МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМЕНІ С.З. ҐЖИЦЬКОГО

ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

першого (бакалаврського) рівня вищої освіти

на тему: «**Особливості створення системи, що використовує глибоке навчання для розпізнавання слів або фраз з аудіозаписів**»

Виконав: студент групи КН-41
Спеціальності 122–«Комп'ютерні науки»
(шифр і назва)

Левицький Олександр Любомирович
(Прізвище та ініціали)

Керівник: в.о доцент, к.ф. – м.н.Чухрай Л.В.
(Прізвище та ініціали)

Рецензенти: _____
(Прізвище та ініціали)

ДУБЛЯНИ-2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ВЕТЕРИНАРНОЇ МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ
ІМЕНІ С.З. ҐЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Перший (бакалаврський) рівень вищої освіти

Спеціальність 122 «Комп'ютерні науки»

"ЗАТВЕРДЖУЮ"
Завідувач кафедри_____
д.т.н., проф, Тригуба А.М.
"_____"_____2025 р.

ЗАВДАННЯ

на кваліфікаційну роботу

Левицькому Олександру Любомировичу

1. Тема роботи: *«*Особливості створення системи, що використовує глибоке навчання для розпізнавання слів або фраз з аудіозаписів*»*
Керівник роботи: в.о доцент, к.ф. – м.н.Чухрай Л.В._____
Затверджені наказом по університету 123/к-с від 25.02.2025

2. Строк подання студентом роботи 16.06.2025 р.

3. Вихідні дані до роботи: *аудіофайли з голосовими командами та їхні текстові транскрипції,бібліотеки глибокого навчання (TensorFlow, PyTorch), засоби обробки аудіо (librosa, pyaudio), інтерфейс користувача на базі Tkinter, зовнішні API для розпізнавання мови (Google Speech-to-Text), обчислювальне середовище з підтримкою GPU, методи попередньої обробки та візуалізації аудіосигналів..*

4. Зміст розрахунково-пояснювальної записки: (перелік питань, які потрібно розробити)

1. Аналіз предметної області
2. Постановка задачі
3. Проектування та реалізація системи розпізнавання аудофайлів на текст
4. Охорона праці та безпека в надзвичайних ситуаціях
Вимоги та пропозиції
Бібліографічний список
Додатки

5. Перелік графічного матеріалу: *Графічний матеріал подається у вигляді презентації*
6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3	в.о доцент, к.ф. – м.н.Чухрай Л.В.		
4	*Городецькии І.М., доцент кафедри фізики, Інженерної механіки та безпеки виробництва*		

7.Дата видачі завдання 25.02.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Етапи виконання кваліфікаційної роботи	Строк виконання етапів роботи	Приміт-ка
1.	Написання першого розділу та означення головних завдань роботи	*28.02.2025 – 03.03.2025*	
2.	Виконання другого розділу, опис інструментів та формування головних показників для розробки	*04.03.2025 – 01.04.2025*	
3.	Виконання третього розділу, реалізація мети роботи	*01.04.2025 – 04.05.2025*	
4.	Виконання четвертого розділу «Охорона праці та безпека в надзвичайних ситуаціях»	*06.04.2025 – 08.05.2025*	
5.	Завершення оформлення розрахунково-пояснювальної записки та презентаційного матеріалу	*01.05.2025 – 01.06.2025*	
6.	Завершення роботи в цілому	*02.06.2025 – 09.06.2025*	

Студент_____ Левицький О.Л
(підпис)

Керівник роботи _____ Чухрай Л.В
(підпис)

УДК: 004.912:004.738.5(477.83)

Кваліфікаційна робота: 54 сторінки текстової частини, 11 рисунків, 5 таблиці, 21 джерело літератури, 1 додаток.

«Особливості створення системи, що використовує глибоке навчання для розпізнавання слів або фраз з аудіозаписів» Левицький О.Л. Кафедра інформаційних технологій. - Дубляни, Львівський НУВМБ, 2025.

У роботі розглянуто процеси побудови системи розпізнавання мовлення з використанням сервісу Google Speech-to-Text API, розроблено алгоритми обробки аудіосигналу, здійснено аналіз шумового середовища та особливостей української мови при розпізнаванні.

Показано етапи реалізації графічного інтерфейсу користувача (на базі Tkinter), створення скриптів для захоплення голосу, збереження результатів у локальних файлах та автоматичне завантаження транскриптів у Google Drive. Представлено діаграму послідовності обробки аудіофайлів та результати тестування.

Реалізовано функції асинхронного запису та багатопотокової взаємодії з API сервісами для забезпечення стабільної роботи інтерфейсу. Сформовано рекомендації щодо безпечної роботи з обліковими даними та конфігурацією (через.env).

Надано приклади форматів вхідних/вихідних файлів, обґрунтовано вибір технологій та засобів реалізації системи.

Ключові слова: розпізнавання мовлення, аудіоаналіз, хмарні сервіси, Google Speech API, Google Drive, Python, Tkinter
Keywords: speech recognition, audio analysis, cloud services, Google Speech API, Google Drive, Python, Tkinter

ЗМІСТ

ВСТУП

У сучасних умовах розвитку інформаційних технологій особливу увагу привертають системи, здатні забезпечити природну взаємодію людини з комп’ютером. Однією з таких технологій, без перебільшення, є розпізнавання мовлення — процес, що полягає в автоматичному перетворенні усного мовлення у текстову форму. Цей напрямок, безсумнівно, відіграє ключову роль у формуванні нових підходів до створення інтерфейсів користувача, систем автоматичного керування та засобів підтримки осіб з обмеженими можливостями.

У даному проєкті, передусім, розглядається типологія систем розпізнавання мовлення, яка, з одного боку, базується на характері вимови (ізольовані, зв’язані, безперервні та спонтанні висловлювання), а з іншого — на особливостях мовців (дикторозалежні та дикторонезалежні моделі). Крім того, суттєве значення має обсяг словника, що використовується системою: від мінімальних наборів команд до великих корпусів мови. Усе це безпосередньо впливає на точність, ефективність і гнучкість розроблюваних рішень.

З огляду на вищевикладене, даний проєкт має на меті надати комплексне уявлення про сучасні підходи до розпізнавання мовлення, їх практичну реалізацію, класифікацію та сфери застосування. Окрему увагу буде приділено специфіці розпізнавання українського мовлення, що, на нашу думку, є вкрай актуальним у контексті зростаючої потреби в національно орієнтованих IT-продуктах.

Отже, дана робота має на меті не лише узагальнити теоретичні основи відповідної технології, але й закласти підґрунтя для подальшого дослідження та практичного впровадження систем розпізнавання мовлення українською мовою.

