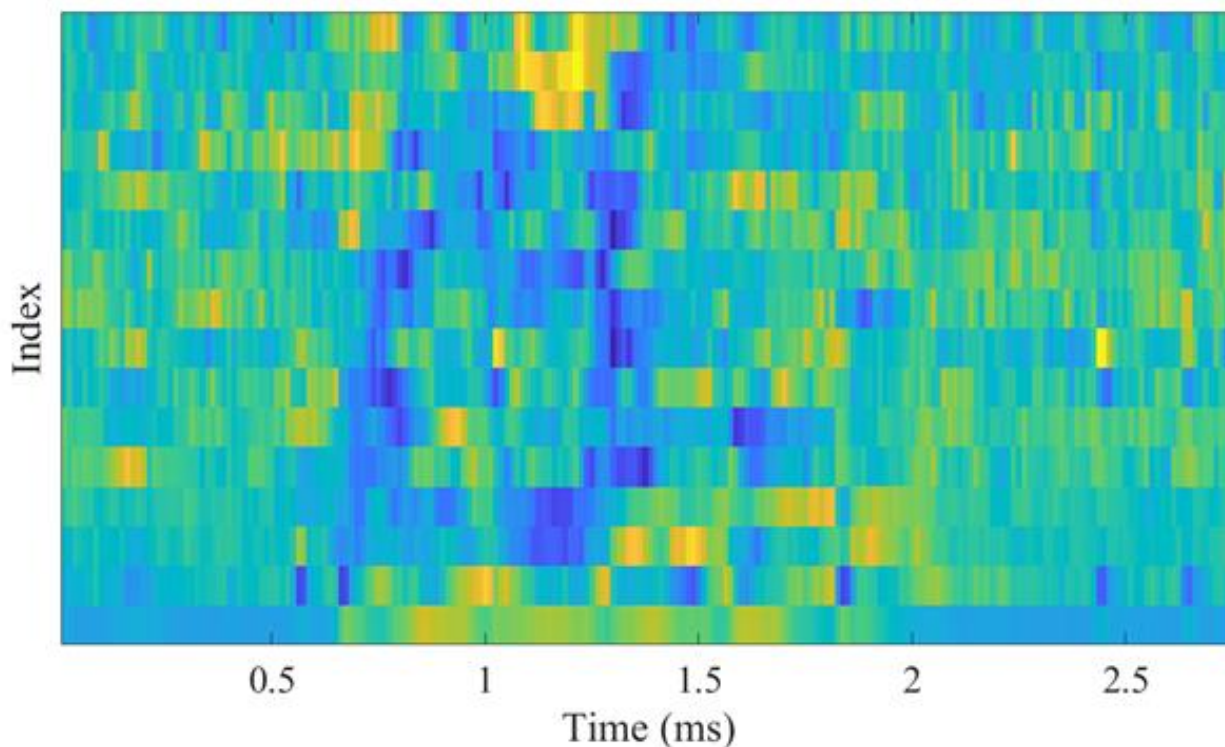


Рисунок 1.10 – Приклад зображення MFCCs

Мел спектрограма містить оригінальну форму спектру, в той час, як MFCCs не надає такої очевидної інтерпретації. Область, у якій рахуються значення MFCCs, це абстрактна область, яка містить інформації про спектральну огинаючу мовленнєвого сигналу.

Переваги MFCCs:

- визначає загальну форму спектру (спектральну огинаючу), що важливо, наприклад, для ідентифікації голосних. У той же час він видаляє тонку спектральну структуру (структуру на мікрорівні), яка часто менш важлива. Таким чином, він фокусується на тій частині сигналу, яка зазвичай є найбільш інформативною

- простий і досить ефективний з точки зору обчислень

- продуктивність добре перевірена та зрозуміла.

Недоліки MFCCs:

- не об'єктивна шкала
- не робасні до шуму
- триангулярний фільтр працює не ідеально, але аналоги мало де випробовувалися і не мають популярності.

1.3 Опис технологій, використаних у роботі

1.3.1 Google Colab

Google Colab – у першу чергу потужний та зручний інструмент для роботи із даними. Представляє собою виділену машину із операційною системою Linux. Дозволяє ефективно працювати із даними, обчисленнями, науковими дослідженнями. Це дуже корисний додаток, бо не всі ПК обладнені високопродуктивними процесорами, відеокартами і оперативною пам'яттю. Деякі задачі машинного навчання просто неможливо виконати, бо де-які фреймворки, наприклад Tensorflow, використовують ядра CUDA, які мають тільки потужні відеокарти. А якщо виконувати такі обчислення на процесорі, то це може зайняти кілька тижнів, якщо не більше.

Саме тому Колаб настільки чудовий. Він надає можливість абстрагуватися від «заліза» і виконувати інженерну задачу, не переймаючись за ресурси.

Треба зазначити, що додаток є браузерним, а отже, кросплатформеним. Це означає, що виконувати роботу у ньому можна не тільки із ПК, але й із мобільних пристроїв. Це зручно, якщо треба швидко зробити невеликі зміни у коді програм, і нема доступу до ПК.

Google надає три можливості користуватися сервісом: Free, Pro, Pro+.

Порівняльна таблиця можливостей і властивостей відповідних варіантів зазначено у таблиці 1.3.

Таблиця 1.3 – Порівняння тарифів на користування Google Colab

	Colab Free	Colab Pro	Colab Pro+
Гарантовані ресурси	незначні	високі	найвищі
GPU (Відеокарта)	K80	K80, T4, P100	K80, T4, P100
RAM	16 GB	32 GB	52 GB
Процеси у бекграунді	ні	ні	так
Макс. Тривалість сесії	12 годин	24 годин	24 годин
Вартість	бескоштовно	9,25 EUR	42,25 EUR

Як видно із порівняльної таблиці, за більші ресурси треба вносити кошти. Найважливішим критерієм є відеоадро.

Таблиця 1.4 – Порівняння відеокарт у Гугл Колаб [14]

GPU	Consumed ML Units	Runtime (hours)	Half precision	Automatic Mixed precision	Training Throughput (tokens/s)	Accuracy - BLEU Score
K80	465	244			1230	15.67
P100	122	32	✓		8800	21.37
T4	34	30	✓	✓	9297	21.37

Судячи з порівняння, видно, що K80 програє іншим відокартам у числах із половинною точністю, не має автоматичної змішаної точності і загальна точність нижче, ніж у P100 та T4.

Також безкоштовна версія Колаб має тільки 16 ГБ оперативної пам'яті, що іноді недостатньо, коли виконуються важкі обчислення.

Спочатку тренування моделей проводилися на безкоштовній версії, але обмеженість ресурсів робила обчислення дуже повільними. Тому було прийнято рішення використовувати Про версію Колаб.

1.3.2 Jupiter

Юпітер – зручний інструмент для написання Python програм. Працює схожим чином, як Matlab. Мається на увазі, що результат команд, які виконуються, не втрачаються і його можна використовувати для подальшого програмування.

Також Юпітер надає зручне форматування програм, яке робить читання і виконання такого коду інтуїтивно зрозумілішим. Варто зазначити, що у Колаб за замовчуванням використовується саме Юпітер.

1.3.4 Django

Django — це високорівневий веб-фреймворк Python, який робить можливим швидко розгортувати безпечні та підтримувані веб-додатки. Django дозволяє зосередитися на написанні додатки. Він безкоштовний і відкритий, має активну спільноту, гарну документацію та багато варіантів безкоштовної та платної підтримки.

Django забезпечує майже усіма інструментами, що можуть знадобитися, по замовчуванню. Тобто не треба сильно квапитися щоб встановлювати багато інших інструментів.

Фреймворк дуже багатогранний і може використовуватися у багатьох кейсах: від систем керування вмістом і побудови енциклопедичних статей, до соціальних мереж і сайтів із новинами. Він сумісний з багатьма фреймворками, які працюють на стороні клієнта і може відправляти повідомлення у багатьох популярних форматах, таких як HTML, RSS-канали, JSON, XML тощо.

Django надає можливість достатньо безпечним чином управляти обліковими записами та паролями користувачів, уникаючи розповсюджених помилок, таких як розміщення інформації про сеанс у файлах cookie, де їх зберігати небезпечно (натомість cookie містять лише ключ, а дані, які закріплені за цим куки зберігатимуться в базі даних).

Django за замовчуванням захищає від розповсюджених уразливостей, таких як SQL injection, cross-site scripting, підробку міжсайтових запитів і клікджекінг.

Django використовує архітектуру при якій всі частини архітектури незалежні одна від одної, а тому вони може бути замінені, якщо того потребує рішення. Наявність чіткого поділу між різними частинами означає, що він може масштабуватися для збільшення трафіку, додаючи обладнання на будь-якому рівні: сервери кешування, сервери баз даних або сервери програм. Деякі з найбільш завантажених сайтів успішно масштабували Django, щоб задовільнити свої вимоги (Instagram і Disqus і т.д.).

Із патернів та принципів програмування Django реалізує «Не повторюйся» (DRY), щоб не було непотрібного дублювання, та «Model View Controller» (MVC).

1.3.5 Praat

Praat – безкоштовна бібліотека для фонетичного аналізу та обробки аудіофайлів. Акцент розробники робили саме на обробці голосових

аудіофайлів. Про це свідчить широкий функціонал програми, пов'язаний саме з голосовими аудіофайлами.

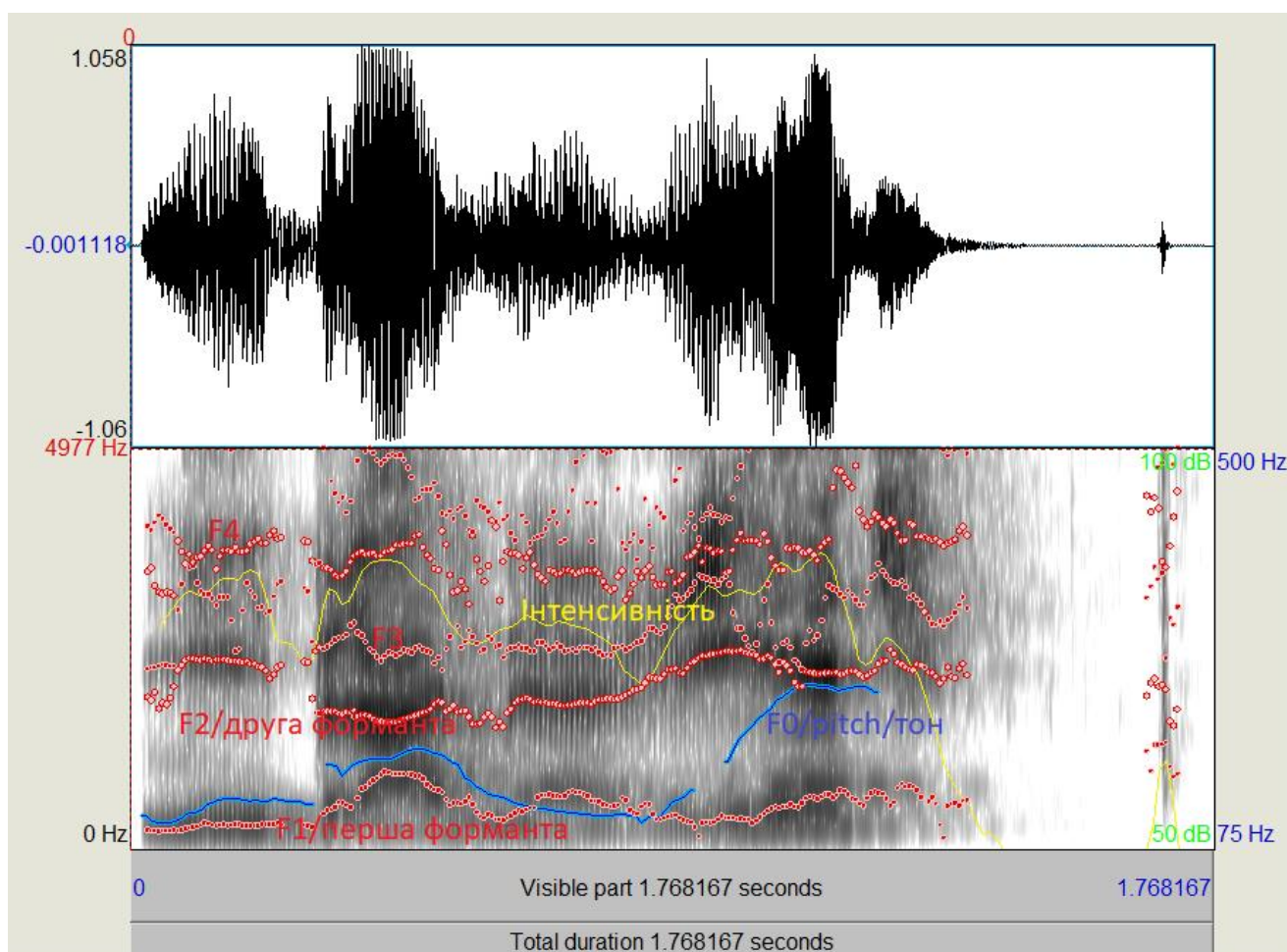


Рисунок 1.11 – Приклад вікна обробки аудіофрагментів мовлення

На рисунку 1.11 показано амплітудну характеристику і спектральну характеристику мовленевого сигналу. На спектрограмі нанесено фундаментальну частоту (F0), наявні форманти (F1, F2, F3..), рівень інтенсивності.

Як видно із рисунку 1.11, більшість найважливіших властивостей мовленевого сигналу можна спостерігати на одному зображенні, що робить роботу з маніпулювання мовленими сигналами дуже зручною.

