ДОДАТОК А

Перелік джерел посилання за науковими напрями керівника та науковців кафедри програмної інженерії

2. Каук В. І. Безпека даних при використанні хмарних обчислень для розробки програмних систем / В. І. Каук, А. О. Трибух // Поліграфічні, мультимедійні та web-технології : матеріали Молодіжної школи-семінару IX Міжнар. наук.-техн. конф., 14-28 травня 2024 р. – Харків : ТОВ «Друкарня Мадрид», 2024. – Т. 2. – С. 86-87.

11. Khovrat, A., Kobziev, V. Using Recurrent and Convulation Neural Networks to Indentify the Fake Audio Messages, IEEE 7th International Conference on Methods and Systems of Navigation and Motion Control, MSNMC 2023 - Proceedings, 2023, P. 174–177.

12. Kizitskyi,M., Turuta, O., Turuta, O. Improving Speaker Verification Model for Low-Resources Languages, CEUR Workshop Proceedings, 2023, 3403, P. 99–113.

16. Lyashenko, V., Rabotiahov, A., Kobylin, O., Kolesnykov, D. Analysis of Human Speech as a Protection Tool in Infocommunication Systems, International Scientific-Practical Conference on Problems of Infocommunications Science and Technology, PIC S and T 2018 - Proceedings, 2019, P. 79–83.

20. Smelyakov, K., Chupryna, A., Darahan, D., Midina, S. Effectiveness of modern text recognition solutions and tools for common data sources, CEUR Workshop Proceedings, 2021, 2870, P. 154–165

21. Yakovlev S., Khovrat, A., Kobziev, V. Using Parallelized Neural Networks to Detect Falsified Audio Information in Socially Oriented Systems, CEUR Workshop Proceedings, 2023, 3624, P. 220–238

ДОДАТОК Б

Апробація результатів роботи

УДК: 004.89 DOI: https://doi.org/10.30837/IYF.IIS.2024.428

ДОСЛІДЖЕННЯ МЕТОДІВ ТА МОДЕЛЕЙ АВТОМАТИЧНОГО РОЗПІЗНАВАННЯ УЧАСНИКІВ АУДІО РОЗМОВИ

Андрєєв І. Г.
Науковий керівник – к.т.н., доц. каф. ПІ Каук В. І.
Харківський національний університет радіоелектроніки, каф. ПІ,
м. Харків, Україна
e-mail: ivan.andrieiev@nure.ua

The object of the study is the process of automatic speech recognition (ASR) and one of the branch of it – speaker recognition or diarization (SD).

The purpose of the work is to study the stages of the process of speech and speakers recognition, the analysis of methods and models of machine learning and neural networks, as well as modern frameworks for training speech recognition models and diarization. Based on the received results of the study it would be selected the existing models, systems or products that would play a role of starting point in the creation of a custom ASR and SD models for enhancing those processes and bringing more values and benefits in different areas of human being.

У сучасному цифровому світі, де аудіо розмови і мовлення є невід'ємною частиною нашого повсякденного життя, проблема розпізнавання мовлення учасників розмови стає все більш актуальною і важливою. Ця проблема має великий потенціал застосування в різних сферах, включаючи телефонну комунікацію, відеоконференції, медіа і багато інших галузей.

Машинне навчання та нейронні мережі є ключовими компонентами сучасних процесів розпізнавання мовлення та визначення учасників розмови. Машинне навчання дозволяє моделі навчатися на основі великої кількості даних та робити прогнози, а нейронні мережі, зокрема глибокі нейронні мережі, є потужними інструментами для обробки аудіо даних та розпізнавання мовлення.

Покращення процесу розпізнавання учасників розмови та оптимізація використання цих технологій можуть дати позитивний вплив у таких напрямах:

– аутентифікація та безпека: питання безпеки стає особливо актуальним у контексті конфіденційності та захисту від несанкціонованого доступу;

– навчання: транскрипція запитань і відповідей студента для аналізу відповідей, наданих професором або студентами;

– здоров'я: окремі коментарі пацієнтів і лікарів як для особистих зустрічей, так і для телефонних консультацій;

Рисунок Б.1 – XXVIII Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті», Харків, 2024. Перша сторінка

– підтримка продажів: відстеження того, хто що сказав на зустрічі з продавцями, і навчання продавця, що говорити та коли мовчати;

– аналіз доповідача: відстежуйте поточні та попередні коментарі певного доповідача під час зустрічей або відстежуйте час розмови під час телефонної розмови, тощо.

Метою цього дослідження є аналіз методів та моделей автоматичного розпізнавання учасників аудіо розмови з використанням машинного навчання і нейронних мереж та встановленню шляхів покращення окремих етапів або всього процесу розпізнавання мовлення загалом. До основних підходів визначення мовлення виділені:

– підхід під наглядом (класифікація): модель навчена розпізнавати обмежену кількість промовців (тобто класів). Цей підхід менш гнучкий, але може бути дуже ефективним, особливо для великої кількості промовців (>5);

– підхід без нагляду (кластеризація): модель групує аудіо сегменти відповідно до промовця на основі виділених аудіо характеристик.

Об'єктом дослідження є процеси автоматичного розпізнавання мовлення (ASR) та розпізнавання учасників розмови (SD), а також існуючі системи та сервіси, що реалізують ці процеси. Найбільш популярні сервіси з розпізнавання мови та визначення учасників розмови базують навчання своїх моделей на основі згорткових (CNN) і рекурентних типів нейронних мережах (RNN).