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Попередня обробка аудіосигналів для систем розпізнавання мовлення

Попередня обробка аудіосигналів для систем розпізнавання мовлення Попередня обробка аудіосигналів є одним із ключових етапів у системах автоматичного розпізнавання мовлення (ASR). Вона значно впливає на якість та точність розпізнавання, оскільки саме від якості вхідного сигналу залежить коректність подальшої обробки, виділення характеристик і класифікації звуків. Без ретельної попередньої обробки навіть найсучасніші алгоритми розпізнавання можуть дати невисокі результати через наявність шумів, спотворень або нестабільної інтенсивності сигналу. [1].

Основні завдання попередньої обробки Попередня обробка спрямована на поліпшення якості аудіосигналу, усунення або зменшення впливу шумів, нормалізацію рівня гучності, розбиття сигналу на логічні сегменти — фонеми або слова, а також підготовку даних для виділення ключових ознак, що є важливими для алгоритмів розпізнавання. Основні етапи попередньої обробки включають:

1)Фільтрація шумів та артефактів

2)Нормалізація амплітуди та рівня гучності

3)Сегментація сигналу

4)Фільтрація шумів та артефактів

На рис. 1.1 наведено процес розпізнавання мовлення для загального розуміння та для отримання загальної картини

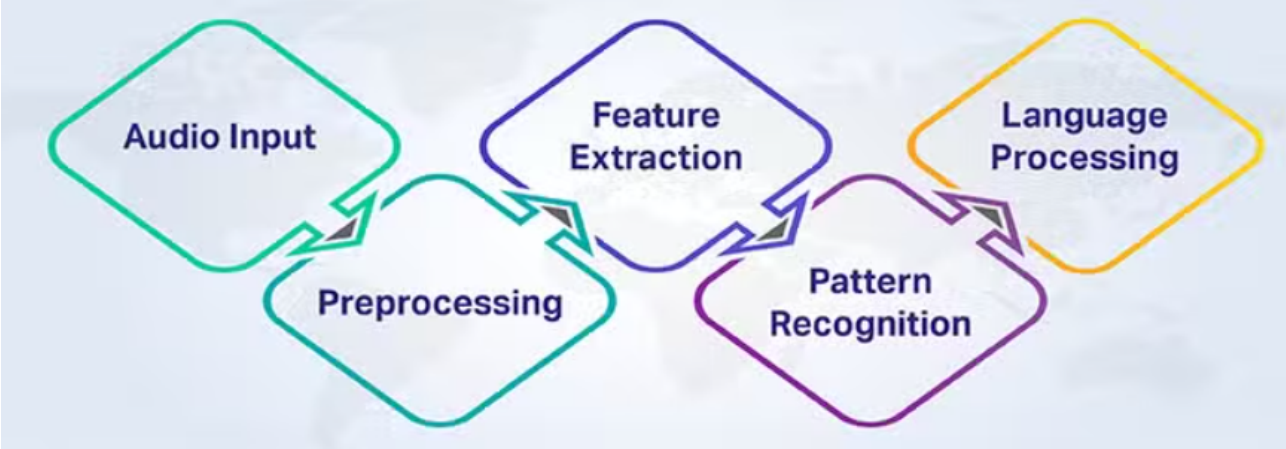

Рисунок 1.1 – Процес розпізнавання мовлення [2]

Однією з найважливіших задач є видалення фонових шумів, які можуть бути викликані як навколишнім середовищем (шум вулиці, технічні звуки, голоси інших людей), так і несправностями апаратури. Без цього етапу шум може «замилювати» корисний мовний сигнал, що призводить до помилок у розпізнаванні. [3].

Для цього застосовують різні види фільтрів:

1)Низькочастотні фільтри (Low-Pass Filters) видаляють високочастотні шуми, що виникають, наприклад, від електромагнітних завад або цифрових артефактів. Вони пропускають частоти нижче певного порогу, пригнічуючи все вище.

2)Високочастотні фільтри (High-Pass Filters) навпаки відсікають низькочастотні шуми, наприклад, гул мережі або шум від механічних вібрацій.

3)Смугові фільтри (Band-Pass Filters) дозволяють виділити корисний спектр мовного сигналу, зазвичай у діапазоні приблизно 300–3400 Гц, що охоплює основні частоти людської мови, одночасно пригнічуючи частоти поза цим діапазоном.

4)Фільтр Калмана — адаптивний фільтр, який враховує статистичні характеристики шуму і сигналу, прогнозує майбутні значення сигналу і коригує виміри, що дозволяє ефективно зменшувати випадкові шуми та коливання. Особливо корисний при роботі з динамічними сигналами в реальному часі.

Адаптивні фільтри шуму — алгоритми, які динамічно підлаштовуються під поточний рівень і тип шуму, наприклад, LMS (Least Mean Squares) фільтри.

В результаті застосування фільтрації сигнал стає більш чистим, без сторонніх звуків, що істотно підвищує ефективність розпізнавання [12].

Нормалізація амплітуди та рівня гучності. Після фільтрації наступним важливим кроком є нормалізація амплітуди аудіосигналу — приведення його рівня гучності до певного стандартного діапазону. Це необхідно через те, що записи мовлення можуть сильно відрізнятися за гучністю: наприклад, користувач може говорити тихо або дуже голосно, мікрофон може бути налаштований по-різному.

Основні методи нормалізації:

1) Пікова нормалізація (Peak Normalization): амплітуда сигналу масштабується так, щоб максимальне значення дорівнювало встановленому рівню (наприклад, 0 dB). Це дозволяє уникнути «зрізання» (clipping) і оптимізує використання динамічного діапазону.

2) Нормалізація середньоквадратичного значення (RMS Normalization): приводить до однакового середнього енергетичного рівня сигналу, що забезпечує більш збалансовану гучність для прослуховування і обробки.

3) Гістограмна вирівнювання (Histogram Equalization): застосовується рідше, але допомагає вирівняти розподіл амплітудних значень у складних випадках.

Ці методи дозволяють забезпечити стабільну якість сигналу, що надходить на наступні етапи обробки.

Сегментація — це розбиття безперервного аудіосигналу на смислові або акустичні одиниці, наприклад, фонеми, склади або слова. Це допомагає алгоритмам розпізнавання працювати більш ефективно, фокусуючись на окремих частинах мовлення.

Сегментація може бути:

1) За амплітудними порогами: визначення початку та кінця мовної частини по зміні інтенсивності сигналу.

2)За спектральними ознаками: виявлення меж фонем або слів за допомогою змін спектральних характеристик.

За допомогою моделей машинного навчання: наприклад, використання рекурентних нейронних мереж (RNN) або глибоких моделей для більш точного визначення сегментів.