Бібліотека має широкий функціонал [15]:

Таблиця 1.5 – Можливості бібліотеки Praat

Функціональніс ть	Приклади			
	Аналіз мовлення	Спектральни й аналіз	Аналіз тону	Аналіз формант інтенсивнос ті
Розмітка і сегментація	Розмітка інтервалів і часових точок	Використання фонетичного алфавіту	Використання аудіофайлів розміром до 2 ГБ	
Алгоритми начвання	Нейронні мережі прямого розповсюджен ня	Дискретна та стохастична теорія оптимальності		
Робота із графікою	Створення інкапсульован их файлів PostScript	Інтегровані математичні та фонетичні символи		
Синтез мовлення	Синтез із використання м тону, формант та інтенсивності	Артикуляційн ий синтез	Акустичний синтез Клатта	

Маніпуляція із мовленням	Зміна тону, тривалості	Фільтрація		
Статистичні операції	Масштабування у великій кількості розмірностей	Принциповий аналіз компонентів	Дискримінаційний аналіз	
Можливості програмування	Легка скриптова мова програмування	Взаємодія із іншими програмами	Відправка повідомлень із командного рядка	Гіпертекстові маніпуляції із звуком

Як видно із таблиці, бібліотека має у своєму розпорядженні багато функціональних можливостей. У першу чергу, треба зазначити що бібліотека має можливість виконання заскриптовані команди. Це саме те, що уможливує автоматизовану обробки аудіо.

Наприклад, можна екстрагувати деякі характеристики сигналу у так звані “Tier”-и.

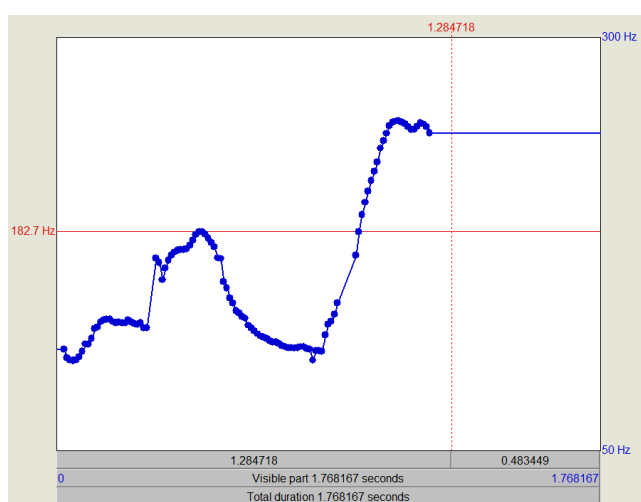


Рисунок 1.12 – «PitchTier» - Екстрагована інформація щодо тональності на голосовому сигналі.

На рисунку 1.12 зображено PitchTier попереднього мовленевого сигналу. Значення зберігаються у такому випадку наступним чином:

«момент часу» - «відповідна частота»

Відповідні маніпуляції можна проводити також із тривалістю, інтенсивністю, озвучуванням (voicing) і тд (рисунок 1.13).

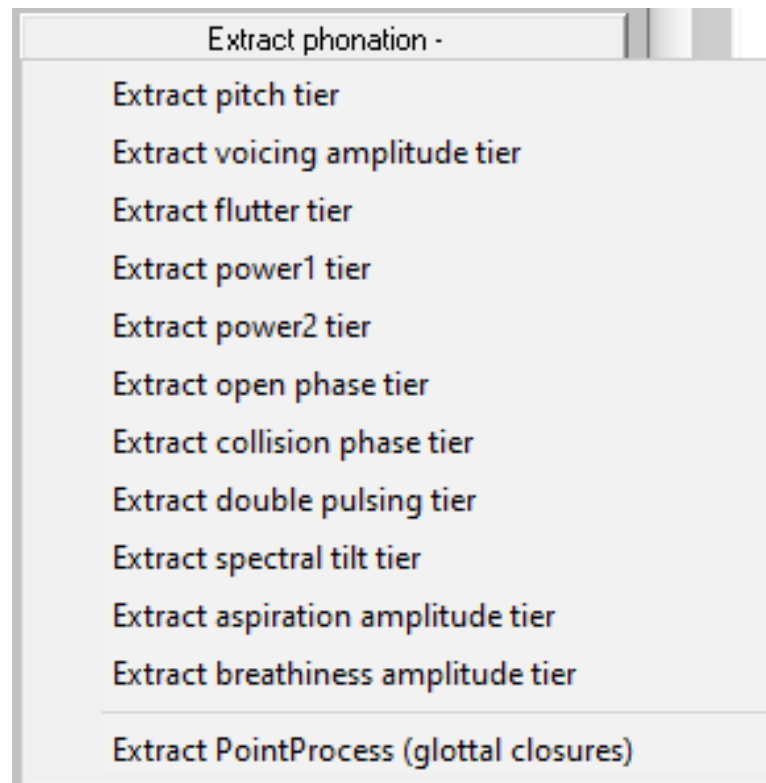


Рисунок 1.13 – Можливі варіанта екстрагованих властивостей мовленевого сигналу

Бібліотека постачається у кількох варіантах: із графічним інтерфейсом та без нього. Виконання скриптів можливе в обох випадках.

ВИСНОВКИ ДО РОЗДІЛУ 1

Було підкреслено важливість розглядання саме проблеми TTS як ключової проблеми даної роботи. Зроблено історичний огляд технологій TTS, їх порівняння. Зазначено технології, якими користуються сьогодні, а саме – на основі параметричних нейронних мереж.

Розглянуто побудову сьогоденішніх TTS. Одночасно з цим зроблено мінімальний фундаментальний огляд теоретичних основ цифрової обробки сигналів для розв’язання проблем обробки мовлення.

2 ПРОЕКТНІ РІШЕННЯ З АВТОМАТИЗОВАНОЇ АДАПТАЦІЇ ІНШОМОВНИХ ВІДЕО

Насамперед повноцінних рішень із перекладу англomовних відео українською не було знайдено.

Найголовнішу частину системи складатиме модуль перетворення україномовного тексту в аудіо. Тому в якості проектних рішень буде порівняно існуючі рішення, які реалізують саме цю функціональність [13].

2.1 Amazon Polly

Найлегший у використанні сервіс із синтезу мовлення. Добре інтегрується у додатки. Зрозумілий API. Підтримує обробку у реальному часі. Непогана робота з вимовою та інтонаціями. Підтримка мови SSML. Наявна оптимізація аудіо.

2.2 Google Cloud Speech

Цей сервіс відрізняється тим, що його найлегше встановити і почати роботу. Адміністрування також на високому рівні. Підтримка сервісу також на висоті. Підтримка мови SSML. Добротна натуральність вимови.

2.3 IBM Watson

Надає тисячу знаків для обробки безкоштовно. Далі - \$0.02 USD за 1000 символів. Підтримка мови SSML. Використовує зазвичай середній та великий бізнес.

2.4 Descript

Добротне програмне забезпечення для роботи із синтезуванням мовлення. Має широку функціональність. Поставляється із приємним інтерфейсом користувача. Найбільш простий у користуванні. Велике

розповсюдження сервісу також наявне. Використовує зазвичай малий бізнес.

Цінова політика:

Creator - \$12/ місяц

Pro - \$24 / місяц

2.5 TTS Рішення із відкритим кодом

Існує багато рішень від різноманітних команд, які також виконують синтезування тексту в аудіо.

Із 2016 року активно розвиваються наступні архітектурні рішення[16]:

- Акустична модель
- Вокодер
- Акустична модель + вокодер
- Скрізна модель(end-to-end)

З них найбільший інтерес представляють скрізні моделі, бо в них навчання моделі виконується уцілому для всього загального процесу синтезування мовлення. Тобто, не потрібно спочатку навчати модель, яка буде прогнозувати акустичні ознаки із тексту, чи модель, яка прогнозуватиме аудіосигнал із акустичних ознак. Навчання виконується безпосередньо із тексту на аудіосигнал. Це робить навчання більш гнучким.

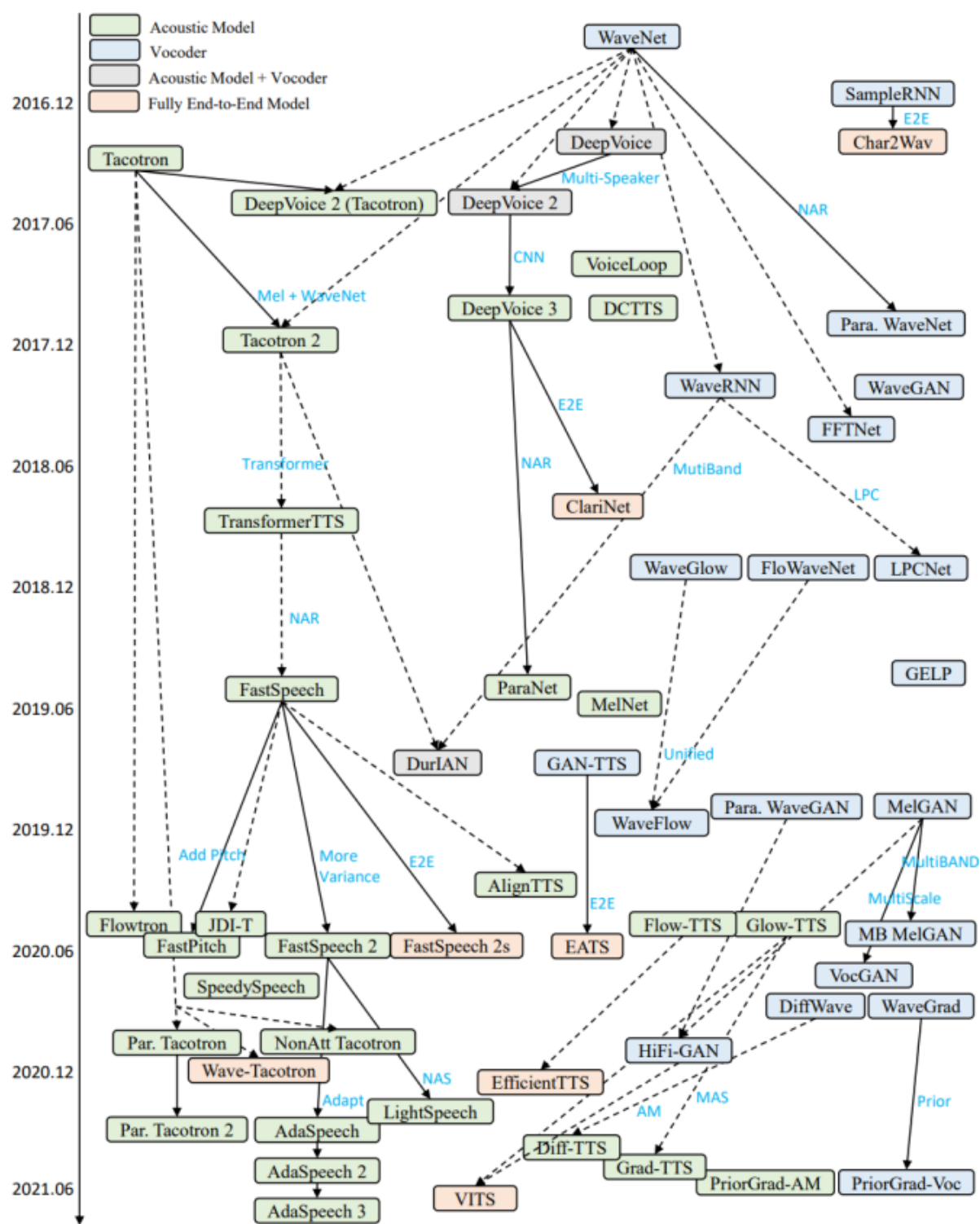


Рисунок 2.1 – Актуальні рішення TTS

З них варто розглянути VITS[17], яка є скрізною.

VITS (Variational Inference with adversarial learning for end-to-end Text-to-Speech) представляє собою умовний варіаційний автоенкодер (Conditional Variational Eutoencoder)

2.5.1 Автоенкодер [18]

Автоенкодер – це модель неймережі, яка намагається найкращим чином зтиснути вхідні дані. При цьому вона намагається реконструювати первинні дані із найменшою похибкою. Наявні наступні елементи:

Енкодер – неймережа, що закодує дані, намагається зтиснути їх до найменшої розмірності, щоб потім передати до прихованого простору.

Прихований простір/Bottleneck/Latent Space – найменший за кількістю нейронів слой, у який потрапляють закодовані дані із енкодеру.

Декодер – неймережа, що намагається відновити первинні дані із прихованого простору.

Вхідний сигнал відновлюється з помилками через втрати при кодуванні, але, щоб їх мінімізувати, мережа змушена вчитися відбирати найважливіші ознаки.

Функцію мінімізації помилки можна представити наступним чином:

$$L(x, f(g(x))), \quad (2.1)$$

де x – первинні дані,

$g(x)$ – закодовані дані в енкодері,

$f(g(x))$ – розкодовані дані в декодері.