Отримані результати цього дослідження стануть потенційно важливою складовою для розвитку технологій обробки аудіо даних та розпізнавання мовлення. Аналіз моделей та процесів, з одного боку, є важливою складовою на шляху до повного розуміння цієї галузі та її перспектив, а з іншого, є реальним шансом використати знання з області машинного навчання та нейронних мереж для вирішення реальних завдань сучасного світу аудіо відносин.

Список використаних джерел

1. Бридінський, В. Побудова системи ідентифікації мовців на основі бібліотеки аудіообробки PyAnnote. Information Technology: Computer Science, Software Engineering and Cyber Security. 2022. №2. С. 3-11.

2. Корнієнко О. Метод відображення мовних сигналів у задачі розпізнавання мовця. Технічні науки та технології. 2017. №3. С. 129-137.

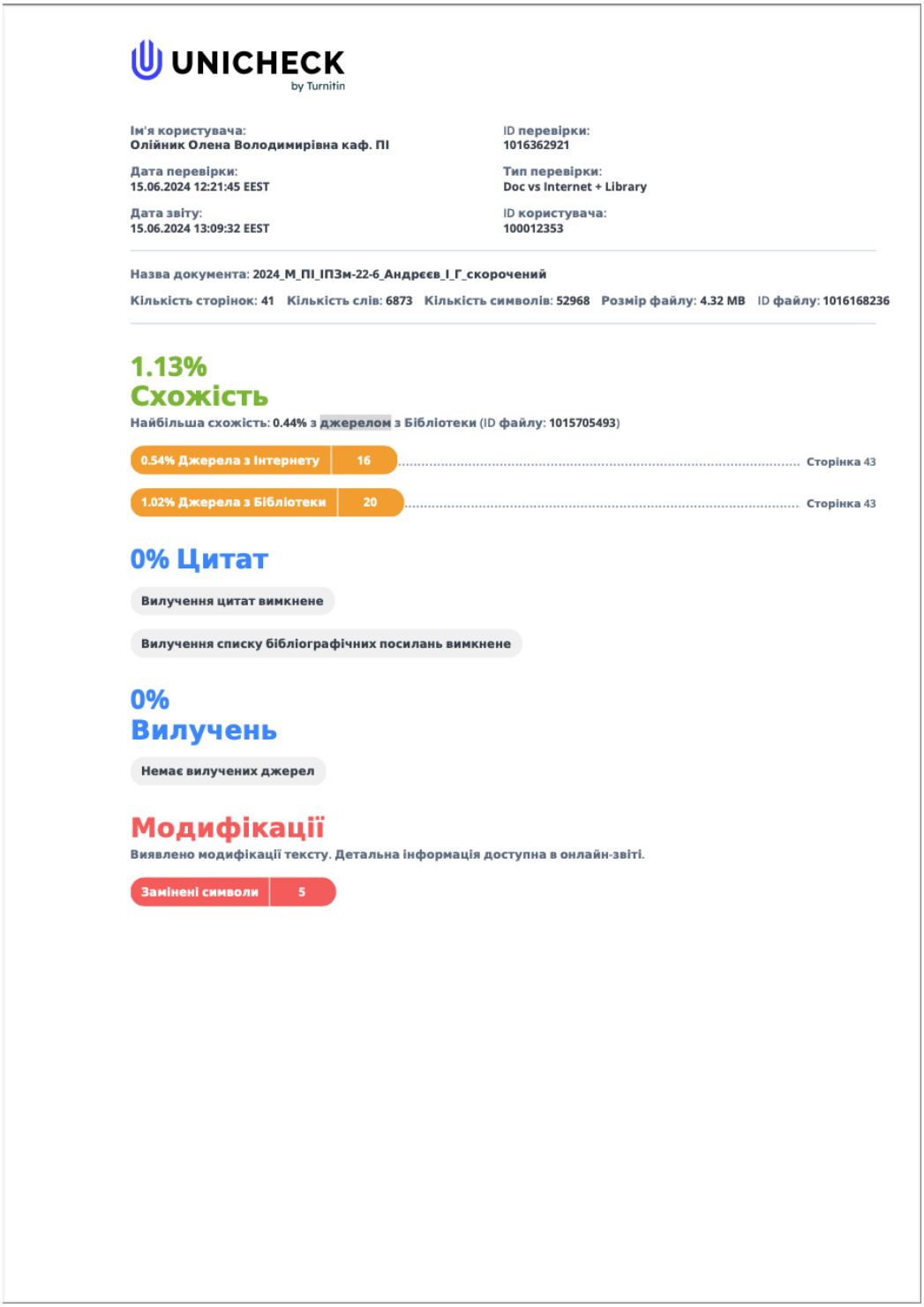
3. Bai Z., Zhang X.-L. Speaker recognition based on deep learning: An overview. Neural Networks. 2021. Vol. 140. P. 65-99.

4. Introduction to Speech Processing. Speaker Diarization. URL: https://speechprocessingbook.aalto.fi (дата звернення: 20.11.2023).