Ефективна сегментація знижує помилки розпізнавання, особливо при присутності пауз, шумів або швидкому темпі мовлення.

1.2 Формати файлів текстових документів

Формати аудіофайлів та їх вплив на якість розпізнавання

Формати аудіофайлів відіграють ключову роль у тому, наскільки точно система зможе розпізнати мовлення. Адже те, як саме зберігаються й передаються звукові дані, безпосередньо впливає на якість аудіо. Чим краща якість, тим вища ймовірність коректного розпізнавання. І навпаки — якщо формат сильно стискає звук або втрачає важливі деталі, алгоритми розпізнавання можуть помилятися.

WAV (Waveform Audio File Format)

WAV — це класичний формат, який зберігає звук у вигляді сирих даних без будь-якого стиснення. Уявіть собі аудіофайл як необроблену звукову хвилю, з усіма нюансами і деталями.

Переваги: Завдяки відсутності стиснення, WAV-файли мають дуже високу якість, що робить їх ідеальними для завдань, де потрібна точність, наприклад, для професійного розпізнавання мовлення.

Недоліки: Однак такі файли займають багато місця, що не завжди зручно для зберігання або передачі.

MP3 (MPEG-1 Audio Layer III)

MP3 — найпопулярніший формат аудіо з стисненням з втратами. Він створений для того, щоб зменшити розмір файлу, видаляючи ті частоти, які людині важко почути.

Через видалення деяких звукових деталей точність розпізнавання може знижуватись, особливо якщо звук містить шум або складні мовні конструкції.

FLAC (Free Lossless Audio Codec)

FLAC — це компроміс між якістю і розміром файлу. Він стискає звук без втрати якості, зберігаючи всі оригінальні деталі. Для розпізнавання FLAC — це як мати WAV, але з меншим об'ємом файлу, що полегшує зберігання та передачу, при цьому не жертвуючи якістю.

AAC (Advanced Audio Coding)

AAC часто вважають наступником MP3, оскільки при тому ж бітрейті він забезпечує кращу якість звуку.Якщо аудіо записане з високим бітрейтом, розпізнавання з AAC може бути навіть кращим, ніж з MP3, але все ж з деякими обмеженнями через стиснення.

OGG (Ogg Vorbis)

OGG — це відкритий формат зі стисненням з втратами, який часто використовується в іграх і потоковому аудіо.

Переваги: Він забезпечує якість, порівнянну з MP3 або AAC, але при меншому розмірі файлу.

Вплив на розпізнавання: Особливо при високих бітрейтах, OGG може стати хорошим вибором для систем розпізнавання мовлення.

AIFF (Audio Interchange File Format)

AIFF — це формат, розроблений Apple, який дуже схожий на WAV, оскільки теж зберігає звук без стиснення.

Нижче наведено табл. 1.1, а саме таблицю порівняння форматів аудіофайлів [4].

Таблиця 1.1 – Формати аудіофайлів [4]

Формат	**Опис**	**Стиснення**	**Вплив на якість розпізнавання**
WAV	Нестиснений формат, зберігає звукові дані у вигляді хвиль	Без стиснення	Найкраща якість для розпізнавання мовлення через відсутність втрат якості
MP3	Популярний формат зі стисненням з втратами	З втратами	Прийнятна якість при високих бітрейтах, можлива втрата деталей звуку
FLAC	Формат зі стисненням без втрат	Без втрат	Висока якість звуку, хороший вибір для розпізнавання мовлення
AAC	Формат зі стисненням з втратами, краща якість порівняно з MP3	З втратами	Хороша якість при високих бітрейтах, можлива втрата деталей звуку
OGG	Вільний формат зі стисненням з втратами	З втратами	Хороша якість при високих бітрейтах, можлива втрата деталей звуку
AIFF	Нестиснений формат, подібний до WAV, розроблений Apple	Без стиснення	Висока якість звуку, хороший вибір для розпізнавання мовлення

Таким чином,можна побачити різницю між форматами аудіофайлів.

1.3 Огляд існуючих speech-to-text моделей та сервісів

Розпізнавання мовлення — це технологія, яка дозволяє комп’ютерам та іншим електронним пристроям аналізувати усне мовлення у вигляді послідовності акустичних сигналів. Після обробки аудіо формується текстовий вивід, який відображає зміст сказаного користувачем. Інакше кажучи, система трансформує мовлення у письмову форму, враховуючи контекст, зміст та комунікативний намір мовця.

Функціонування таких систем базується на розподілі мовного сигналу на невеликі звукові одиниці — фонеми. Оскільки усне мовлення може містити варіанти вимови, акценти, діалектизми або звукову редукцію, розпізнавання правильного написання слів є досить складним завданням.

Для подолання цих викликів використовуються методи штучного інтелекту та нейролінгвістичної обробки мови (Natural Language Processing, NLP). Вони дозволяють враховувати контекст та передбачати найбільш ймовірні слова, підвищуючи точність транскрипції.

Основні компоненти систем розпізнавання мовлення

Системи автоматичного розпізнавання мовлення зазвичай складаються з таких основних модулів:

1) Акустична модель — аналізує вхідний аудіосигнал та ідентифікує окремі фонеми;

2) Мовна модель — прогнозує послідовність слів, забезпечуючи граматичну та контекстуальну відповідність;

3) Словник вимови — містить фонетичні транскрипції слів, забезпечуючи відповідність між звуковими та текстовими формами;

4) Декодер — інтегрує інформацію з усіх попередніх модулів для формування остаточного тексту, обираючи найбільш ймовірну послідовність слів. [1].

На рис. 1.2 наведено схему процесу розпізнавання мовлення

Рисунок 1.2 – Схема процесу розпізнавання мовлення

Застосування розпізнавання мовлення. Технологія розпізнавання мовлення широко використовується в бізнесі. Вона спрощує ведення нотаток під час зустрічей, допомагає створювати документацію на основі голосових записів, а також застосовується у клієнтському обслуговуванні — наприклад, у системах інтерактивного голосового реагування (IVR) чи віртуальних помічниках, які обробляють запити клієнтів.

У сфері продажів розпізнавання мовлення допомагає аналізувати діалоги з клієнтами, що дозволяє компаніям краще розуміти потреби споживачів і оптимізувати стратегії взаємодії. Крім того, автоматизація процесів за допомогою розпізнавання мовлення дозволяє співробітникам зосередитися на більш стратегічних завданнях, підвищуючи загальну продуктивність і знижуючи ймовірність помилок, пов'язаних з людським фактором.

Інтеграція з іншими системами, такими як системи управління відносинами з клієнтами (CRM), дозволяє покращити взаємодію та управління даними. Це допомагає створювати більш персоналізований досвід для клієнтів і підвищує їхню задоволеність.