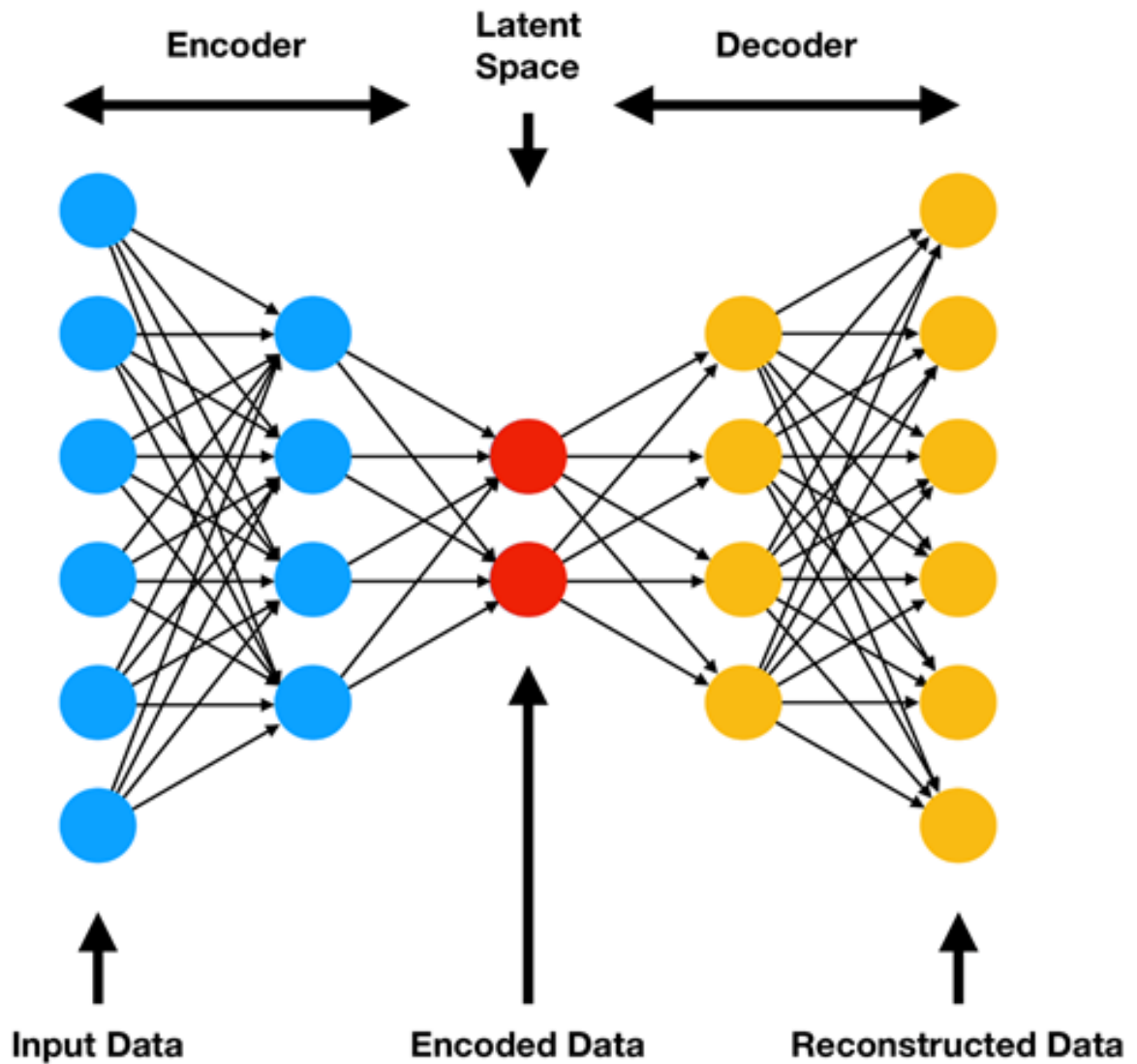


Рисунок 2.2 – Загальна архітектура автоенкодерів

2.5.2 Варіаційний Автоенкодер (Variational Autoencoders)

Варіаційний Автоенкодер – це автоенкодер, який навчається закодувати об’єкти в заданий прихований простір і відповідно, декодувати з нього.

Це означає, що найменший слой задається не саме даними, які потрапили на вхід, а вірогідністю закодованих даних потрапити на деякий простір.

Являється генеративною моделлю, у тому сенсі, що ми хочемо мати змогу генерувати правдоподібні фейкові сеймпли, які максимально схожі на наші первинні сеймпли з тренувального набору даних.

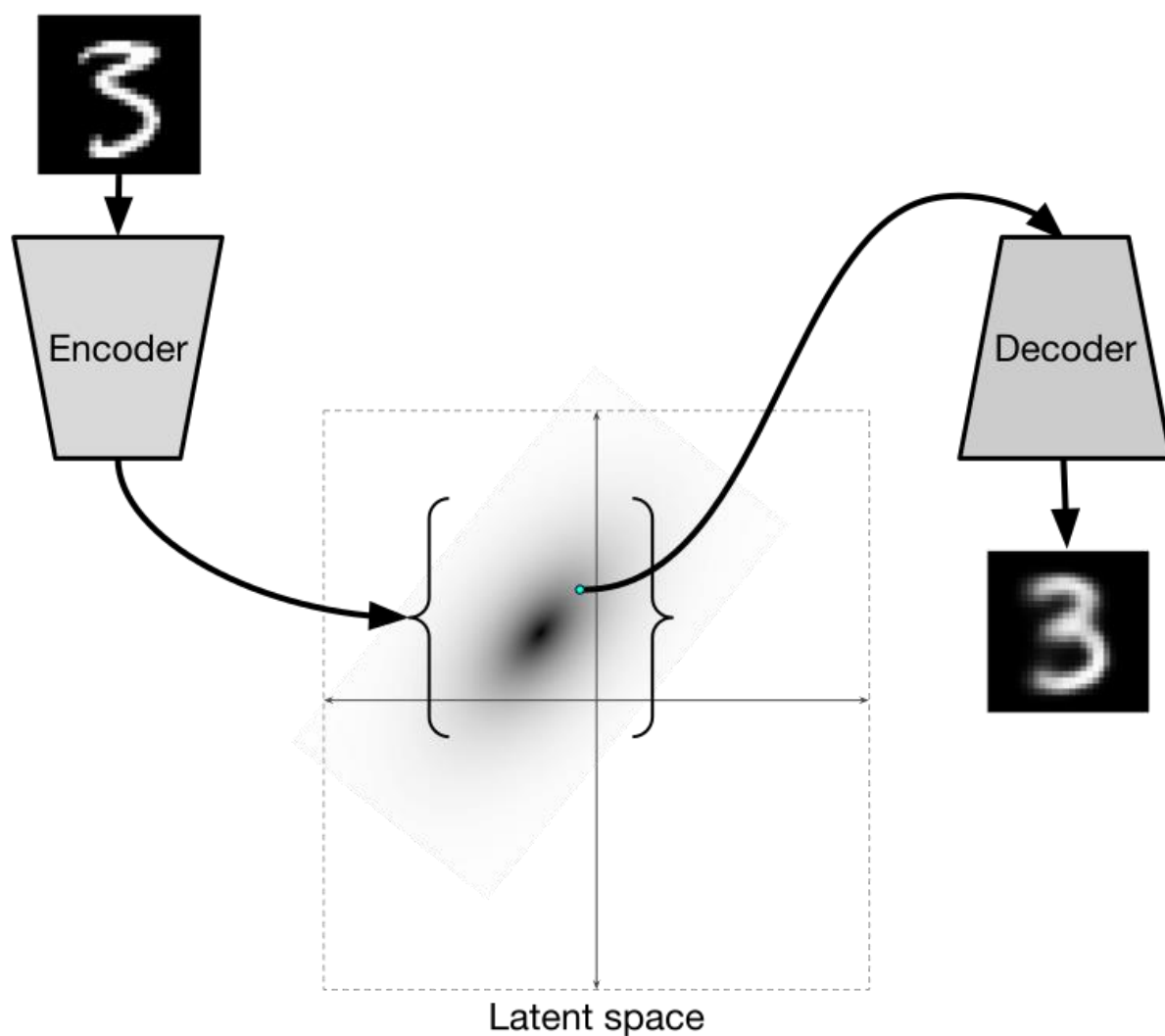


Рисунок 2.3 – Як працює варіаційний автоенкодер

Умовний варіаційний автокодер має додатковий вхід як для кодера, так і для декодера. Це дозволяє декодеру створювати нові сеймпли на вимогу.

Наприклад, щоб створити аудіо певного тексту, ми вводимо текст в декодер разом із випадковою точкою в прихованому просторі, відібраною зі

стандартного нормального розподілу. Навіть якщо для отримання двох різних аудіо подається одна й та сама точка, процес буде працювати правильно, оскільки система більше не покладається на прихований простір для кодування первинного аудіо. Замість цього прихований простір кодує іншу інформацію, яка стосується аудіо.

2.5.3 Як працює VITS

Загальна архітектура запропонованої моделі складається з постеріорного енкодера, апріорного енкодера, декодера, дискримінатора та стохастичного предиктора тривалості. Постеріорний енкодер і дискримінатор використовуються лише для навчання[17].

2.5.3.1 Постеріорний енкодер

Постеріорний енкодер складається з блоків WaveNet, які використовуються у Glow-TTS[20] та WaveGlow[21]. Блок WaveNet складеться з шарів розширеної згортки із закритим блоком активації та пропущеними зв'язками. Слой лінійної проекції вираховує середнє значення нормального постеріорного розподілу.

2.5.3.2 Апріорний енкодер

Апріорний енкодер складається з текстового енкодера, що обробляє вхідні фонемі c_{text} та потіку нормалізації f_{θ} , що покращує гнучкість апріорного розподілу. Використовується відносне позиціювання тексту, на відміну від абсолютного. Із c_{text} можна отримати прихований h_{text} , через енкодер та слой лінійної проекції, що виробляє середнє та варіативне значення, яке використовується для побудови апріорного розподілу.

Потік нормалізації – це стек із афінних слоїв, які складається зі стеку блоків WaveNet.

2.5.3.3 Декодер

Декодер представляє собою генератор HiFi-GAN V1 [22]. Він складається зі стеку транспонованої згортки. На виході отримується сума виходів залишкових блоків, що мають розрізнені розміри поля сприйняття.

2.5.3.4 Дискримінатор

Використовується дискримінатор, як у HiFi-GAN, який підтримує багато періодичностей. Це сума під-дискримінаторів віконної функції Морковіана[23], кожна з яких працює на різній частоті.

2.5.3.5 Стохастичний предиктор тривалості

Стохастичний предиктор тривалості оцінює розподіл тривалості фонем с умовного текстового інпуту h_{text} . Для ефектиної параметризації стохастичного предиктора тривалості, його блоки доповнюються розширеними та роздільними по глибині шарами згортки.

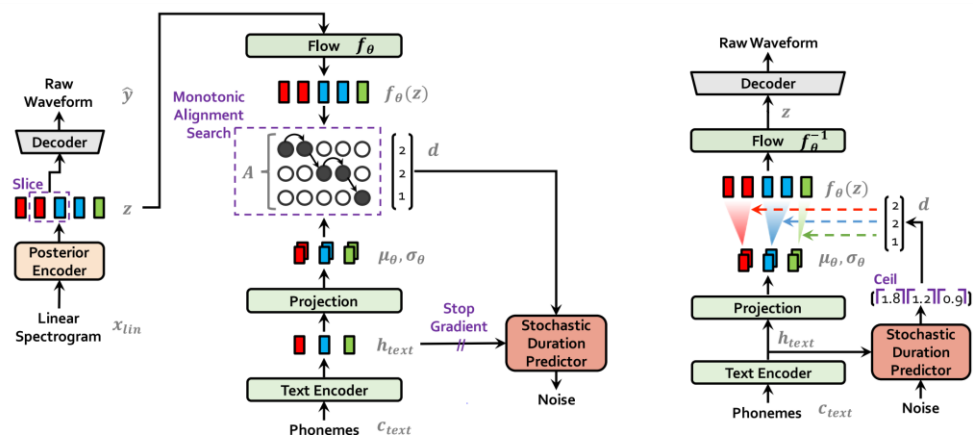


Рисунок 2.4 – Архітектура VITS на етапі навчання та на етапі виведення.

2.5.3.6 Варіативний інференс

VITS можна виразити як умовний варіаційний автоенкодер, у якому максимізується нижня границя (або нижня границя свідчення правдоподібності/ ELBO), логарифмічної правдоподібності даних $\log p_{\theta}(x|c)$:

$$\log p_{\theta}(x|c) \geq E_{q_{\phi}(z|x)} \left[\log p_{\theta}(x|z) - \log \frac{q_{\phi}(z|x)}{p_{\theta}(z|c)} \right], \quad (2.2)$$

де $p_{\theta}(z|c)$ показує апріорний розподіл прихованих змінних z заданої умови c ;

$p_{\theta}(x|z)$ - функція правдоподібності даних x ;

$q_{\phi}(z|x)$ – приблизний постеріорний розподіл.

Тобто, тренувальна похибка тоді буде негативною нижньою границею свідчення правдоподібності, яку можна розглядати як суму реконструкційних втрат і розходження Кульбака — Лейблера.

Цільовими даними для реконструкційних втрат будемо вважати мел – спектрограму - x_{mel} . Ми підвищуємо дискретизацію прихованих змінних z до області сигналу \hat{y} через декодер та перетворюємо \hat{y} в область $\widehat{x_{mel}}$.

Тоді реконструкційні витрати будуть втратами між передбаченими та цільовими мел спектрограмами. Вони виражаються так:

$$L_{recon} = \|x_{mel} - \widehat{x_{mel}}\| \quad (2.3)$$

Реконструкційні витрати визначаються в області мел, щоб поліпшити перцептивну якість, через те, що вона краще відповідає слуховим властивостям людини.

Входами апріорного енкодера c є фонемі c_{text} екстраговані із тексту та вирівнювання A між фонемами та прихованими змінними. Вирівнювання – матриця монотонної уваги із розмірністю $|c_{text}| \times |z|$, яка показує як довго

кожна вхідна фонема розтягується щоб бути вирівнюваною у часі із цільовим аудіо.