Рисунок Б.2 – XXVIII Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті», Харків, 2024. Друга сторінка

ДОДАТОК В

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



UNICHECK
by Turnitin

Ім'я користувача:
Олійник Олена Володимирівна каф. ПІ

ID перевірки:
1016362921

Дата перевірки:
15.06.2024 12:21:45 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
15.06.2024 13:09:32 EEST

ID користувача:
100012353

Назва документа: 2024_М_ПІ_ІПЗм-22-6_Андрєєв_І_Г_скорочений

Кількість сторінок: 41 Кількість слів: 6873 Кількість символів: 52968 Розмір файлу: 4.32 MB ID файлу: 1016168236

1.13%
Схожість

Найбільша схожість: 0.44% з джерелом з Бібліотеки (ID файлу: 1015705493)

0.54% Джерела з Інтернету 16 ..... Сторінка 43

1.02% Джерела з Бібліотеки 20 ..... Сторінка 43

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 5

Рисунок В.1 – Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ

ДОДАТОК Г

Слайди презентації

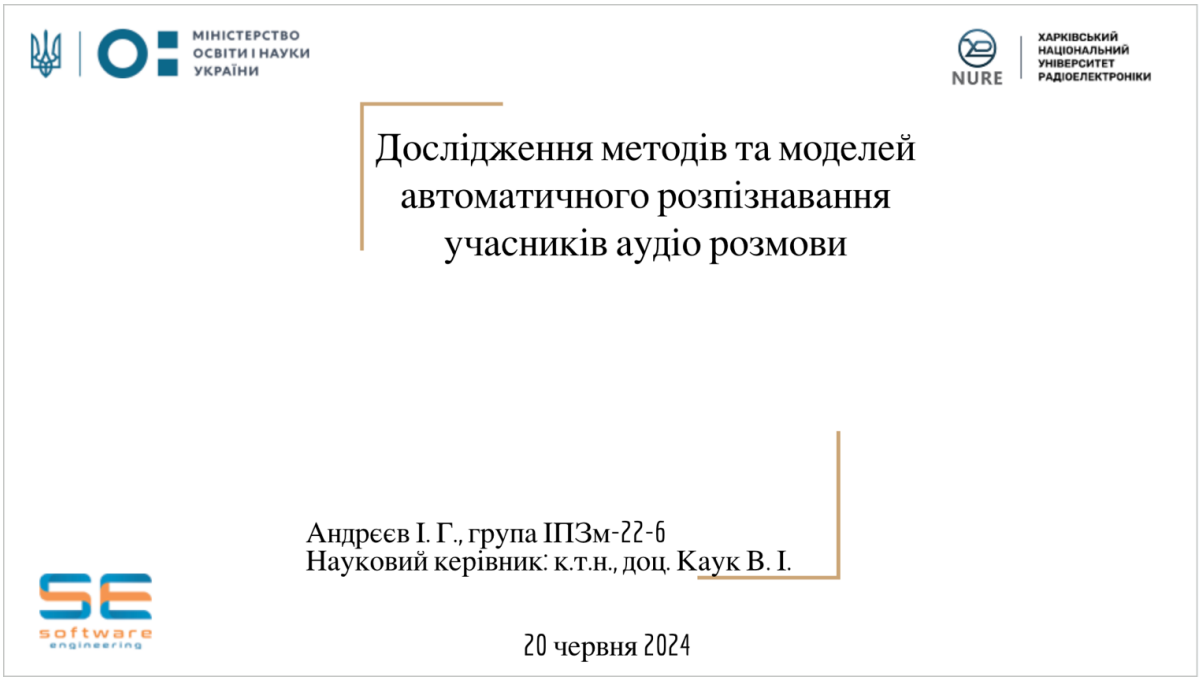


Рисунок Г.1 – Слайд №1 презентації

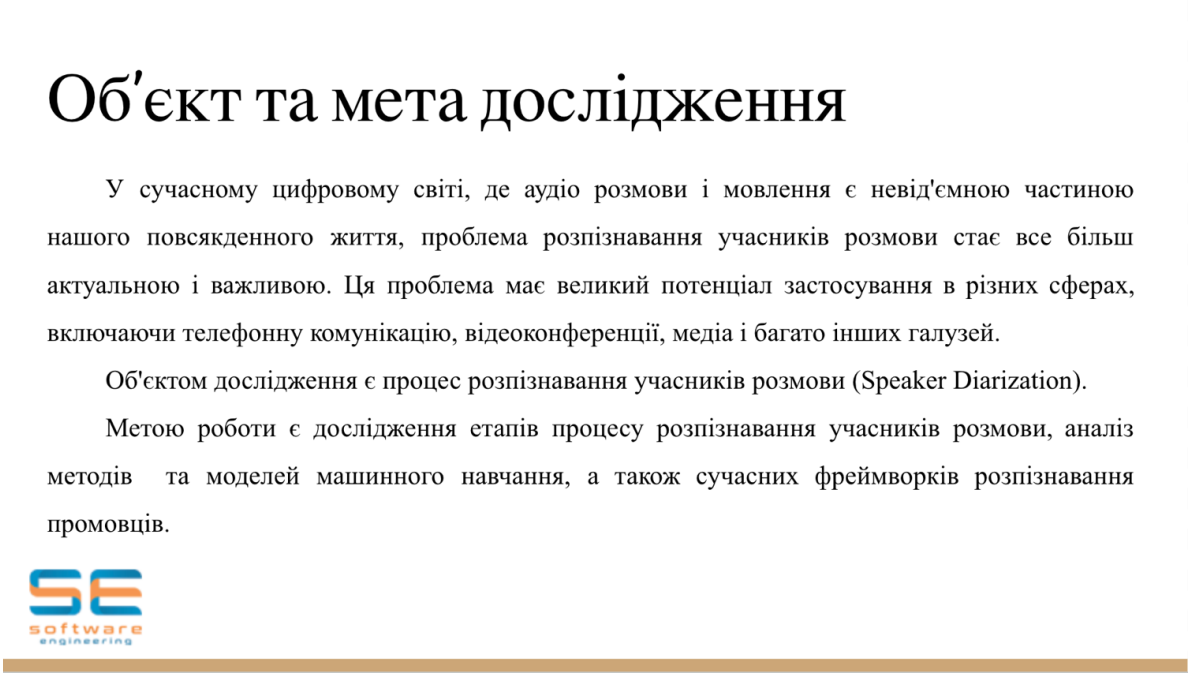


Рисунок Г.2 – Слайд №2 презентації

Огляд процесу розпізнавання мовців

У контексті розпізнавання мовця з підходом без нагляду, наразі є найбільш поширеним підходом, послідовність виконання включає кілька завдань для виявлення голосової активності та сегментації аудіо на мовні та немовні сегменти, а потім групування їх відповідно до промовця

Послідовність кроків розпізнавання промовців з використанням підходу без нагляду