На рис 1.3 можна побачити схему інтеграції системи розпізнавання мовлення з CRM системою

Рис. 1.3 – Схема інтеграції системи розпізнавання мовлення з CRM системою [13]

Персональне використання та соціальний аспект. У повсякденному житті ця технологія реалізується у голосових помічниках (Siri, Alexa, Google Assistant), які інтерпретують команди користувачів. Також програмне забезпечення для конвертації мовлення у текст використовується для створення нотаток, нагадувань, ведення щоденників або диктування листів. [1].

Особливу соціальну роль розпізнавання мовлення відіграє для людей з інвалідністю, надаючи альтернативні засоби введення інформації та покращуючи доступність цифрових сервісів.

1.4 Проблеми та виклики у сфері розпізнавання мовлення

Перші спроби автоматичного розпізнавання мовлення почалися ще в 1950-х роках. Уже в 1952 році Девід, Бідалф і Балашек із компанії Bell Telephone Laboratories представили перший прототип системи, здатної розпізнавати мову. Це був один із перших завершених зразків такої технології.

У наступні десятиліття багато науково-дослідних центрів, зокрема й в Україні, активно працювали над удосконаленням цих систем, намагаючись усунути їхні технічні обмеження. Варто згадати, що ще в 1970-х роках у Радянському Союзі проводилися масштабні лінгвістичні дослідження, які заклали теоретичну основу для сучасних мовних технологій.

Однак справжній прорив у цій галузі відбувся лише у 1986 році. Саме тоді американське агентство DARPA (Агентство передових оборонних дослідницьких проєктів) розпочало серйозні розробки у сфері мовного розпізнавання, які дали новий імпульс розвитку технологій. [1].

Таблиця 1.2 – Історія технологій ASR [1]

Десятиліття	Еволюція ASR
1950s	Технологія розпізнавання мовлення була вперше представлена Bell Laboratories у 1950-х роках. Bell Labs створили віртуальний розпізнавач мовлення, відомий як «Одрі», який міг ідентифікувати числа від 1 до 9, коли їх вимовляв один голос.
1960s	У 1952 році IBM випустила свою першу систему розпізнавання голосу «Shoebox». Shoebox міг розуміти та розрізняти шістнадцять розмовних англійських слів.
1970s	У 1976 році Університет Карнегі-Меллона розробив систему «Гарпія», яка могла розпізнавати понад 1000 слів.
1990s	Після тривалого майже 40-річного очікування компанія Bell Technologies знову здійснила прорив у галузі, створивши інтерактивні системи розпізнавання голосу з телефонного зв’язку, які можуть диктувати людську мову.
2000s	Це був період змін для технології ASR, оскільки великий технологічний гігант Google почав працювати над технологією розпізнавання мовлення. Вони створили вдосконалене програмне забезпечення для мовлення з рівнем точності приблизно 80%, завдяки чому воно стало популярним у всьому світі.
2010s	Останнє десятиліття стало золотим періодом для ASR, коли Amazon і Apple запустили своє перше в історії мовленнєве програмне забезпечення на основі штучного інтелекту Alexa і Siri.

В Україні дослідження в цій сфері також мають тривалу історію — понад шістдесят років. Сьогодні важливу роль у розвитку автоматичного розпізнавання образів відіграє Українська асоціація з обробки інформації та розпізнавання образів, яка об’єднує провідних фахівців з технічних університетів країни.

Розробка систем розпізнавання мовлення й далі залишається актуальним і стратегічно важливим напрямом у сфері штучного інтелекту. Але водночас існує чимало проблем, які розробникам доводиться враховувати.

Однією з ключових є варіативність мовлення. Люди можуть говорити з різною швидкістю, інтонацією, акцентом, використовувати синоніми та навіть скорочення. Через це системи непросто точно інтерпретувати сказане. Рішенням цієї проблеми стало використання нейронних мереж і глибокого машинного навчання, які дозволяють системам адаптуватися до мовних відмінностей, навчаючись на великих обсягах даних.

Інша важлива проблема — це завади в аудіо, такі як шум, переривання мовлення чи нечітка вимова. У таких випадках застосовуються алгоритми обробки звуку та акустичної фільтрації, які допомагають очистити сигнал і підвищити точність розпізнавання.

Серйозним викликом є також обмежений обсяг навчальних даних для рідковживаних мов. Тут у пригоді стають підходи, що базуються на перенесенні знань із близьких мов, а також краудсорсингові платформи, де користувачі допомагають збирати мовні дані.

Ще один нюанс — вузькоспеціалізовані сфери, наприклад медицина чи право, де точність розпізнавання має вирішальне значення. Для таких галузей створюються спеціалізовані системи з урахуванням галузевої термінології та специфічних мовних моделей.

Окремо варто згадати і проблему надмірної чутливості систем. Вони іноді помилково сприймають шум або випадкові звуки як мовні команди. Це питання вирішується за допомогою алгоритмів, які аналізують контекст і фільтрують зайві сигнали, що дозволяє уникати небажаних спрацьовувань.

РОЗДІЛ 2 ПОСТАНОВКА ЗАДАЧІ

2.1 Обгрунтування вибору та актуальність теми

У сучасних умовах безперервного науково-технічного прогресу технології активно розвиваються, удосконалюються та інтегруються у всі сфери життя. Водночас взаємодія людини з комп’ютером, тобто комунікація з інформаційними системами, досягає якісно нового рівня. Традиційні засоби введення, як-от клавіатура й миша, доповнюються більш природними, інтуїтивно зрозумілими формами, серед яких особливе місце посідає розпізнавання мовлення.

Голосове управління — керування пристроями за допомогою усних команд — відкриває нові можливості підвищення ефективності, зручності та доступності цифрових систем як у повсякденному житті, так і в професійній діяльності: в офісах, на виробництві, у сфері обслуговування тощо. У цьому контексті розробка програмної системи для розпізнавання голосових команд українською мовою є надзвичайно актуальним і перспективним напрямом.

Актуальність цієї теми зумовлена низкою важливих факторів:

1) Зростання популярності голосових інтерфейсів. У світі стрімко поширюються такі голосові помічники, як Google Assistant, Amazon Alexa, Siri. Проте більшість із них підтримують переважно англійську або інші найпоширеніші мови, тоді як українська залишається недостатньо представленою. Це обмежує її повноцінне застосування в щоденному спілкуванні з пристроями.