Основною ціллю являється впровадження більшої інформації із високою роздільною здатністю для постеріорного енкодера. Для цього використовуються спектрограма первинного аудіо у лінійному масштабі x_{lin} у порівнянні із мел-спектрограмою. Водночас, модифікований вхід не порушує властивостей варіаційного інференсу.

Тоді розходження Кульбака — Лейблера рахується так:

$$L_{kl} = \log q_{\phi}(z|x_{lin}) - \log q_{\theta}(z|c_{text}, A), \quad (2.4)$$

де $z \sim q_{\phi}(z|x_{lin}) = N(z; \mu_{\phi}(x_{lin}), \sigma_{\phi}(x_{lin}))$.

Факторизований нормальний розподіл виконується щоб параметризувати апріорний та апостаріорний енкодери. Це покращує виразність мовлення. Також використовується потік нормалізації f_{θ} [19], який уможливує інвертоване перетворення розподілу сеймпла у більш комплексний розподіл, який виконує правило заміни змінної.

2.5.3.7 Оцінка вирівнювання, що отримана з варіативного інференсу

Щоб оцінити вирівнювання A між вхідним текстом та цільовим мовленням, виконується так званий Пошук Монотонного Вирівнювання[20](Monotonic Alignment Search/MAS). Це допомагає шукати таке вирівнювання, яке максимізує схожість даних, які параметризовані нормалізованим потоком f :

$$\begin{aligned} A &= \arg \max_{\hat{A}} \log p(x|c_{text}, \hat{A}) = \\ &= \arg \max_{\hat{A}} \log N(f(x); \mu(c_{text}, \hat{A}), \sigma(c_{text}, \hat{A})). \end{aligned} \quad (2.5)$$

Але оскільки очікується нижня границя свідоцтва правдоподібності, а не логарифмічна схожість, треба перевизначити Пошук Монотонного

Вирівнювання для знаходження вирівнювань, які максимізують нижню границю свідоцтва правдоподібності:

$$\begin{aligned}
 A &= \arg \max_{\hat{A}} \log p_{\theta}(x_{mel}|z) - \log \frac{q_{\phi}(z|x_{lin})}{p_{\theta}(z|c_{text}, \hat{A})} = \\
 &= \arg \max_{\hat{A}} \log p_{\theta}(z|c_{text}, \hat{A}) = \\
 &= \log N(f_{\theta}(z); \mu_{\theta}(c_{text}, \hat{A}), \sigma_{\theta}(c_{text}, \hat{A})) \quad . \quad (2.6)
 \end{aligned}$$

2.5.3.8 Змагальне тренування для поліпшення якості синтезування

Щоб адаптувати змагальне тренування, додається дискримінатор D , який розрізняє вихід, створений декодером G від первинного аудіо y . Використовуються два типи функції витрат: найменших квадратів для змагального тренування та додадкова функція, яка співставляє ознаки для тренування генератора:

$$L_{adv}(D) = E_{(y,z)} [(D(y) - 1) + (D(G(z)))]^2 \quad (2.7)$$

$$L_{adv}(G) = E_z [(D(G(z)) - 1)^2] \quad (2.8)$$

$$L_{fm}(G) = E_{(y,z)} \left[\sum_{l=1}^T \frac{1}{N_l} \|D^l(y) - D^l(G(z))\| \right], \quad (2.9)$$

де T - показує загальну кількість слоїв у дискримінаторі,

D^l - виводе співставлені ознаки l -ого слою дискримінатору з кількістю ознак N_l .

2.5.3.9 Функція витрат

Функція витрат має наступний вигляд:

$$L_{vae} = L_{recon} + L_{kl} + L_{dur} + L_{adv}(G) + L_{fm}(G) \quad . \quad (2.10)$$

Функція складається з тренування варіційного автоенкодера та генеративної змагальної мережі.

ВИСНОВКИ ДО РОЗДІЛУ 2

У розділі було розглянуто та проаналізовано існуючі рішення з синтезування мовлення та порівняно їх між собою. Зроблено аналітичний огляд та запропоновано рішення, яке задовільнятиме потребам проекту найкращим чином.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПО АВТОМАТИЗОВАНОМУ ОЗВУЧУВАННЮ ПЕРЕКЛАДЕНОГО ТЕКСТУ З ВІДЕО

3.1 Датасети

В якості даних для тренування було взято кілька датасетів, а саме:

- Ukrainian Open Speech To Text Dataset 4.2 part 2[24]
- M-AILABS Ukrainian dataset[25]
- Mozilla Common Voice has the Ukrainian model[26]
- VoxForge Repository[27]

3.1.1 Ukrainian Open Speech To Text Dataset 4.2

Цей датасет знайдено на сайті Kaggle, яка займається організацією змагань серед інтузіастів по DataScience, на обліковому записи одного з учасників.

Склад датасету наведено на рисунку 3.1.

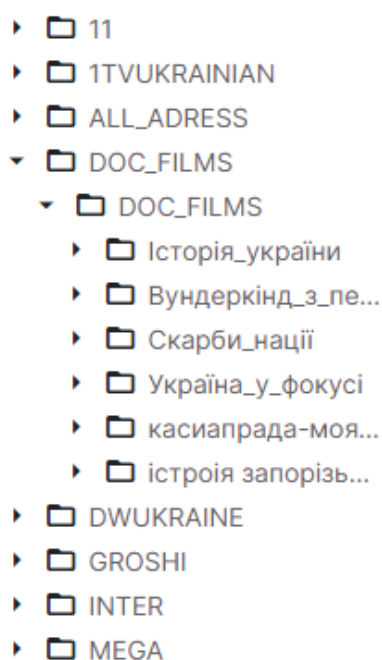


Рисунок 3.1 – Склад датасету Ukrainian Open Speech To Text Dataset 3.2

У наявності записи де-яких ТБ каналів та документальних кіно. Загальна тривалість аудіоконтенту – 27 годин. Загальна кількість входжень – 13470 аудіофрагментів. Частота дискретизації – 24050 Гц.

3.1.2 M-AILABS Ukrainian dataset

Цей датасет було зібрано компаніями Nash Format та Gwara Media. Аудіофрагменти розрізняються по тривалості від 1 до 20 секунд, а їх загальна тривалість – 87 годин 8 хвилин. Розмір – 9.3 Гб. Наявні як чоловічі так і жіночі аудіофрагменти.

Зкладається із аудіокнижок української літератури, тексти з яких були опубліковані між 1884 і 1964 роками і є у відкритому доступі. Наприклад, «Чорна рада», «Захар Беркут», «Кайдашева сім'я» та інші.

```
kaydasheva_s000017|А як я свисну за садком, чи вийдеш?|А як я свисну за садком, чи вийдеш?;;;
kaydasheva_s000018|а Лаврін усе стояв з Мелашкою і не мав сили одійти од неї.|а Лаврін усе стояв з
kaydasheva_s000019|Люди з кутка позбігались і дивились в ворота й через тин. Декотрі сусіди почали
kaydasheva_s000020|Гм? – мукнув Карпо, стоячи коло хати.|Гм? - мукнув Карпо, стоячи коло хати.;;;
```

Рисунок 3.2 – Приклад описового файлу датасету

Більшість файлів описових даних використовують формат UTF-8, для того, щоб коректно відображати символи української абетки.

Побудовані описові файли наступним чином: спочатку задається назва аудіофайлу, потім текст, який відповідає йому, після того – нормалізований текст, тобто такий, озвучивши який, це буде відповідати словам на аудіо. Наприклад, числа вимовляються повністю(10 – десять), скорочення розкриваються (і т.д. – і так далі) та інші подібні маніпуляції.

Всі файли поставляються у wav-форматі, моно. Частота дискретизації 16000 Гц.

3.1.3 Mozilla Common Voice Ukrainian model

Компанія Mozilla вже кілька років займається підтримкою сервісів для покращення сучасних показників розпізнавання та генерації мовлення. Кожен може зробити свій внесок та оцінити генеровані аудіофрагменти, або транскриптувати(записати текст) оригінальні аудіофрагменти, або згенерувати аудіо із тексту.

Це робиться станом на 2022 рік для 87 мов, і для української в тому числі.

Розмір україномовної частини датасету складає 2 ГБ. Кількість записаних годин – 76, з них перевірено – 63. Формат аудіо - MP3. Кількість голосів – 684.

Варто відзначити, що привабливість цього датасету в тому, що він постачається із додатковими даними щодо виконавця, а саме: вік, стать та акцент.

client_id	path	sentence	up_votes	down_votes	age	gender	accents	locale	s
d4bed3c8900d5c97a	common	А совість - це така культура, що не кожен її ку	2	0	fourties	male		uk	
d4bed3c8900d5c97a	common	Попри всю свою хоробрість, демократи прак	2	0	fourties	male		uk	
d4bed3c8900d5c97a	common	Тобто фактично ми можемо...	2	0	fourties	male		uk	
d4bed3c8900d5c97a	common	Корова це дрібниця порівняно з тим.	2	0	fourties	male		uk	
d4bed3c8900d5c97a	common	Це за даними внутрішнього дослідження од	2	0	fourties	male		uk	
d4bed3c8900d5c97a	common	Прокурор тим часом переглянувся з Другим:	2	0	fourties	male		uk	

Рисунок 3.3 – Приклад датафрейму датасету від Мозіла

По заверенням розробника, це має покращити якість обробки даних. Це може також використовуватись для більш детального аналізу.

3.1.4 VoxForge Repository

VoxForge — датасет мовлення із відкритим кодом, який був налаштований для збору транскрибованого мовлення для використання з

безкоштовними механізмами розпізнавання мовлення. Користувачі можуть записувати і транскрибувати свої аудіофрагменти.

Датасет невеликий — усього коло 1 години тривалості. Аудіофрагменти постачається з частотою дискретизації у двох варіантах — 8 та 16 кГц.

3.2 Процес навчання TTS

3.2.1 Конфігурація

Конфігурація представляє собою json файл, у якому описані параметри навчання. Де-які із параметрів представлено на рисунку 3.2. Також файл конфігурації містить параметри слоїв мережі. Параметри мережі декодера представлено у лістингу 3.1.

Лістинг 3.1.

```
"optimizer": "AdamW",
"optimizer_params": {
  "betas": [
    0.8,
    0.99
  ],
  "eps": 1e-09,
  "weight_decay": 0.01
},
```

Лістинг 3.2.

```
"resblock_kernel_sizes_decoder": [
  3,
  7,
  11
],
"resblock_dilation_sizes_decoder": [
  [
    1,
    3,
    5
  ],
  [
    1,
    3,
    5
  ],
  [
    1,
    3,
    5
  ]
],
"upsample_rates_decoder": [
  8,
  8,
  2,
  2
],
```

3.2.2 Препроцесінг

Датасет для навчання складається із великої кількості даних із різних джерел. Для того, щоб усі дані могли коректно оброблятися, потрібно привести їх до однакових властивостей.

3.2.2.1 Приведення до однакової дискретизації

Першочерговою властивістю аудіофайлів є дискретизація(sample rate). Прийнято використовувати 8, 16, 22 та 43 кГц. У нашому випадку мережа налаштована на те, щоб обробляти аудіофайли із дискретизацією 16 кГц.

3.2.2.2 Приведення до однакової файлової структури

Аудіодані із різних джерел мають різні файлові структури. Десять файлів метаданих називаються одним чином, десять іншим. Десять рівень вкладеності файлів один, десять інший. Інколи, коли даних у датасеті дуже багато, це призводить до додаткової вкладеності. Все це означає, що структура файлів повинна бути однаковою, щоб скрипт міг коректно прочитати усі файли.

3.2.2.3 Приведення до однакового формату

Зазвичай файли аудіозапису поставляються у форматі «.wav». Але іноді трапляються такі, які мають інший формат. Наприклад mp3, ogg. Не менш важливо, щоб усі файли мали однаковий формат, бо інакше вони просто не зможуть бути прочитані.

3.2.2.4 Приведення до однакової структури метаданих

Різні датасети мають різні описові файли(метадані) аудіофайлів. Деякі у форматі txt, інші – csv, ще інші – data. Попри це, вони можуть мати різні структури. Наприклад у першій колонці мати назву файлу, у другій –

його текст, а у іншому випадку – навпаки. Також трапляється, що шлях до аудіофайлу задано без формату файлу, а десь – із його урахуванням. Без даної частини препроцесінгу файли можуть бути прочитані невірно.

3.2.3 Тренування

Використано рішення від soqui-TTS[29].

Лістинг 3.3

```
!CUDA_VISIBLE_DEVICES="0" python /content/TTS/TTS/bin/train_tts.py \
  --config_path /content/DS/config.json \
  --restore_path /content/DS/model.pth.tar
```

Модель була натренована за допомогою AdamW оптимізатора [28] з параметрами $\beta_1 = 0.8$, $\beta_2 = 0.99$ та зниженням ваги $\lambda = 0.01$. Темп навчання встановлено на значення 1×10^{-9} . Використовувався один графічний процесор NVIDIA V100 GPUs. Тренування тривало до 260 тисяч кроків.

Після тренування точність моделі можна показати на рисунку 3.4.

```
| > avg_loader_time: 1.89823 (+0.02439)
| > avg_loss_gen: 2.77929 (-0.34105)
| > avg_loss_kl: 3.31358 (+0.02596)
| > avg_loss_feat: 6.69060 (-0.12185)
| > avg_loss_mel: 17.60333 (-0.08725)
| > avg_loss_duration: 1.56241 (-0.01183)
| > avg_loss_0: 31.94921 (-0.53602)
| > avg_loss_disc: 2.34168 (+0.15830)
| > avg_loss_1: 2.34168 (+0.15830)
```

Рисунок 3.4 – Результати навчання

Основний показник точності – avg_loss_mel. Похибка становить коло 17%. Це означає, що мел-спектральні частоти можна відтворити із досить високою точністю. Інші показники взагалом також мають не дуже великі значення, а отже, навчання пройшло успішно.