Рисунок Г.3 – Слайд №3 презентації

Архітектура підходу до навчання

Сучасні популярні сервіси розпізнавання промовців базують навчання своїх моделей на основі RNN типі нейронної мережі. RNN використовують повторювані з'єднання, які дозволяють інформації зберігатися та перетікати від одного кроку до наступного. Це дозволяє їм фіксувати тимчасові залежності та послідовні шаблони в даних. RNN мають приховані стани, які зберігають пам'ять про минулі вхідні дані, що робить їх здатними навчатися з історичного контексту.

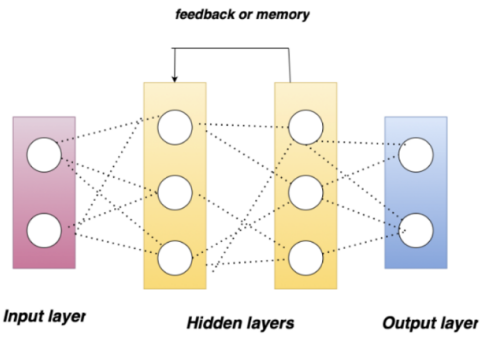


Рисунок Г.4 – Слайд №4 презентації

Огляд існуючих сервісів

Аспект	Amazon Transcribe	Speech Service MS Azure	Speech-to-Text Google Cloud	AssemblyAI	IBM Watson
Модифікація	Словник, пунктуація, ідентифікація мови	Використання власної моделі	Зміна голосу, адаптація мови	Аналіз настроїв, виявлення теми	Адаптація до специфічної лексики
Сфери використання	Контакт-центри, створення медіаконтенту	Різні напрями у підприємництві	Системи навігації, онлайн зустрічі	Віртуальні асистенти, чат боти	Бізнес-аналітика, маркетинг
Особливі властивості	Генеративне підсумовування дзвінків, рівень впевненості слів	Власні нейронні моделі голосу, переклад у реальному часі	Маркування аудіо даних	Використання аудіо інтелекту	Виявлення ключових слів, визначення емоцій
Легкість використання	Мінімальні зміни для існуючих клієнтів	Простота використання для розробників	Спрощена обробка та аналіз даних	Проста інтеграція з API	Зручний інтерфейс та інтеграція