Згідно з популяризацією штучного інтелекту, на рис.2.1 можна побачити, в яких діях люди найчастіше використовують голосових асистентів

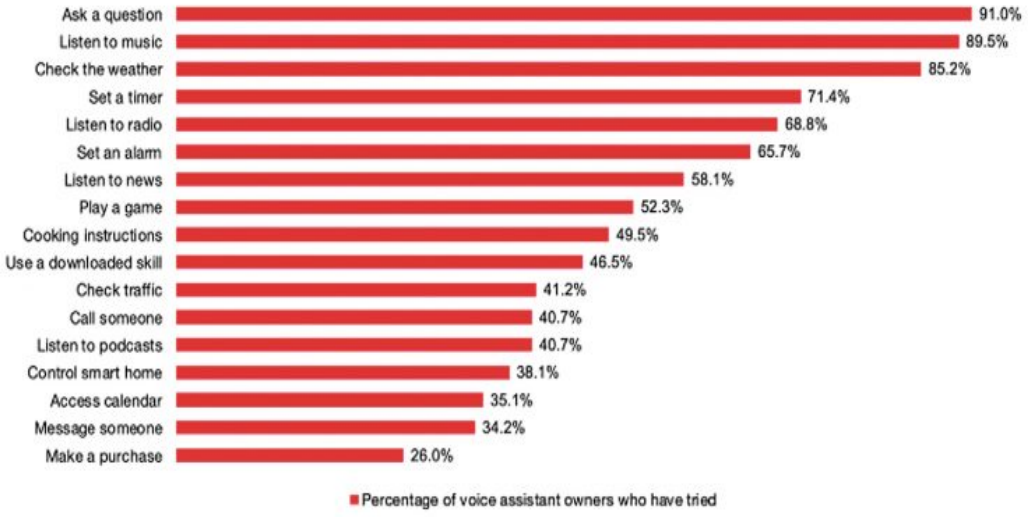

Рисунок 2.1 – Діаграма популярних дій, що виконуються за допомогою голосових помічників [21]

Також, можна зазначити популярність голосових асистентів на мобільних пристроях, нижче наведено рис. 2.2, де зображено, яка вікова категорія використовує голосових помічників

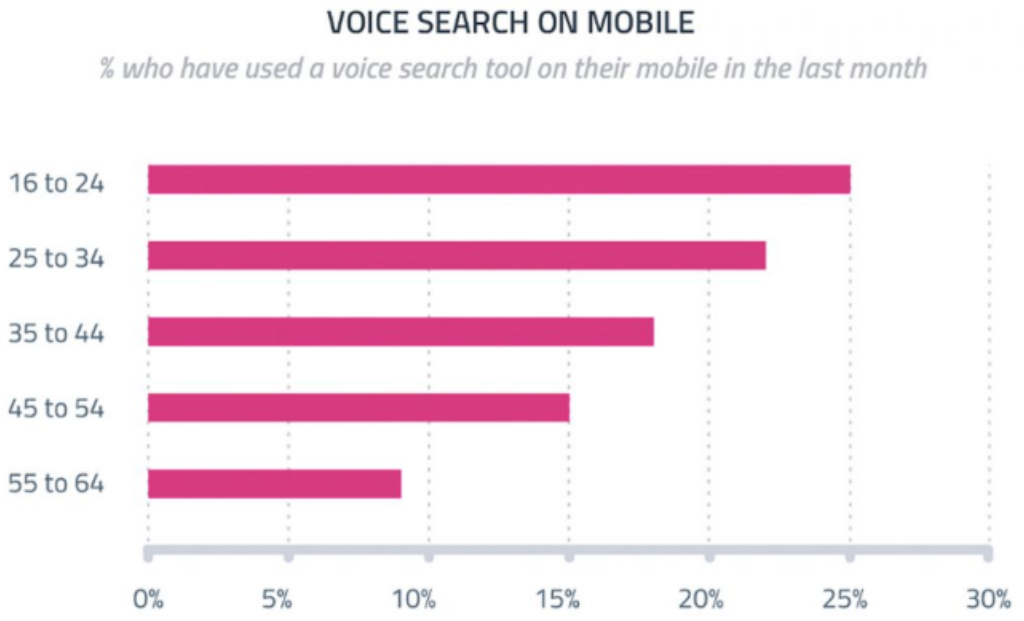

Рисунок 2.2 - Використання голосового пошуку на мобільних пристроях за віковими групами [21]

2) Потреба у національних мовних технологіях. Розробка системи саме для української мови сприятиме не лише її збереженню, а й активному розвитку в ІТ-сфері. Така система стане корисною для впровадження в державні сервіси, освітні платформи, медичні застосунки, банківські системи тощо, де використання української мови часто є законодавчо обґрунтованим.

3) Інклюзивність і доступність. Голосове управління є особливо важливим для осіб з обмеженими фізичними можливостями, які не можуть користуватись традиційними засобами введення, як-от клавіатура чи сенсорний екран. Розробка системи розпізнавання мовлення українською мовою значно розширить їхні можливості щодо взаємодії з цифровим середовищем.

4) Прогрес у сфері штучного інтелекту. Технології машинного навчання й глибинних нейронних мереж забезпечують високу точність розпізнавання мовлення, навіть в умовах шуму, акцентів або варіацій темпу й інтонації. Це дозволяє створювати адаптивні, чутливі до мовних особливостей користувача системи, які значно покращують взаємодію «людина-комп’ютер».

5) Потреби бізнесу та промисловості. У добу цифровізації бізнес-структури, логістика, виробництво та служби підтримки дедалі активніше впроваджують голосові інтерфейси. Вони дозволяють автоматизувати процеси, підвищити ефективність взаємодії та покращити клієнтський досвід. Україномовна система розпізнавання команд відкриє шлях до створення національного ІТ-продукту, здатного задовольнити специфічні потреби вітчизняного споживача та посилити економічну автономію.

6) Наукові виклики та дослідницький потенціал. Розпізнавання українського мовлення є складним технічним і лінгвістичним завданням через морфологічну багатозначність, гнучку граматику, варіативність наголосів і діалекти. Тому створення точної й надійної системи вимагає міждисциплінарного підходу, глибокого лінгвістичного аналізу та інноваційних методів у сфері обробки природної мови.

Вибір цієї теми також зумовлений особистим зацікавленням у сферах штучного інтелекту, обробки природної мови (NLP), програмної інженерії та

людино-комп’ютерної взаємодії. Реалізація проєкту передбачає не лише створення прикладної програмної системи, а й глибоке вивчення методів обробки аудіосигналів, навчання моделей розпізнавання мовлення, а також проєктування інтуїтивного користувацького інтерфейсу для зручної роботи з програмою.

Отже, розробка системи розпізнавання голосових команд українською мовою є не лише актуальним інженерним завданням, а й соціально значущим проєктом, що сприятиме технологічній незалежності України, підвищенню доступності інформації, розвитку національних мовних технологій і відкриттю нових перспектив для досліджень та інновацій у сфері цифрових рішень для бізнесу, освіти, медицини та державного управління.