3.3 Програмування веб-застосунку

3.3.1 Налаштування серверу Django

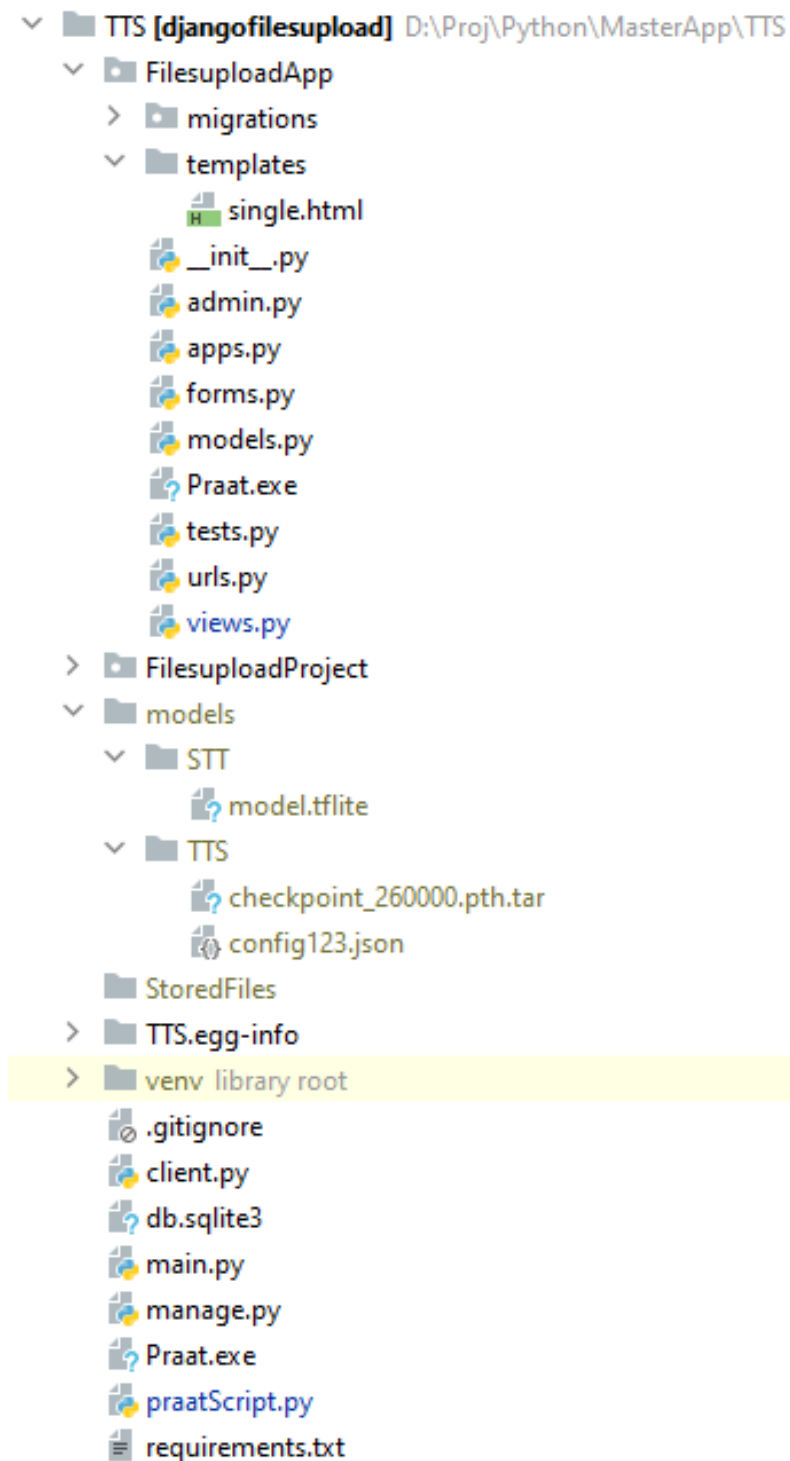


Рисунок 3.5 – Структура проекту

Деякі файли, які треба зазначити, наведені у таблиці 3.1

Таблиця 3.1 – Файли проєкту

Файл/папка	Призначення
model.tflite	Модель STT. Взята також із соqui-STT
checkpoint_260000.pth.tar	Модель TTS
config.json	Конфігураційний файл TTS
client.py	Скрипт для роботи з STT
praatScript.py	Скрипт для генерації praat-файлів.
Praat.exe	Додаток Praat. Виконує скрипти, згенеровані у praatScript.py. У роботі використовуються для заміни характеристик аудіо, через що надається емоційне забарвлення.

3.3.2 Зберігання файлів сесії перекладу

Це початковий етап роботи програми. Для кожної процедури (сесії) перекладу окремого відео створюється папка у каталозі. Там будуть зберігатися усі необхідні файли для маніпуляцій з відео.

3.3.3 Екстрагування аудіо із відео

За допомогою бібліотеки `moviepy` виконується екстрагування аудіодоріжки із відео. Потім вона зберігається у відповідну папку сесії.

3.3.4 Зміна дискретизації

Де-які аудіодоріжки можуть бути різного рівня дискретизації. Наступна команда бібліотеки `librosa` виконує зміну цієї частоти, для того, щоб усі фрагменти мали однакову частоту.

3.3.5 Виконання частини STT

Наступний етап – це зчитування тексту мовлення із аудіодоріжки. Це виконується за допомогою моделі, позиченої у [29]. У випадку, якщо разом із відео були завантажені субтитри транскрибованого аудіо, вони будуть братися замість того тексту, що отримується із моделі STT. Зчитування субтитрів виконується за допомогою бібліотеки `pysrt`.

3.3.6 Автоматичний переклад

Для того, щоб адекватно перекласти наданий у субтитрах, чи екстрагований із аудіо текст, використовується бібліотека `deep_translator`. Вона об'єднує у собі такі популярні сервіси перекладу, як Google, DeepL, Yandex і т.д.

3.3.7 Виконання частини TTS

Надалі виконується саме частина синтезування мовлення із до цього перекладеного тексту. Виконується це за допомогою CLI, але може також бути виконано із вихідного коду. Зроблено це для того, щоб не конфліктували версії бібліотек.

3.3.8 Виконання адаптування емоційної частини

Для коректного накладання емоційних рис, оброблятися будуть розбиті на речення аудіофрагменти. Таким чином емоційний посил речення буде зберігатися.

Кожне англomовне речення буде аналізуватися на негативний, позитивний чи нейтральний настрій. Ця процедура виконується за допомогою вже готової моделі, запозиченої у [30]. Приклад даних датасету наведено на рисунку 3.6.


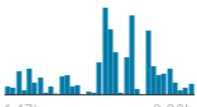
1. target: the polarity of the tweet (0 = negative, 2 = neutral, 4 = positive)					
# 0	# 1467810369	▲ Mon Apr 06 22:19...	▲ NO_QUERY	▲ _TheSpecialOne_	▲ @switch
target	id	date	flag	user	text
		774362 unique values	1 unique value	659775 unique values	15 unique values
0	4	1.47b	2.33b		
0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset can't upc Facebook it... and as a res today ...
0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichar many time ball. Mar save 50% out of bc

Рисунок 3.6 – Витяг із датасету моделі

У розроблювальному додатку на даний момент наявні дві емоції – щаслива і зла.

Ці емоції характеризуються різними властивостями. Оскільки модель TTS генерує нейтральну вимову, треба виходити з того, що цю вимову треба «вдосконалити» таким чином, що б можна було розрізнити одну емоцію від іншої. Такою проблемою займалися вчені із публікації [31](таблиця 3.2).

Таблиця 3.2 – Зміна характеристик аудіо із зміною емоціональних рис.

Parameters	fear	sadness	anger	happiness
$F0_{top}$	-12.5%	-15.8%	+32.6%	+35.6%
$F0_{bottom}$	-0.3%	-8.1%	+17.7%	+24.5%
$F0_{mean}$	-12.4%	-13.9%	+23.3%	+37.2%
$D_{syllable}$	-3.1%	+4.0%	-9.4%	-2.4%
E	-5.1%	-6.8%	+2.0%	+3.3%

Такий взаємозв'язок параметрів надає можливість додавати емоційні риси нейтральному аудіо. Цей конкретний алгоритм зветься «LINEAR MODIFICATION MODEL». У його рамках необхідні нам властивості (Висота, інтенсивність або тривалість) будуть змінюватись у нейтральному аудіо так, щоб досягти бажаної емоції.

Взагалі існують й інші алгоритми, що вирішують цю задачу (Gaussian mixture model (GMM), та classification and regression tree (CART)) [31]. Для них характерні більш точні й складні маніпуляції із аудіо, але в цій роботі буде використовуватися перший алгоритм.

Маніпуляції із аудіо зручно проводити із ПЗ «Praat». Попри візуальний інтерфейс, програма дозволяє виконувати скрипти. Саме через скрипт, який додає необхідні адаптовані параметри до нейтрального аудіо, у цій роботі будуть розрізнятися емоції. Приклад спектрограми нейтрального аудіо наведено на рисунку 3.7. Приклади спектрограми аудіо із емоцією «веселий та злий» наведено на рисунках 3.8 та 3.9 відповідно.

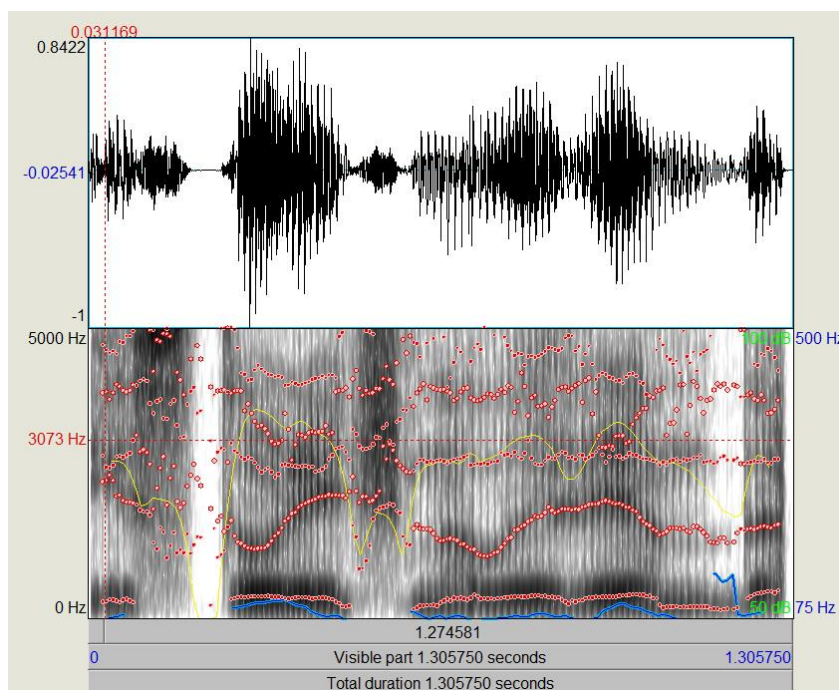


Рисунок 3.7 – Оригінальна спектрограма

Далі із нейтральним аудіо виконуються зміни параметрів. Після виконання необхідних маніпуляцій, а саме збільшенню піків частоти на 35,6 відсотків, прогавин частоти на 24,5 %, середньої частоти на 37,2 %, зменшенню тривалості голосних на 2,4 % та збільшенню амплітуди мовлення на 3,3 %, отримується наступна спектрограма.