Рисунок Г.5 – Слайд №5 презентації

Схема потенційної системи

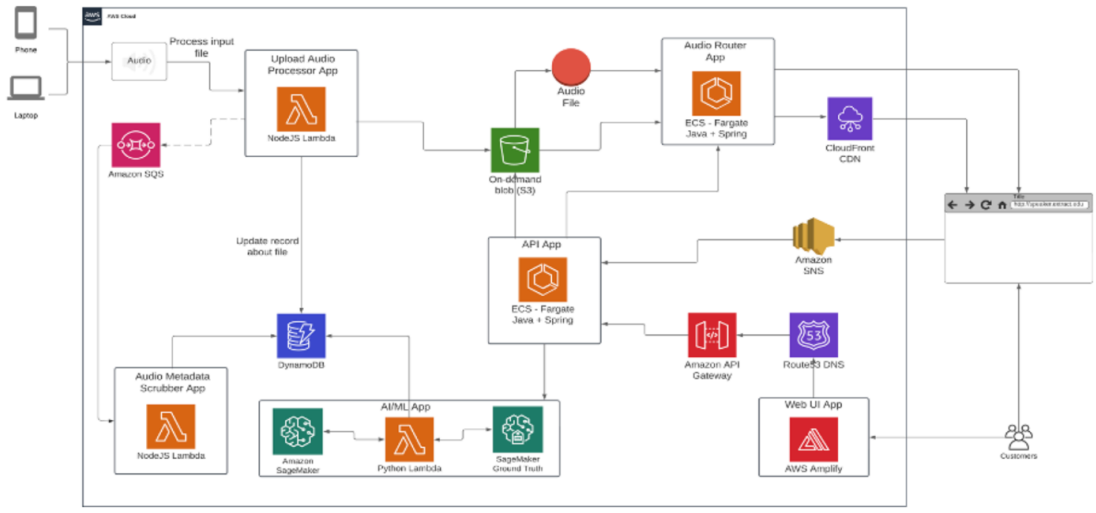


Рисунок Г.6 – Слайд №6 презентації

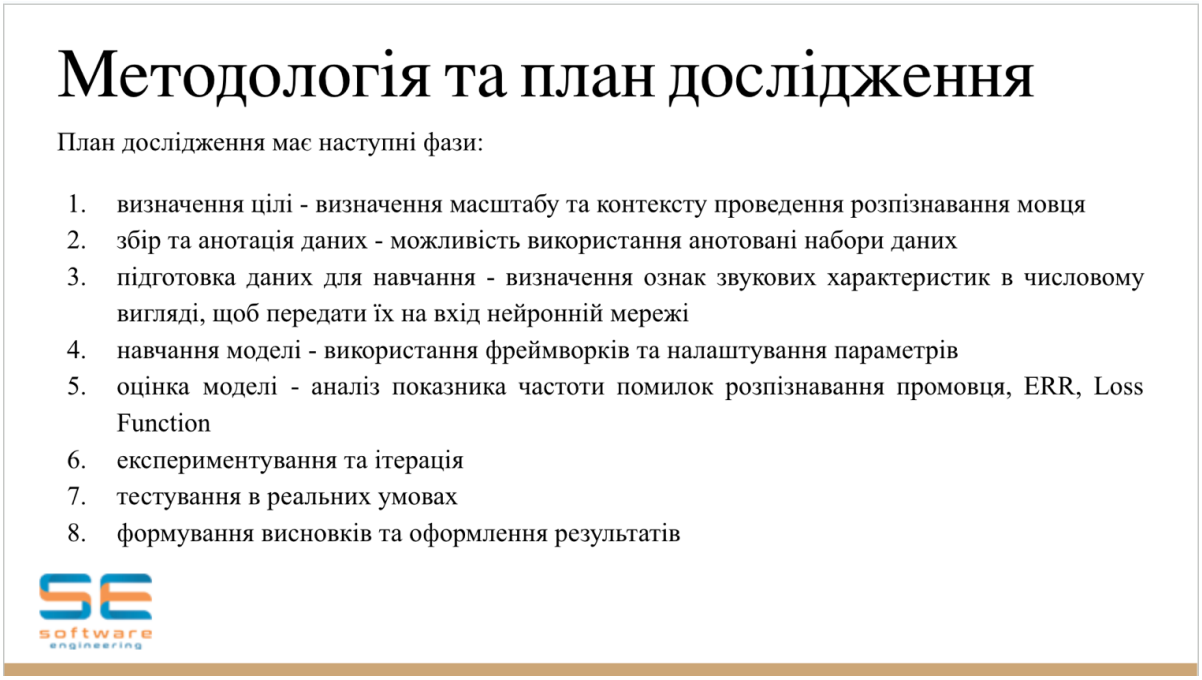


Рисунок Г.7 – Слайд №7 презентації

Рисунок Г.8 – Слайд №8 презентації

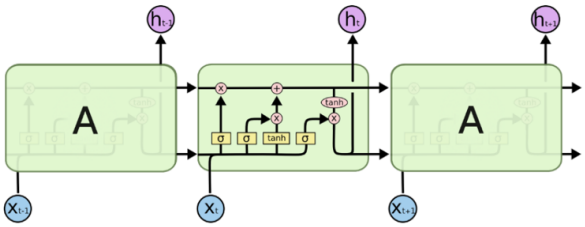
Методи аугментація аудіо записів

- додавання шуму;
- зміна швидкості відтворення;
- зміна висоти тону;
- реверберація;
- часова зміна;
- комбінація методів.

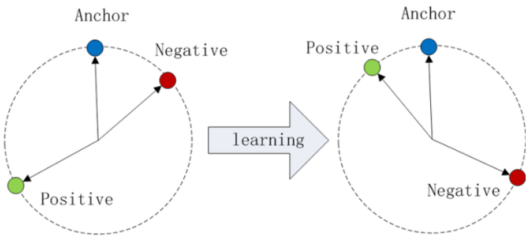
```python
import numpy as np
import librosa
import soundfile as sf
import os

# Завантаження аудіофайлу
directory_path = os.path.join(os.path.expanduser("~"), 'Documents/Nure/Mag_robota/DataForProcessing')
file_path = directory_path + '/Conference.wav'
y, sr = librosa.load(file_path, sr=None)

# Додавання білого шуму
noise = np.random.randn(len(y))
y_noisy = y + 0.005 * noise

# Зміна швидкості відтворення
y_fast = librosa.effects.time_stretch(y, rate=1.25)
y_slow = librosa.effects.time_stretch(y, rate=0.75)

# Зміна висоти тону
y_pitch_up = librosa.effects.pitch_shift(y, sr=sr, n_steps=4)
y_pitch_down = librosa.effects.pitch_shift(y, sr=sr, n_steps=-4)

# Збереження результатів
sf.write(directory_path + '/Conference_noisy.wav', y_noisy, sr)
sf.write(directory_path + '/Conference_fast.wav', y_fast, sr)
sf.write(directory_path + '/Conference_slow.wav', y_slow, sr)
sf.write(directory_path + '/Conference_pitch_up.wav', y_pitch_up, sr)
sf.write(directory_path + '/Conference_pitch_down.wav', y_pitch_down, sr)
```