2.2 Вибір підходу до розпізнавання мовлення

У наш час системи розпізнавання мовлення дедалі активніше впроваджуються у різні сфери життя — від смартфонів до “розумних” будинків. І, зокрема, правильний вибір підходу до розпізнавання мови — це, без перебільшення, ключовий крок до успішного впровадження таких технологій. На мою думку, тут важливо враховувати кілька критеріїв: тип мовлення, кількість дикторів, особливості словника, ну і навіть контекст, у якому працює система.

Слід, перш за все, зважати на тип мовлення, з яким має працювати система.

Ізольоване мовлення — це система, яка розпізнає окремі слова або команди, вимовлені чітко, з паузами. Як на мене, цей варіант ідеально підходить для простих голосових інтерфейсів — скажімо, у побутовій техніці чи, наприклад, в автомобілях. Перевагою таких систем є простота реалізації і висока точність, бо немає потреби в аналізі мовного потоку — все обмежується розпізнаванням чітких, розділених слів.

Зв’язане мовлення — дуже схоже на ізольоване, але тут слова вимовляються майже без пауз, хоча й залишаються досить чіткими. Такі системи дозволяють більш природне спілкування з комп’ютером, зберігаючи при цьому непоганий рівень точності.

Безперервне мовлення — це вже зовсім інший рівень. Користувач може говорити вільно, у своєму темпі, майже як у реальному житті. Отже, комп’ютер повинен не просто розпізнати слова, а ще й правильно розділити мовний потік, зрозуміти контекст і зміст. Я вважаю, що такі системи найкраще підходять для диктування текстів або стенографії, однак їхня розробка значно складніша, бо потрібно «зчитувати» не тільки слова, а й інтонацію, паузи, навіть акценти.

Спонтанне мовлення — найскладніший виклик для будь-якої системи. Тут людина говорить так, як звикла: з паузами, емоціями, обмовками, помилками. Виявляється, така мова часто включає неповні речення, повтори, навіть вигуки — і обробити це все, чесно кажучи, неймовірно складно. Тим не менш, сучасні ASR-системи (Automatic Speech Recognition), зокрема ті, що використовують штучний інтелект, уже здатні працювати з таким типом мовлення, хоча точність ще далека від ідеалу.

З іншого боку, варто враховувати і те, для скількох дикторів призначена система. Системи бувають дикторозалежні і дикторонезалежні.

Дикторозалежні системи "заточені" під конкретну людину. Якщо, наприклад, зробити таку систему для менеджера кол-центру, то вона буде чудово розпізнавати саме його голос. Плюс у тому, що її легко й недорого навчати, однак мінус — відсутність гнучкості: інші користувачі можуть мати проблеми з точністю.

Дикторонезалежні моделі — це вже складніша штука. Вони повинні розуміти будь-який голос, незалежно від того, хто говорить. До речі, саме такі системи використовуються у банках, контакт-центрах, голосових асистентах. Проте розробка таких систем складна і вимагає багато ресурсів. Іноді доводиться обмежувати словник, а це, відповідно, знижує гнучкість.

До речі, важливим чинником є також обсяг словника, з яким працює система. Це впливає як на точність, так і на швидкість розпізнавання.

Малий словник (1–100 слів чи фраз) — ідеальний для простих команд. Наприклад, “ввімкни світло”, “запусти музику”.

Середній словник (101–1000 слів) — дозволяє реалізувати вже більш розмовні інтерфейси, скажімо, у мобільних додатках.

Великий словник (1001–10 000 слів) та дуже великий (понад 10 000 слів) — це вже справжні мовні системи, які можна використовувати для перекладачів, систем диктування або розшифровки.

Але чим більший словник — тим складніше системі швидко й точно розпізнавати слова. Одним словом, тут важливо знайти баланс між можливостями та реальними потребами.

2.3 Аналіз якості розпізнавання мовлення

Оцінка якості розпізнавання мовлення є критичним етапом у розробці та впровадженні будь-якої системи автоматичного розпізнавання мовлення (АРМ, або ASR — Automatic Speech Recognition). Надійність системи,також, напряму залежить від її здатності точно інтерпретувати голосові команди користувача, враховуючи фон, темп мовлення, акценти, інтонації та інші фактори. Тому, згідно сказаного, глибокий аналіз якості розпізнавання дозволяє не лише обґрунтувати ефективність запропонованого підходу, але й визначити напрямки для подальшого вдосконалення.

Основні метрики оцінки якості розпізнавання мовлення:

1) Word Error Rate (WER) — один з основних індикаторів ефективності системи розпізнавання. Він розраховується як відношення кількості помилок до загальної кількості слів у правильній (еталонній) транскрипції. Коефіцієнт помилок слів (WER) обчислює кількість неправильно знайдених слів, поділену на загальну кількість слів у тексті. Нижчий WER вказує на кращу продуктивність

Помилки включають:

Заміни (substitutions) — коли розпізнане слово відрізняється від правильного.

Пропуски (deletions) — коли слово не було розпізнано.

Вставки (insertions) — коли система "додала" слово, якого не було в оригіналі.

Нижче наведено формулу для розрахунку WER, де враховуються заміни, пропуски та вставки

$$\mathrm{WER} = \frac{(S+D+I)}{N}$$

Ось так виглядає формула WER [11].

2) Character Error Rate (CER) — використовується при розпізнаванні мов із великою морфологічною варіативністю або тоді, коли важлива точність на рівні символів. CER особливо важлива для мов з довгими або складними словами, як-от українська.

Метрика коефіцієнта помилок символів (CER) кількісно визначає різницю між прогнозованим текстовим виводом вашої системи, таким як транскрибована мова або текст, витягнутий із зображень, та правильним еталонним текстом. Вона обчислює мінімальну кількість вставок, видалення та замін символів, необхідних для перетворення прогнозованого тексту на еталонний текст. Наприклад, у системі перетворення мовлення на текст, якщо окрема літера у складному медичному терміні неправильно транскрибується, метрика коефіцієнта помилок символів точно фіксує цю конкретну помилку, ніж метрики на рівні слів. Такий рівень точності є критично важливим для команд, які працюють над тональними або алфавітними мовами, де зміни окремих символів можуть суттєво змінити значення [17].

Метрика коефіцієнта помилок символів (CER) оцінює точність тексту на рівні символів, тоді як коефіцієнт помилок слів (WER) оцінює точність на рівні слів. Метрика коефіцієнта помилок символів особливо ефективна, коли невеликі помилки мають значні наслідки, наприклад, у завданнях кодування, формальних

документах або спеціалізованому жаргоні. На противагу цьому, WER більше підходить для оцінки ширших семантичних елементів, таких як зв'язність речень у чат-боті або плавність фраз у машинному перекладі. Кожен показник цінний у різних контекстах. Наприклад, один відсутній символ у назві продукту може спричинити плутанину в описі електронної комерції; у цьому випадку точність показника коефіцієнта помилок символів ефективно враховує цю помилку.