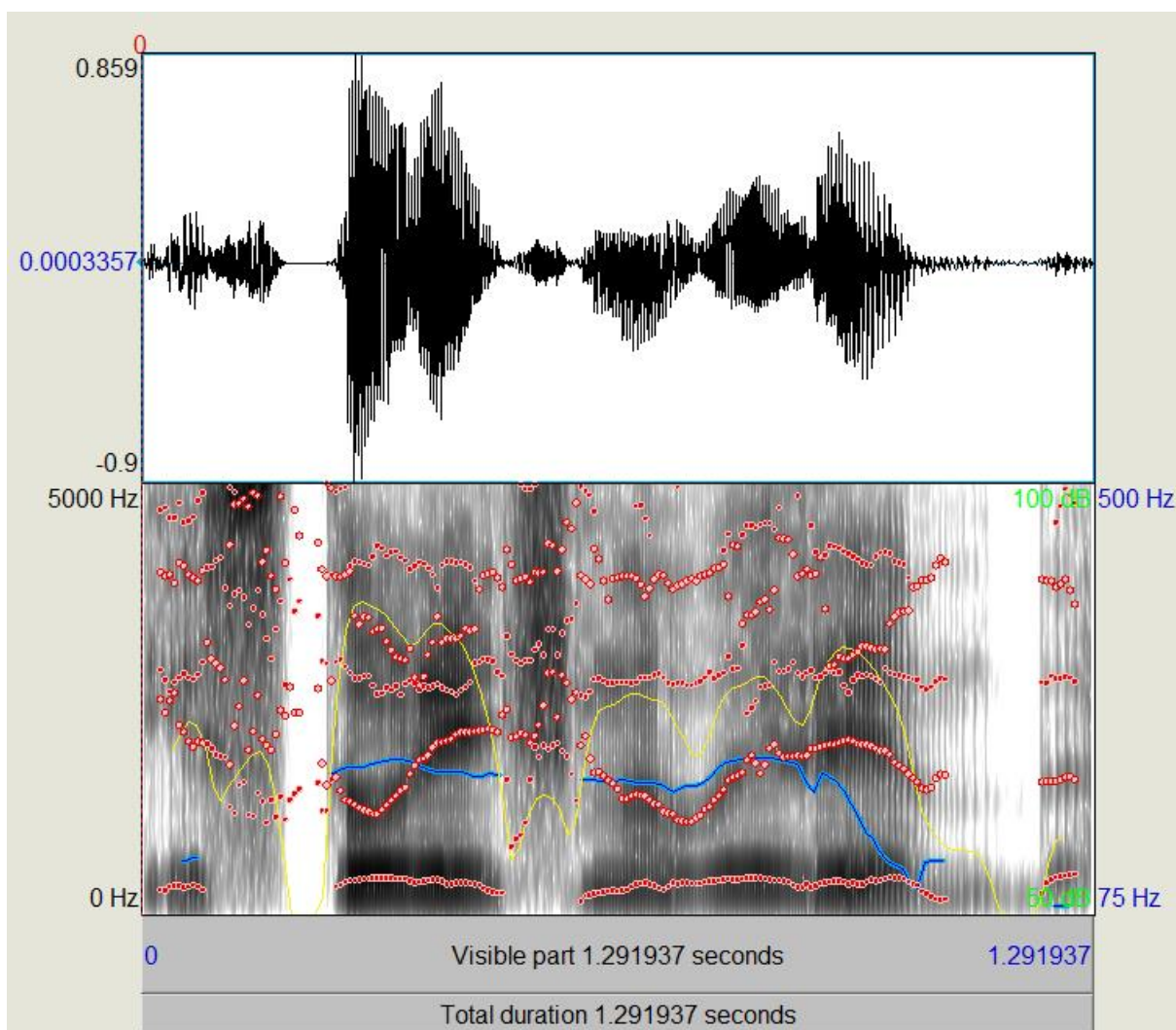


Рисунок 3.8 – Спектрограма до емоції «веселий»

Відповідні маніпуляції відбуватимуться також для емоції «злий». Значення змінюваних параметрів наведену у таблиці. На рисунку наведено спектрограму для зміненого аудіо із використання емоції «злий».

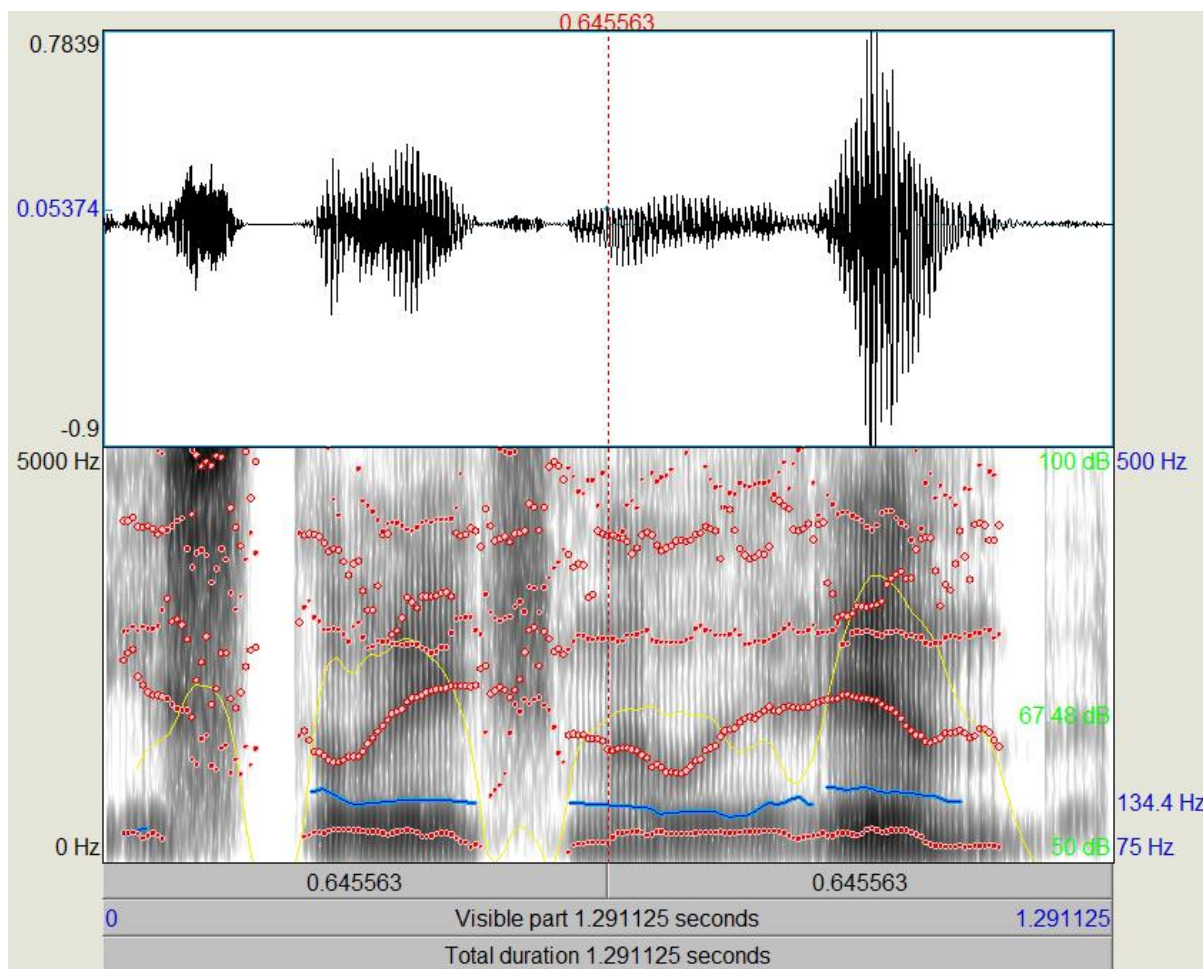


Рисунок 3.9 – Спектрограма аудіо після накладання емоції «злий»

Після накладання емоційних рис, аудіо справді перетворюється та принаймні створюється враження, що це не однакові аудіо. Попри це дійсно стала присутня деяка емоціональна забарвленість. У перспективі можна здійснювати більш складні перетворення із аудіо. Приклади деяких перетворених аудіо можна знайти за посиланням <https://gopnikada.github.io/> у моєму репозиторії.

3.3.9 Перевірка якості зміни емоцій

Задля перевірки якості накладання емоцій, використовується модель, опублікована у [32]. В рамках цієї роботи, авторами побудовано модель,

яка робить передбачення відносно емоції, що наявна у аудіо. Варто відмітити, що ця модель може працювати у режимі реального часу.

Приклад роботи програми наведено на рисунку 3.10. Емоції, що можуть бути класифіковані: anger, boredom, disgust, fear, happiness, neutral, sadness.

Загальна точність передбачення емоції голосу цієї моделі - 68.9 % [32], що вважається задовільним.

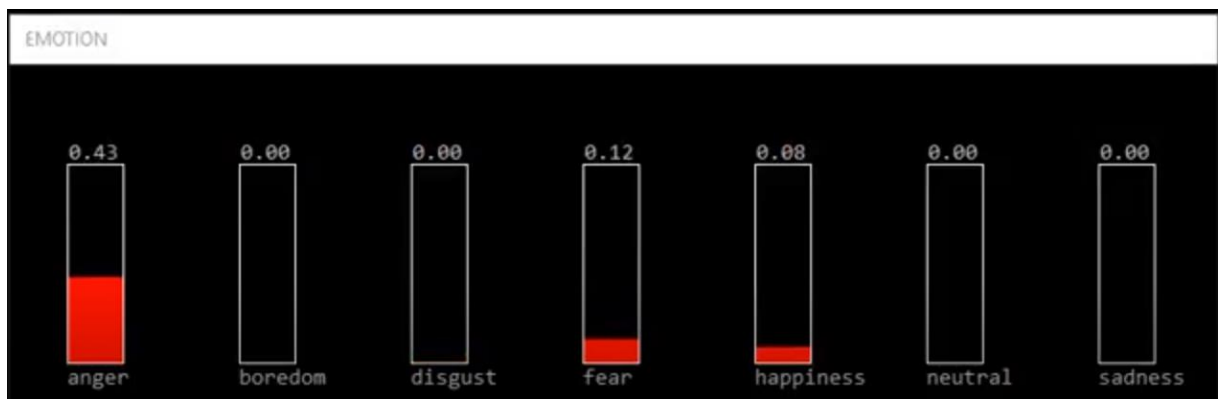


Рисунок 3.10 – Вікно роботи програми

У рамках моєї роботи цю модель було використано для перевірки коректності змінених аудіо.

Було записано 100 нейтральних аудіофрагментів та відкореговано кожний для досягання двох емоцій: «щасливий» та «злий» по 100 аудіофрагментів для емоцій «щасливий» та 100 для «злий».

Після цього відповідні емоційні аудіофрагменти було проаналізовано за допомогою моделі розпізнавання емоцій.

Для усіх фрагментів емоції «щасливий», загальна точність її розпізнавання склала 65,3 %.

Для усіх фрагментів емоції «злий», загальна точність її розпізнавання склала 59,7 %.

3.3.10 Накладання зміненого аудіо на відео

Після виконання адаптування емоційних складових, згенерована аудіодоріжка заміняє собою оригінальну. Важливо віжмитити, що при цьому довжина аудіодоріжок повинна бути однаковою. Ці операції виконуються також за допомогою бібліотеки `moviepy`.

3.3.11 Повернення зміненого відео клієнту

Після виконання необхідних процедур, клієнт отримує остаточний варіант відео.

ВИСНОВКИ ДО РОЗДІЛУ 3

Було покроково показано процес створення застосунку. Побудовано модель нейтральної вимови. До цього надано датасети та відповідні конфігурації тренування а також наведено результати моделі.

Розглянуто модель по виявленню емоційних рис із англомовного тексту. Цю модель застосовано у додатку.

За допомогою передбачених із тексту емоційних рис, відбувається накладання цих рис на нейтральну вимову за рахунок зміни параметрів аудіофрагментів.

Імплементовано додаток по автоматизованому перекладу англомовного відео в українськомовне і надано відповідний код застосунку.

4 РОЗРОБКА СТАРТАП-ПРОЕКТУ

Принципова ідея проекту

У цьому підрозділі зроблено аналіз та надано у таблицях наступне:

- ідея – додаток з автоматизованого озвучування перекладаних субтитрів відео;
- напрямки використання, які можуть бути;
- у чому виграє юзер;
- що відрізняє систему від подібних рішень.

У таблиці 4.1 описана ідея стартап-проекту, а саме розкритий зміст ідеї, наведені напрямки застосування та сформульовані основні вигоди, що отримує користувач.

Таблиця 4.1 – Опис ідеї стартап проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка ПО для синхронного перекладу і озвучування тексту, який наявний у мовленні відео.	1. Перегляд відео на мові, якою користувач краще володіє.	Користувачу треба менше часу, щоб зрозуміти зміст відео, оскільки не потрібно мануально перекладати кожне незнайоме слово.
	2. Інтегрування в існуючий сервіс.	Якщо користувач – компанія, яка займається хостингом відео, то інтегрування такого додатку покращить показники переглядів

Проаналізовано потенційні економічні та технічні переваги стартап проекту. Відповідні результати наведено у таблиці 4.2, а саме:

- технічні та економічні властивості ідеї стартапу
- перелік конкурентів/товарів-замінників/аналогів, існуючих на ринку; зібрано інформацію стосовно техніко-економічних показників для ідеї проекту та конкурентів;
- проаналізовано показники: для розроблюваної ідеї визначено ті, що мають:
 - 1) гірші значення (W, слабкі);
 - 2) аналогічні (N, нейтральні) значення;
 - 3) кращі значення (S, сильні).

Для порівняння були обрані два найбільш схожі конкуренти, а саме Microsoft Translator та Veed.io.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	характеристики ідеї	Концепції конкурентів			Слабкі (W), нейтральні (N) та сильні (S) сторони		
		Мій проект	Microsoft Translator	Veed.io	W	N	S

1.	Переклад тексту у довільну мову	Ні	Так	Так	+		
2.	Переклад	Так	Так	Ні, тільки текст			+
3.	Вартість	Для користувача безкоштовно	Безкоштовно	Коштовно			+
4.	Час на сприйняття контенту	Найменше часу	Менше часу	Багато часу		+	
5.	Можливість вбудувати додаток в інші сервіси	Так	Так	Ні			+

Зробивши порівняльний аналіз побудованого рішення, та інших подібних рішень, було зроблено висновок, що побудованому рішенню притаманно більше сильних сторін, ніж аналогам.

4.2 Технологічний аудит ідеї проекту

У рамках цієї секції проведено аудит технологічних рішень, використовувати котрі можливо реалізувати проектну ідею.

Технологічну здійсненність ідеї проекту було проведено за допомогою аналізу таких складових:

- технологія, за допомогою якої побудовано продукт
- чи існує вже технологія, або ж вона потребує розробки
- доступність технології.