Рисунок Г.9 – Слайд №9 презентації

Машинне навчання у програмній реалізації



Повторюваний LSTM модуль



Триплетна втрата з косинусною подібністю

Рисунок Г.10 – Слайд №10 презентації

Виконання процесу розпізнавання мовця

Приклад виконання кроків з розпізнавання промовця з використанням Resemblyzer

```python
encoder = VoiceEncoder()
_, cont_embeds, wav_splits = encoder.embed_utterance(wav, return_partials=True, rate=16)

print(f'cont_embeds = ${cont_embeds}')

clusterer = SpectralClusterer(
    min_clusters=3,
    max_clusters=6)

labels = clusterer.predict(cont_embeds)
print(f'labels = ${labels}')

labelling = create_labelling(labels, wav_splits)
labeled_speakers = build_speakers(labelling)

print(f'labelling = ${labeled_speakers}')

speaker_embeds = [encoder.embed_utterance(speaker_wav) for speaker_wav in speaker_wavs]
similarity_dict = {name: cont_embeds @ speaker_embed for name, speaker_embed in
                   zip(speaker_names, speaker_embeds)}

## Run the interactive demo
interactive_diarization(similarity_dict, wav, wav_splits)
```

Рисунок Г.11 – Слайд №11 презентації

Результат розпізнавання мовця

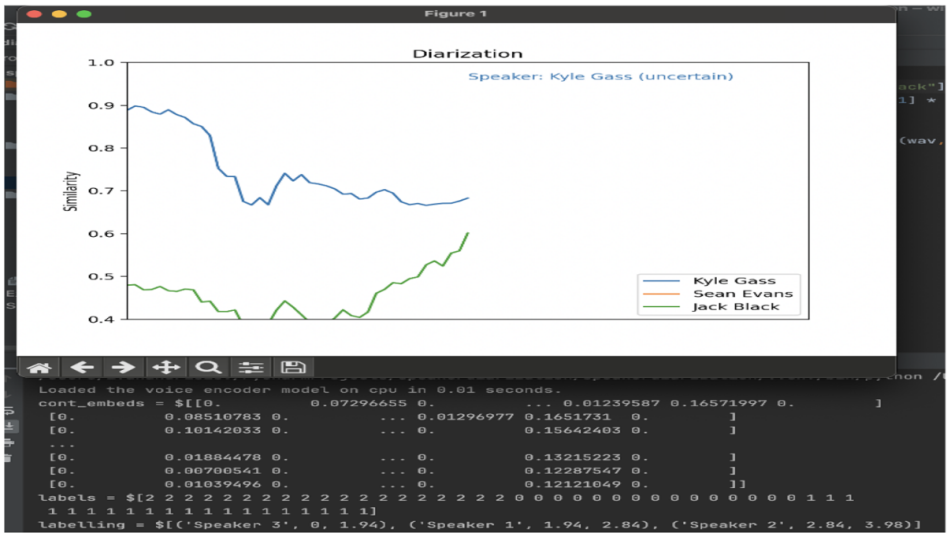
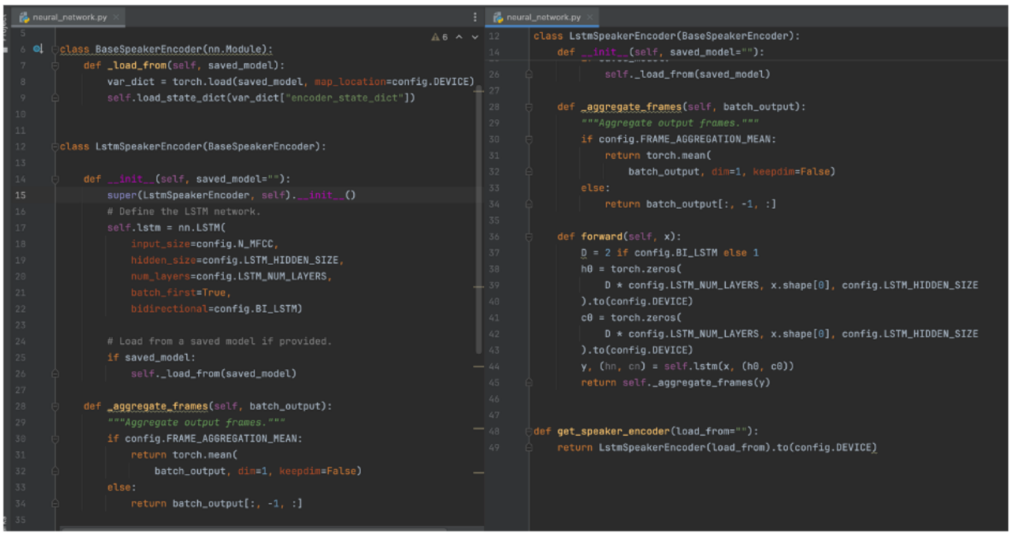