Нижче наведено, а саме CER розраховується як сума вставок, видалень та замін символів, поділена на загальну кількість символів у тексті посилання.

$$\mathrm{CER} = (\mathrm{S} + \mathrm{D} + \mathrm{I}) / \mathrm{N}$$

де:

I – Кількість вставок символів;

D – Кількість видалень символів;

S - Кількість замін символів;

N - Загальна кількість символів у тексті посилання; [17].

Оцінка F1 – це альтернативна метрика машинного навчання, яка оцінює прогностичні здібності моделі, детально розглядаючи її продуктивність за класами, а не загальну продуктивність, як це робиться за точністю. Оцінка F1 поєднує дві конкуруючі метрики – точність та повноту моделі, що призводить до її широкого використання в новітній літературі.

Метрика точності обчислює, скільки разів модель зробила правильний прогноз по всьому набору даних. Це може бути надійною метрикою лише за умови, що набір даних збалансований за класами, тобто кожен клас набору даних має однакову кількість вибірок [7].

Матриця плутанини відображає прогностичну ефективність моделі на наборі даних. Для набору даних бінарних класів (який складається, припустимо, з «позитивних» та «негативних» класів) матриця плутанини має чотири основні компоненти:

Істинно позитивні результати (TP): кількість зразків , правильно передбачених як «позитивні».

Хибнопозитивні результати (ХП): кількість зразків, помилково передбачуваних як «позитивні».

Істинно негативні результати (TN): кількість зразків , правильно передбачених як «негативні».

Хибнонегативні результати (ХН): кількість зразків, помилково передбачуваних як «негативні».

Матриця плутанини відображає прогностичну ефективність моделі на наборі даних. Як показано на рисунку 2.1, вона складається з чотирьох основних компонентів: істинно позитивні (TP), хибнопозитивні (FP), істинно негативні (TN) та хибнонегативні (FN)

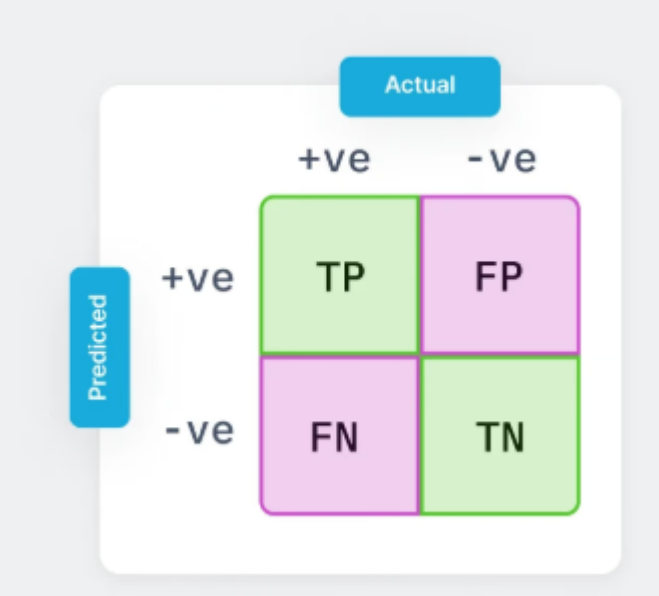

Рисунок 2.3 - Матриця плутанини [7]

Використовуючи компоненти матриці плутанини, ми можемо визначити різні показники, що використовуються для оцінки класифікаторів — точність, прецизійність, повноту та F1-бал.

Використовуючи компоненти матриці плутанини, ми можемо визначити різні показники, що використовуються для оцінки класифікаторів. Нижче представлені формули для обчислення точності (Precision) та повноти (Recall).

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

З точки зору чотирьох основних елементів матриці плутанини, F1 Score також можна записати формулу, наведену нижче [7].

$$F1\ Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

На табл. 2.1 представлено порівняльну таблицю open-source систем розпізнавання мовлення. Як видно з таблиці, Whisper (large) має найнижчі показники WER та CER, а також підтримує найбільшу кількість мов, включаючи українську

Таблиця 2.1 - Порівняльна характеристика open-source систем розпізнавання мовлення [20]

Система	**WER (%) ↓**	**CER (%) ↓**	**RTF ↓**	**Lang Support ↑**	**UA Support**
Whisper (large)	**~4–10**	**<5**	~0.8–1.2	**99+**	Так
Vosk	~12–20	~10–18	**<0.5**	~20	Так
Wav2Vec 2.0	~6–15	~5–12	~0.9–1.5	~10+ (з адаптацією)	Частково
DeepSpeech	~15–30	~12–25	~0.7–1.0	~15	Частково

На табл. 2.2 показано порівняльну таблицю хмарних (комерційних) систем розпізнавання мовлення. Згідно з таблицею, Google Speech-to-Text має найкращі показники якості та підтримки української мови, а також мінімальний RTF

Таблиця 2.2 - Порівняльна характеристика хмарних (комерційних) систем [20]

Система	WER (%) ↓	CER (%) ↓	RTF ↓	Lang Support ↑	UA Support
Google Speech-to-Text	**~4–8**	**<5**	**<0.5**	**120+**	Так
Amazon Transcribe	~6–12	~6–10	~0.6–0.9	~75	Частково
Microsoft Azure Speech	**~4–9**	~5–9	~0.6–1.0	~100+	Частково
IBM Watson Speech	~8–15	~10–18	~0.7–1.2	~50	Частково

Завдяки цим таблицям, ми можемо спокійно порівняти, які плюси і мінуси є в тій чи іншій системі.

РОЗДІЛ 3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАВАННЯ АУДІОФАЙЛІВ НА ТЕКСТ

3.1 Аналіз предметної області та вибір технології розпізнавання мовлення

Розпізнавання мовлення є складною задачею у сфері обробки природної мови та штучного інтелекту, що передбачає перетворення аудіосигналу, який містить людську мову, у текстовий формат для подальшого аналізу або використання.

Особливості української мови в контексті розпізнавання. Українська мова має низку фонетичних і лінгвістичних особливостей, які ускладнюють завдання розпізнавання. Зокрема, велика кількість голосних звуків, наголоси, діалектні варіації, зміни інтонації та мелодії вимови впливають на точність розпізнавання. Крім того, у реальних умовах часто виникають шуми або перешкоди, що суттєво знижують якість вхідного аудіо сигналу.