Технологічну здійсненність ідеї проекту наведено у таблиці 4.3.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технологія реалізації	Наявність технології	Доступність технології
1.	Розпізнавання тексту мовлення з відео	Google Speech-To- Text	Наявна	Безкоштовна до 60 хвилин аудіо, далі \$0,006/15 секунд
2.	Переклад тексту	Google Textt o-Speech	Наявна	Платна, \$4,00 USD / 1 мільйон символів

3.	Синтезування мовлення	VITS End-To-End TTS model	Наявна	Безкоштовна
4.	Платформа керування синтезованим мовленням	coqui-TTS[29]	Наявна	Безкоштовна
5.	Накладання аудіохарактеристик первинного аудіофрагменту на синтезований	Praat	Наявна	Безкоштовна
6.	Накладання синтезованого мовлення поверх оригінального із урахуванням звуків оточення.		Наявна	Безкоштовна

Згідно аналізу, технологічна і технічна реалізація цього проекту можлива. Визначено технологічний стек, варто виконувати ідею проекту.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

В межах даного підпункту було визначено ринкові можливості, які можна використати під час ринкового впровадження проекту, та ринкові загрози, які можуть перешкодити реалізації проекту, що дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового

середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Перш за все, у таблиці 4.4 було проведено аналіз попиту, а саме наявність попиту, обсяг, динаміка розвитку ринку.

Таблиця 4.4 – Попередня характеристика потенційного ринку стартаппроекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Головних гравців(кількість), од	2
2.	Обсяг загальних продажів, грн/ум.од	250 грн/ум.од міс
3.	Ринкова динаміка	Зростає
4.	Наявність обмежень для входу	Немає
5.	Специфічні вимоги до стандартизації та сертифікації	Немає
6.	Середня норма рентабельності в галузі, %	Відсутні дані

Відповідно до поперенього аналізу із таблиці, даний ринок привабливий для входження.

У таблиці 4.5 надано групи потенційних клієнтів, яким розроблювальна система может показатися цікавою. Попри це сформовано перелік вимог продукту кожної групи.

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимог и спожив ачів до товару
1.	Можливість сприймати іномовне відео без перешкод	Фізичні особи, які хочуть сприймати іномовні відео, але не володіють цією мовою на тому рівні, що це робиться без перешкод.	Школи та інші навчальні заклади можуть надавати знижку учасникам, що впливає на кількість користувачів	Існуюч ий веб інтерфе йс або веб застосу нок чи мобиль ний додато к. Зрозумі лість побудо ви інтерфе йсу. Достат ній рівень якості перекл аду та синтезу мовлен ня. Просто та

				викори стання
2.	Інтеграція застосування в якості сервісу для інших додатків	Фізичні або юридичні особи, які володіють відео-хостингами та мають на меті покращити якість наданих послуг.	Не визначно	Наявність гнучкого, виразного та функціонального API, що дозволяє тонким чином сконфігурувати сервіс. Достатня

				швидко дія, можлив ість розпара лелити задачу. Високи й рівень якості перекл аду та синтезу мовлен ня
--	--	--	--	--

За аналізом потенційних груп клієнтів слідкує аналіз ринкового середовища та побудовано таблиці факторів, що уможливають ринкове впровадження проекту а також фактори, що можуть перешкоджати цьому впровадженню.

Фактори, що сприяють ринковому впровадженню проекту наведені у таблиці 4.6.

Таблиця 4.6 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
----------	--------	------------------	-----------------------------

1.	Наявність функціоналу, якого немає у конкурентів	Немає конкурентів, у яких повністю наявні функції, наведені у розроблюваному додатку.	Акцент на переваги продукту над конкурентами. Розгортання більшого числа функцій.
2.	Зростання кількості контенту на англійській мові	Розширення можливостей для сприйняття англомовного контенту.	Розгортання на глобальному ринку.

Фактори, що перешкоджають ринковому впровадженню проекту наведені у таблиці 4.7.

Таблиця 4.7 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Конкуренція	Нові конкуренти на ринку	Зміна ціноутворення. Додавання нового функціоналу.

			Інтегрування у велику кількість сервісів.
2.	Непрогнозована велика популярність системи	Бракує обчислювальних можливостей для адекватної роботи	Горизонтальне масштабування.

Наступною наведено пропозиційний аналіз: показано загальні риси конкуренції на ринку та ступеневий аналіз конкуренції у таблиці 4.8

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
1. Чиста	Існують додатки з подібним функціоналом	Додаток повинен мати широкий функціонал.

2. Національна	Направленість додатку виражається особливостями, притаманними людям конкретної країни	Додати інші мови для обробки.
3. Внутрішньо галузева	Товари задовольняють одну потребу, мають відмінності в ціні, функціях, інтерфейсі	Низька ціна, бонуси, знижки за паушальними закупівлями.
4. Товарно-видова	Існують застосування зі схожим функціоналом	Створити особливі можливостей для юзера.
5. Нецінова	Більшість додатків безкоштовні	Надання більшої кількості функцій
6. Не марочна	Більшість компаній ноунейми	Інтеграція із популярними сервісами та клієнтами.

Після аналізу конкуренції в таблиці 4.9 проведено детальніший аналіз умов конкуренції.

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові в аналізі	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари заміни

	Додатки-прекладачі, додатки із синтезу мовлення	Додатки аналоги з опрацювання тексту на аудіо	Мережа Інтернет	фінансовий, споживчий, освітній	Замінники простіші для розуміння, дешевші
Висновки:	Конкуренція низька	Поки не анонсовані	Відповідність стандартам	Не диктують правил, масовий ринок	Безкоштовний період для тестування

Згідно з результатами таблиця щодо конкуренції, було зроблено висновок, що ринок відкритий для того, щоб увійти і потребує товару, що буде найрацим за існуючі рішення та давати більше функціоналу, ніж конкурентів.

Відповідно до аналізу конкуренції й урахуванням характеристик ідеї проекту, вимог споживачів до товару та маркетингово-середовищний факторів, було зазначено та наведено перелік факторів конкурентоспроможності у таблиці 4.10.

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)

1	Новизна	Досі не існує конкурентів, які б виконували повний набір заданого функціоналу. Взагалом вони зосередженні ні виконанні однієї чи кількох функцій.
2	Підтримка	Підписка на додаток буде супроводжуватися постійними оновленнями, які покращуватимуть додаток, додаватимуть більше функціоналу і можливість позбавлення від багів та недосконалостей.
3	Простота користування	Програмний додаток має абстрагувати від деталей розробки та давати можливість зосередитися на сприйнятті контенту.
4	Швидкодія	Сервіс повинен досить швидко обслуговувати запити користувача.

За визначеними факторами конкурентоспроможності проведено аналіз сильних та слабких сторін стартап-проекту у таблиці 4.11

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів- конкурентів у порівнянні з «veed.io»					
			- 3	- 2	- 1	0	+1	+2 +3
1	Новизна	20	+					
2	Підтримка	17			+			
3	Простота користування	20				+		
4	Швидкодія	18				+		

Останнім кроком аналізу ринкових можливостей з впровадження проекту є складання SWOT-аналізу за допомогою обраних ринкових загроз та можливостей, та сильних і слабких сторін (табл. 11).

У таблиці 4.12 наведено SWOT-аналіз стартап-проекту.

Таблиця 4.12 – SWOT- аналіз стартап-проекту

Сильні сторони: унікальний функціонал, виразливий API, простий інтерфейс	Слабкі сторони: треба постійно впроваджувати нові оновлення із покращення якості сервісу
Можливості: приваблива ціна	Загрози: відсутність інформації про продукт, поява конкуренції

За допомогою SWOT-аналізу розроблено альтернативи ринкової поведінки для впровадження стартап-проекту до ринку та орієнтовний час

ринкової реалізації відповідно до потенційних проектів конкурентів, що можуть бути виведені до ринку.

Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів у таблиці 4.13

Таблиця 4.13 – Альтернативи ринкового впровадження стартапу

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Активна реклама	Висока. Товар на слуху	6 міс на рекламну компанію
2	Система бонусів	Середня, користувачі можуть звикнути до функціоналу	1 рік
3	Колоборації із школами, університетами	Середня, поповнення користувачів за рахунок клієнтів шкіл та університетів	3 роки

4.4 Розроблення ринкової стратегії

Першим кроком передбачається визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів. Опис цільових груп потенційних споживачів наведено у таблиці 4.14.

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№ п/ п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Фізичні особи	Готові	Високий	Низька. Аналогічних додатків немає	Просто, конкуренти не являють собою загрозу.
2	Застосунки- відео- хостинги	Можуть розглянути як якісне покращення їхнього сервісу.	Середній	Низька, немає аналогів.	Складніть є середньою, бо процедура запровадження нових функцій від сторонніх виробників займає час і передбачає відповідні перевірки.
Які цільові групи обрано: Фізичні особи, застосунки- відео-хостинги					

За результатами аналізу потенційних груп споживачів (сегментів) було обрано цільові групи, для яких планується пропозиція товару, та визначено стратегія охоплення ринку за наступною логікою :

- у разі, базування фокусу на одному сегменті обереться стратегія концентрованого маркетингу;
- у разі, колоборації із кількома сегментами, зокрема розробляючи для них окремо програми ринкового впливу – використається стратегія диференційованого маркетингу;
- у разі співпраці із усім ринком, пропонується стандартизована програма використагтя масового маркетингу.

Для роботи в обраних сегментах ринку було сформувано базову стратегію розвитку (таблиця 4.15) та сформовано базову стратегію конкурентної поведінки (таблиця 4.16).

Таблиця 4.15 – Визначення базової стратегії розвитку

№ п / п	Обрана альтернати ва розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспром ожні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
------------------	---	------------------------------	--	---------------------------------

1	Орієнтація на загальних користувачів	Ставка на якість мовлення, різноманіття голосів	Простота у використанні	Стратегія диференціації – товар відмінний від конкурентів за своїм функціоналом (персоналізація під користувача)
---	--------------------------------------	---	-------------------------	--

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Так	Не буде	Можливо, алгоритми обробки голосу	Стратегія лідера: наступальна (охопити максимально ринок, забрати користувачів у конкурентів)

Відповідно до вимог споживачів з цих сегментів до стартап компанії та до кінцевого продукту, а також у відповідності до обраної базової стратегії з розвитку та стратегії з конкурентної поведінки було спроектовано стратегію позиціонування. Вона полягає у формуванні такої ринкової позиції у якій споживачі мають ідентифікувати проект. Стратегія позиціонування описана у таблиці 4.17.

Таблиця 4.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартаппроекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Якість перекладу	Стратегія диференціації	Висока якість перекладу безкоштовно.	Зручий та швидкий якісний переклад.
2	Якість синтезу мовлення	Стратегія диференціації	Мовлення звучить натурально.	Натуральне звучання мовлення.
3	Зручність інтерфейсу	Стратегія диференціації	Великі кнопки, мало текстового вводу	Задовільна робота на усіх девайсах, на яких є браузер та Інтернет

4	Наявність зручного API	Стратегія диференціації	Зручний інтерфейс для взаємодії із додатком.	Швидке та інформативне звернення до додатку.
---	------------------------	-------------------------	--	--

4.5 Розроблення маркетингової програми стартап-проекту

Для того, щоб сформувавши маркетингову програму спочатку треба навести маркетингову концепцію товару, що отримає споживач. У таблиці 4.18 наведено відповідні результати аналізу конкурентоспроможності товару.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Подивитися іноземне відео українською	Можливість прослуховувати відео українською, які оригінально було на одній мові.	Можливість користувача не витрачати час на переклад контенту відео.

Надалі розроблено тривірневу маркетингову модель товару: уточнено ідею продукту та/або послуги, його фізичні складові, особливості процесу його надання.

Трирівнева маркетингова модель представлена у таблиці 4.19.

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Застосунок для перекладу іноземних відео.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Вартість обслуговування	М	Вр
	2. Знижки	Нм	Вр
	3. Безвідмовність	М	Тх
	4. Собівартість	М	Тх
	5. Зручність інтерфейсу	М	Е
	6. Стильний дизайн	Нм	Ор
	7. Вимогливість до ресурсів девайсу	М	Тх
	Якість: стандарти ISO для ПО.		
	Пакування: веб-застосування		
Марка: OLEG LLC, SmoothView			
III. Товар із підкріпленням	До продажу – знижка на підписку на 6 і більше місяці 15%		
	Після продажу – безкоштовні оновлення		
За рахунок чого потенційний товар буде захищено від копіювання: захист алгоритмічної реалізації			

Далі визначаються цінові межі (Таблиця 4.20)

Таблиця 4.20 – Визначення меж встановлення ціни

№ п/ п	Функціонал	Рівень цін на товаризамінни ки	Рівень цін на товарианало ги	Рівень доходів цільової групи споживач ів	Верхня та нижня ціна
1	Перегляд відео до 1 год.	Безкоштовні	Безкоштовні	12 – 16 тис грн	Нижня: Безкоштовні Верхня: Безкоштовні
2	Перегляд відео від 1 до 3 год.	-	Безкоштовні і або 2-4 долари	12 – 16 тис грн	Нижня: Безкоштовні Верхня: 4 долари
3	Користуван ня API	-	Безкоштовні і, але мають обмеження або 4-6 доларів	12 – 16 тис грн	Нижня: 3 долари Верхня: 7 доларів
3	Загалом	-	-	12 – 16 тис грн	Нижня: Безкоштов ні Верхня: 11 доларів

Наступним кроком це визначення оптимальної системи збуту продукту, в межах якого приймається наступне рішення:

- проводити збут самостійно чи звертатися до посередників
- вибір та обґрунтування оптимальної глибини каналу збуту;
- вибір та обґрунтування виду посередників.

Система збуту продукту описана у таблиці 4.21

Таблиця 4.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Безкоштовні версії товару протягом 14 днів. У разі, якщо зацікавленості оплата щомісячно.	Реклама, місце в рейтингу програм	Канал одного рівня	Вебзастосування

2	Готові платити можливість програмно управляти застосунком.	Реклама, підтримка функції акції та бонуси за користування	Канал дворівневий	Вебзастосування
---	--	--	-------------------	-----------------

Останнім кроком маркетингової програми являється розробка концепції маркетингових комунікацій, що спирають на обрану основу для позиціонування. Вона наведена у таблиці 4.22.