Рисунок Г.12 – Слайд №12 презентації

Приклад коду побудови власної моделі



```python
class BaseSpeakerEncoder(nn.Module):
    def _load_from(self, saved_model):
        var_dict = torch.load(saved_model, map_location=config.DEVICE)
        self.load_state_dict(var_dict["encoder_state_dict"])


class LstmSpeakerEncoder(BaseSpeakerEncoder):

    def __init__(self, saved_model=""):
        super(LstmSpeakerEncoder, self).__init__()
        # Define the LSTM network.
        self.lstm = nn.LSTM(
            input_size=config.N_MFCC,
            hidden_size=config.LSTM_HIDDEN_SIZE,
            num_layers=config.LSTM_NUM_LAYERS,
            batch_first=True,
            bidirectional=config.BI_LSTM)

        # Load from a saved model if provided.
        if saved_model:
            self._load_from(saved_model)

    def _aggregate_frames(self, batch_output):
        """Aggregate output frames."""
        if config.FRAME_AGGREGATION_MEAN:
            return torch.mean(
                batch_output, dim=1, keepdim=False)
        else:
            return batch_output[:, -1, :]
```

```python
class LstmSpeakerEncoder(BaseSpeakerEncoder):
    def __init__(self, saved_model=""):
            self._load_from(saved_model)

    def _aggregate_frames(self, batch_output):
        """Aggregate output frames."""
        if config.FRAME_AGGREGATION_MEAN:
            return torch.mean(
                batch_output, dim=1, keepdim=False)
        else:
            return batch_output[:, -1, :]

    def forward(self, x):
        D = 2 if config.BI_LSTM else 1
        h0 = torch.zeros(
            D * config.LSTM_NUM_LAYERS, x.shape[0], config.LSTM_HIDDEN_SIZE
        ).to(config.DEVICE)
        c0 = torch.zeros(
            D * config.LSTM_NUM_LAYERS, x.shape[0], config.LSTM_HIDDEN_SIZE
        ).to(config.DEVICE)
        y, (hn, cn) = self.lstm(x, (h0, c0))
        return self._aggregate_frames(y)


def get_speaker_encoder(load_from=""):
    return LstmSpeakerEncoder(load_from).to(config.DEVICE)
```

Рисунок Г.13 – Слайд №13 презентації

Результат навчання власної моделі

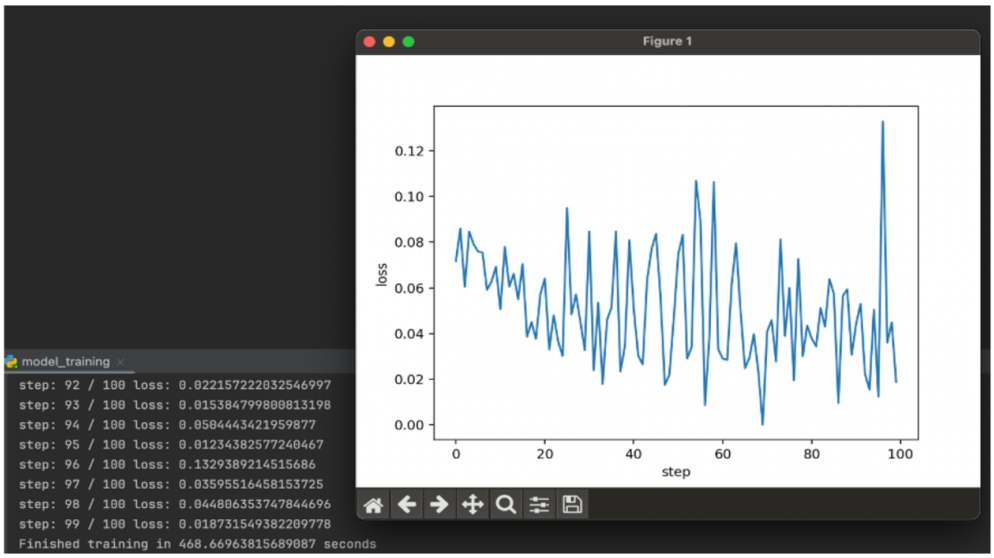


Рисунок Г.14 – Слайд №14 презентації

Результат тестування власної моделі

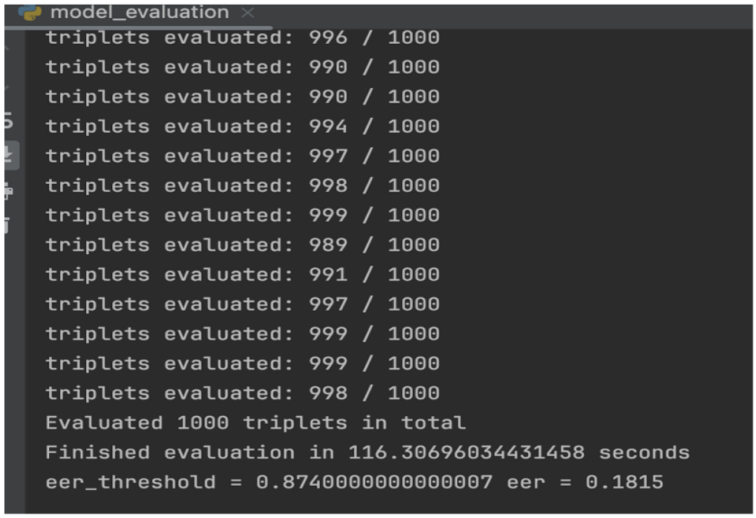


Рисунок Г.15 – Слайд №15 презентації

Результати експерименту, частина А

Кількість шарів / Кількість кроків	16	32	64	128	256
100	0.0247	0.0194	0.0187	0.0162	0.0151
500	0.0193	0.0173	0.0156	0.0148	0.0134
1000	0.0176	0.0159	0.0137	0.0122	0.0113
2000	0.0162	0.0144	0.0121	0.0105	0.0093
3500	0.0147	0.0128	0.0109	0.0097	0.0079
5000	0.0133	0.0116	0.0101	0.0088	0.0065

Рисунок Г.16 – Слайд №16 презентації

Результати експерименту, частина Б

Кількість шарів / Кількість кроків	16	32	64	128	256
100	0.0467	0.0414	0.0398	0.0365	0.0341
500	0.0432	0.0403	0.0387	0.0332	0.0315
1000	0.0416	0.0362	0.0331	0.0304	0.0293
2000	0.0401	0.0352	0.0329	0.0285	0.0261
3500	0.0387	0.0341	0.0305	0.0259	0.0221
5000	0.0323	0.0276	0.0235	0.0202	0.0179

Рисунок Г.17 – Слайд №17 презентації

Результати експерименту, частина В

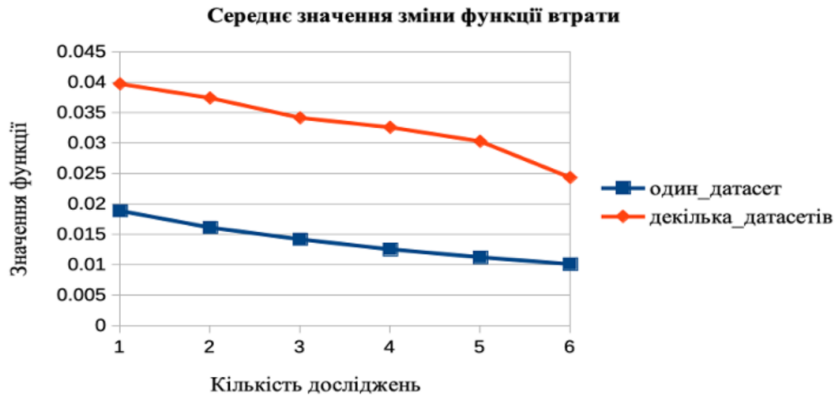


Рисунок Г.18 – Слайд №18 презентації

Результати експерименту, частина Г

Кількість триплетів	Результат ERR	
	З одним набором даних	З декількома наборами даних
50	0.3416	0.5256
100	0.2932	0.4137
500	0.2021	0.3328
1000	0.1815	0.2316
2000	0.1398	0.1412
3000	0.0743	0.0998

Рисунок Г.19 – Слайд №19 презентації

Аналіз отриманих результатів

Під час проведення дослідження було проведено порівняльну характеристику результатів роботи існуючих моделей на базі Reseblyzer та Pyannote audio. Існуючі рішення продемонстрували доволі точні результати та дали поштовх у напрямку побудови власної моделі.

Аналізуючи отримані значення власного навчання видно, що процес навчання може бути оптимізованим за рахунок модифікації параметрів обробки аудіо файлів, формування та трансформації проміжних даних, таких я к триплетів, а також через розширення набору даних через застосування різних тренувальних або перевірочних сетів.

Треба виділити, що ефективність та швидкість роботи з навчанням нейронної мережі також залежить від фізичних особливостей пристрою та оптимізації з мультипроцесорною обробкою команд, що не є критичним для невеликих наборів даних чи спрощеною послідовністю кроків, але може відігравати критичну роль у порівняно комплексних запитах та налаштуваннях.

Рисунок Г.20 – Слайд №20 презентації

Підсумки

Незважаючи на вже існуючі результати у цьому напрямку, є ще багато можливостей для подальшого вдосконалення системи розпізнавання учасників розмови. До них можна віднести використання більш великих та різноманітних даних для навчання та перевірки моделі, розширення функціональності за рахунок оптимізації трансформерів характеристик промовців, для цього можуть бути використанні більш прогресивні ресурси PyTorch чи альтернативний поставник бібліотек, такий як TensorFlow від Google або SageMaker від AWS. Також вибір більш ефективних параметрів для навчання, а саме конфігурація прихованих проміжних шарів або кількості кроків з обробки аудіо проміжків чи вже опрацьованих характеристик або експериментування з можливими налаштуваннями фізичних аспектів машини, таких як використання багатопроцесорного виконання програми або переходом між CPU та GPU процесорами.

Рисунок Г.21 – Слайд №21 презентації

ДОДАТОК Д

Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008: 2015

1

Експертний висновок результатів перевірки кваліфікаційної роботи

студент (посада) | програмної інженерії (кафедра) | ІПЗм-22-6 (група)

Андрєєв Іван Георгійович

(прізвище, ім'я, по батькові)

Зауваження

Пункт ДСТУ 3008-2015	Зміст пункту	Сторінка кваліфікаційної роботи
1	2	3
	7.1 Загальні положення	
	7.3 Нумерація сторінок звіту	
	7.4 Нумерація розділів, підрозділів, пунктів, підпунктів	
	7.5 Рисунки	
	7.6 Таблиці	
	7.7 Переліки	
	7.8 Примітки	
	7.9 Виноски	
	7.10 Формули та рівняння	
	7.11 Посилання	
	7.13 Список авторів	
	7.14 Скорочення та умовні позначки	
	7.15 Додатки	

зауважень немає

Експерт _____ (підпис) Олена ОЛІЙНИК (прізвище, ініціали)

16.06.2024

Рисунок Д.1 – Експертний висновок результатів перевірки кваліфікаційної роботи