Таблиця 4.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
-------	---------------------------------------	--	--	----------------------------------	--------------------------------

1	Очікують максимальну простоту сприйняття контенту іншомовного походження	Соцмережі, пошта, сповіщення на смартфон	Комплексни й підхід	Повідомлення має переконати споживача що користування додатком покращити обізнанність у іншомовних сегментах відео- контенту.	Without boundaries. So. Now!
---	--	---	------------------------	--	------------------------------------

ВИСНОВКИ ДО РОЗДІЛУ 4

У розділі сформульовано та надано опис ідеї стартап-проекту з аналізом потенційно існуючих техніко-економічних переваг ідеї. Проведено аудит використаної технології, за допомогою якої можливо реалізувати ідею цього проекту та зроблено визначення технологічний плану, який є оптимальним для реалізації ідеї.

Визначено ринкові можливості, що можна використати у ринковому впровадженні проекту, та загрози ринкову, що можуть потенційно перешкодити реалізації цього проекту. Попри це було проаналізовано стан середовища ринкового, потреб клієнтів та пропозицій конкурентів.

Як результат було визначено, що ідея стартап-проекту з технічної точки зору може бути реалізована і також має високу конкурентоспроможність. Розроблено ринкову стратегію (стратегії охоплення ринку, базова стратегія розвитку, стратегія позиціонування.)

В кінці розділу розроблено маркетингову програму стартап-проекту.

ВИСНОВКИ

У дисертації обгрунтовано важливість розробки систем, які покращують сприйняття контенту саме відеоформату. Насьогодні більшість інформації передається саме у відео. Поперше, інформацію так легше зприймати. По-друге, це дозволяє робити інші речі одночасно із прослуховуванням. По-третє, саме ментальна навантаженість уможливує більш виразливе сприйняття. Із тексту може бути інколи не зрозуміло, яке саме враження мається на увазі та який посил хотів передати автор. Також треба додати, що відео сприймається більш уважно, особливо, коли присутні яскраві образи, люди, чи наведена схематична інформація. Треба також відмітити, що частина відеоконтенту взагалі становиться більше. Все більше кіно, серіалів, лекцій і т.д.

Проблематика, яка ставиться у роботі, актуальна у наш час. Ми живемо у час глобалізації, і домінуюча мова сьогодні – англійська. Усі новітніші та актуальніші лекції та кіно зазвичай виходять саме англійською.

Це ставить бар'єри для людей, які не володіють цією мовою.

Уже існують часткові рішення цієї проблеми, а саме, автоматизовані перекладені субтитри, як, наприклад у Youtube чи Netflix. Але все ж таки це тільки текст. Тобто велика частина інформації все ж втрачаються. Також по суті, із відео залишається тільки читання тексту, бо при прочитуванні важко зосередитися на змісті. Отже, цей підхід не є остаточним у вирішенні цієї проблеми.

Система, що розроблюється у цій дисертації має на меті підвищити рівень сприйняття іншомовного відно(у цьому випадку – із англійської в українську, але можливо зробити для інших мов). Це зробить перегляд більш зручним та зрозумілим а також допоможе зосередитись на змісті і контексті відео, передавши якщо не всю, то більшу частину інформації вихідного відео.

Зпочатку зроблено огляд технологій, які можуть надавати часткове рішення цієї проблеми, підкреслено їх недоліки та надано приклад системи, що оптимально б розв'язувала проблему.

Для побудови системи було побудовано модель відтворення українського тексту в україномовне аудіо. При цьому точність відтворюваного мовлення вважається задовільною.

Для більш органічного звучання голосу, згенероване із тексту мовлення забарвлюється емоційними рисами. У роботі зроблено акцент на емоції щастя та злості. Їх легше відтворити, ніж інші, такі як сонливість, утомленність, страх. Але вдосконаливши систему, їх теж можна відтворити. Якість додання емоційних рис вважається прийнятною.

Побудовано повноцінний веб- додаток, який у сприятливій для користувача манері, виконує усі перераховані процедури. Цей додаток може бути застосован у якості стороннього API або сервісу та підключен до інших сервісів чи додатків. Також він може бути розширен через додання нових мов та емоцій.

Також надано стартап-модель да цієї системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. ITU-T Rec. P.10 (2006) Vocabulary for performance and quality of service
2. P Seeviour, J Holmes, and M Judd. Automatic generation of control signals for a parallel formant speech synthesizer. In ICASSP'76. IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 1, pages 690–693. IEEE, 1976.
3. Dennis H Klatt. Software for a cascade/parallel formant synthesizer. the Journal of the Acoustical Society of America, 67(3):971–995, 1980.
4. Jonathan Allen, Sharon Hunnicutt, Rolf Carlson, and Bjorn Granstrom. Mitalk-79: The 1979 mit text-to-speech system. The Journal of the Acoustical Society of America, 65(S1): S130–S130, 1979.
5. Cecil H Coker. A model of articulatory dynamics and control. Proceedings of the IEEE, 64(4): 452–460, 1976.
6. Christine H Shadle and Robert I Damper. Prospects for articulatory synthesis: A position paper. In 4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis, 2001.
7. Joseph Olive. Rule synthesis of speech from dyadic units. In ICASSP'77. IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 2, pages 568–570. IEEE, 1977.
8. Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. In Sixth European Conference on Speech Communication and Technology, 1999.
9. Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 2016.

10. Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. In Sixth European Conference on Speech Communication and Technology, 1999.
11. <https://wiki.aalto.fi/pages/viewpage.action?pageId=149890776>
12. <https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC>
13. <https://www.g2.com/compare/amazon-polly-vs-ibm-watson-text-to-speech-vs-google-cloud-text-to-speech-vs-azure-text-to-speech-api>
14. <https://www.kaggle.com/general/198232>
15. <https://www.fon.hum.uva.nl/praat/>
16. <https://arxiv.org/pdf/2106.15561.pdf>
17. <https://arxiv.org/pdf/2106.06103.pdf>
18. <https://ijdykeman.github.io/ml/2016/12/21/cvae.html>
19. Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In International Conference on Machine Learning, pp. 1530–1538. PMLR, 2015.
20. Kim, J., Kim, S., Kong, J., and Yoon, S. Glow-TTS: A Generative Flow for Text-to-Speech via Monotonic Alignment Search. Advances in Neural Information Processing Systems, 33, 2020.
21. Prenger, R., Valle, R., and Catanzaro, B. Waveglow: A flow-based generative network for speech synthesis. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3617–3621. IEEE, 2019.
22. Kong, J., Kim, J., and Bae, J. HiFi-GAN: Generative Adversarial networks for Efficient and High Fidelity Speech Synthesis. Advances in Neural Information Processing Systems, 33, 2020.
23. Kumar, K., Kumar, R., de Boissiere, T., Gustin, L., Teoh, W. Z., Sotelo, J., de Brebisson, A., Bengio, Y., and Courville, A. C. MelGAN: Generative

Adversarial Networks for Conditional waveform synthesis. volume 32, pp. 14910–14921, 2019.

24. <https://www.kaggle.com/datasets/aikhmelnysky/ukrainian-open-speech-to-text-dataset-42-part-2/code>

25. http://www.caito.de/data/Training/stt_tts/uk_UK.tgz

26. <https://commonvoice.mozilla.org/uk/datasets>

27. <http://www.repository.voxforge1.org/downloads/uk/Trunk/>

28. Loshchilov, I. and Hutter, F. Decoupled Weight Decay Regularization. In International Conference on Learning Representations, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.

29. <https://github.com/coqui-ai/TTS>

30. <https://www.kaggle.com/code/paoloripamonti/twitter-sentiment-analysis>

31. Jianhua Tao, Member, IEEE, Yongguo Kang, and Aijun Li. TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 14, NO. 4, JULY 2006

32. Wagner, Johannes and Lingenfelser, Florian and Baur, Tobias and Damian, Ionut and Kistler, Felix and Andr, Elisabeth - The social signal interpretation (SSI) framework: multimodal signal processing and recognition in real-time. P 831-834. 2013.

5 ДОДАТКИ

ДОДАТОК А. Програмний код додатку

```

import os
from django.http import HttpResponseRedirect
from django.shortcuts import render
from django.http import HttpResponse, HttpResponseNotFound
import praatScript
from .forms import UploadFileForm
from django.views.decorators.csrf import ensure_csrf_cookie
from moviepy.editor import *
import uuid
from os import walk
import librosa
import soundfile as sf
import subprocess
from deep_translator import GoogleTranslator
import pathlib
from client import *
from wsgiref.util import FileWrapper
import pysrt

Rootpath = pathlib.Path().resolve()

saveFolder = Rootpath.joinpath('StoredFiles')
sttModelPath =
Rootpath.joinpath('models').joinpath('STT').joinpath('model.tflite')

@ensure_csrf_cookie
def upload_file(request):
    if request.method == 'POST':
        # form = UploadFileForm(request.POST, request.FILES)
        #if form.is_valid():
        sessionId = str(uuid.uuid1())
        SessionFolderPath = saveFolder.joinpath(sessionId)
        os.mkdir(SessionFolderPath)
        FormFiles = request.FILES

        for fileName, value in FormFiles.items():
            fileToSaveName = value.name
            saveFilePath = SessionFolderPath.joinpath(fileToSaveName)
            handle_uploaded_file(saveFilePath, value)

        filesInDir = next(walk(SessionFolderPath), (None, None, []))[2]
        videoFileName = list(filter(lambda x: x.split('.')[1] != 'srt',
filesInDir))[0]
        subsFileName = list(filter(lambda x: x.split('.')[1] == 'srt',
filesInDir))[0]
        subsFilePath = SessionFolderPath.joinpath(subsFileName)
        audioName = videoFileName.split('.')[0] + '.wav'
        audioName16k = videoFileName.split('.')[0]+"16k" + '.wav'

        audioPath = SessionFolderPath.joinpath(audioName)
        audioPath16k = SessionFolderPath.joinpath(audioName16k)

        videoPath = SessionFolderPath.joinpath(videoFileName).__str__()

```



```

#todo clear
#res = subprocess.run(["ls"], capture output=True)

audioclip = AudioFileClip(videoPath)
audioclip.write_audiofile(audioPath, fps=None, nbytes=2,
                          buffersize=2000,
                          codec=None, bitrate=None,
ffmpeg_params=None,
                          write_logfile=False, verbose=True,
logger='bar')

y, sr = librosa.load(audioPath)
data = librosa.resample(y, sr, 16000)
sf.write(audioPath16k, data, 16000)

ds = Model(sttModelPath.__str__())
fin = wave.open(audioPath16k.__str__(), "rb")
audio = np.frombuffer(fin.readframes(fin.getnframes()), np.int16)
textToTranslate = ''
subs = pysrt.open(subsFilePath.__str__(), encoding='utf-8')

if filesInDir.__len__() == 2:
    textToTranslate = subs.text.replace('\n', ' ')
    for sub in subs.data:
        print(sub)
else:
    textToTranslate = ds.stt(audio)

#todo match time

translatedText = GoogleTranslator(source='auto', target='uk')\
    .translate(textToTranslate)

synthedFileName = "synthed.wav"
synthedSpeechPath = SessionFolderPath.joinpath(synthedFileName)

ttsModelPath = Rootpath.joinpath('models').joinpath('TTS')\
    .joinpath('checkpoint_260000.pth.tar')
ttsConfigPath = Rootpath.joinpath('models').joinpath('TTS')\
    .joinpath('config123.json')

ttsCliCommand = f'tts --text "{translatedText}" ' \
    f'--model_path {ttsModelPath.__str__()} ' \
    f'--config_path {ttsConfigPath.__str__()} ' \
    f'--out_path {synthedSpeechPath.__str__()} '

synthResponse = subprocess.call(ttsCliCommand, shell=True) # 1 -
error, 0 - ok

adjusted_audioFileName = "adjusted.wav"

```

```

        adjusted_audioPath =
SessionFolderPath.joinpath(adjusted_audioFileName)

        praatScriptFileName = "praatScript.praat"
        praatScriptPath = SessionFolderPath.joinpath(praatScriptFileName)

        praatScriptText =
praatScript.generatePraatScriptText(SessionFolderPath.__str__(),
audioName16k,

synthedFileName, adjusted_audioFileName)
        praatScript.createPraatFile(praatScriptText, praatScriptPath)

        adjustedSynthResponse =
subprocess.call(f'{Rootpath.joinpath("Praat.exe")} '
                f'--run "{praatScriptPath.__str__()}"', shell=True) #

        videoclip = VideoFileClip(videoPath)
        audioclip_adjusted = AudioFileClip(adjusted_audioPath.__str__())
        videoclip_changed_audio = videoclip.set_audio(audioclip_adjusted)

        videoclip_adjustedFileName = "adj" + videoFileName
        videoclip_adjustedFilePath = SessionFolderPath.\
            joinpath(videoclip_adjustedFileName).__str__()

        videoclip_changed_audio.write_videofile(videoclip_adjustedFilePath)

        subTitlesdata = ""

        file = FileWrapper(open(videoclip_adjustedFilePath, 'rb'))
        response = HttpResponse(file, content_type='video/mp4')
        response['Content-Disposition'] = f'attachment;
filename={videoclip_adjustedFileName}'

        return response

    else:
        form = UploadFileForm()
        return render(request, 'single.html', {'form': form})

def handle_uploaded_file(path, f):
    with open(path, 'wb+') as destination:
        for chunk in f.chunks():
            destination.write(chunk)

```

ДОДАТОК Б. Схема структурна бізнес-процесу автоматизований переклад відео