Виклики у шумовому середовищі. Розпізнавання у шумовому середовищі (наприклад, вулиця, офіс, кафе) потребує застосування алгоритмів, стійких до перешкод. Класичні методи шумозаглушення і сучасні нейронні мережі дозволяють покращити якість сигналу до розпізнавання.

Вибір технології: Google Speech-to-Text API. На мою думку, для реалізації розпізнавання в проєкті була вибрана сервісна технологія Google Speech-to-Text API. Це хмарне рішення, яке базується на гібридних нейронних мережах, що поєднують CNN (конволюційні мережі) та RNN (рекурентні мережі), а також LSTM (довготривала короткочасна пам’ять), що дозволяє ефективно працювати з аудіо послідовностями.

Для реалізації систем розпізнавання мовлення, зокрема таких як Google Speech-to-Text API, використовуються архітектури нейронних мереж, подібні до зображеної на рис.3.1

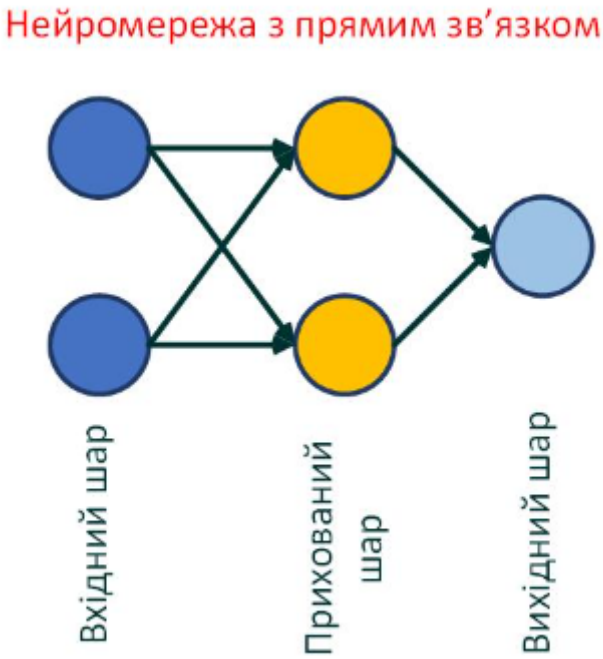

Рисунок 3.1 - Структура нейронної мережі з прямим зв'язком, що використовується в системах розпізнавання мовлення [18]

Використання Google API має декілька переваг:

1) Відсутність необхідності самостійного навчання та налаштування моделей. Google надає високоякісні, постійно оновлювані моделі.

2) Підтримка української мови з урахуванням лінгвістичних особливостей.Можливість роботи зі складними аудіо, включаючи фонові шуми.

Вхідні дані та їх обробка.Вхідними даними є аудіофайли у форматі WAV, які мають параметри: 1)Частота дискретизації — 16 кГц; 2) Моно канал; 3) Формат PCM (Linear16);

Аудіофайл проходить мінімальну попередню обробку — нормалізацію гучності та збереження у відповідному форматі для сумісності з API. Внутрішня

обробка аудіо (виділення спектрограм, MFCC — коефіцієнтів мел-частотної кепстральної характеристики) здійснюється Google на своїй стороні.

3.2. Реалізація інтеграції Google Speech API

Для запису аудіо, користувач через графічний інтерфейс, створений на базі Tkinter, може розпочати запис голосу за допомогою кнопки. Запис здійснюється за допомогою бібліотеки pyaudio, яка захоплює аудіопотік із мікрофону.

Інтерфейс програми для запису аудіо. Опираючись на рисунок 3.2, можна побачити, що інтерфейс програми включає дві основні кнопки — «Запис» та «Стоп», які відповідають за керування процесом захоплення голосу [16].

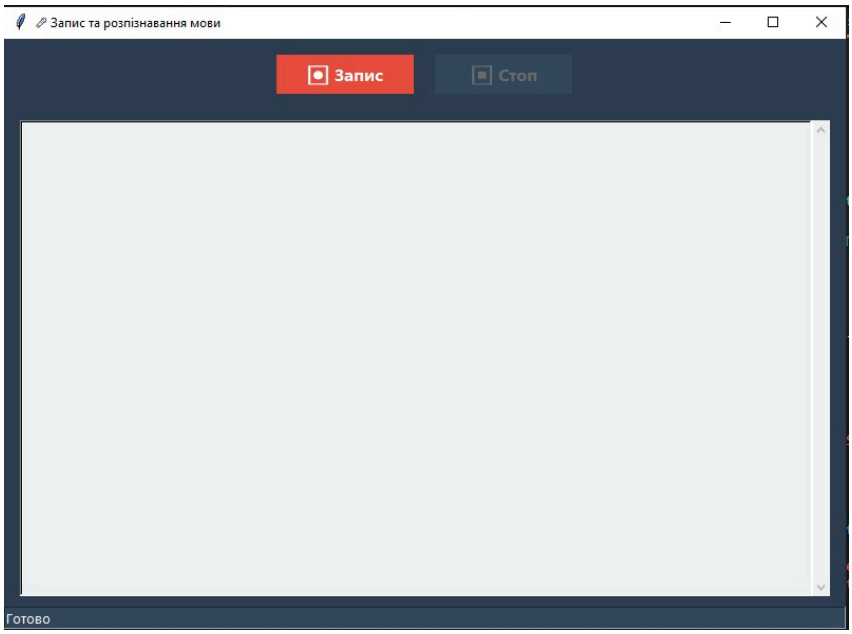

Рисунок 3.2 – Інтерфейс програми з кнопками "Запис" і "Стоп"

Після завершення запису аудіодані зберігаються у локальний файл формату .wav у відповідній директорії проєкту з унікальним іменем (наприклад, record_YYYYMMDD_HHMMSS.wav), що дозволяє уникнути конфліктів і спрощує подальшу обробку.

Використання .env та .gitignore. Для безпечної роботи з Google API ключі аутентифікації зберігаються у файлі .env у форматі:

Цей файл виключено з репозиторію за допомогою .gitignore, щоб уникнути випадкового розповсюдження конфіденційної інформації.

Взаємодія з Google Speech-to-Text API. Після збереження аудіофайлу відбувається його передача у Google Speech API для розпізнавання. Процес виконується так:

Ініціалізується клієнт Google Speech із завантаженим ключем.

Файл аудіо читається у бінарному режимі і упаковується у об'єкт RecognitionAudio.

Створюється конфігурація RecognitionConfig, де вказується формат аудіо, частота дискретизації і мова (uk-UA).

Виконується виклик client.recognize(), який повертає текстову транскрипцію.

Відображення результатів розпізнавання. Як показано на рисунку 3.3, після обробки аудіо вікно програми оновлюється і містить розпізнаний текст — результат роботи Google Speech API [16].

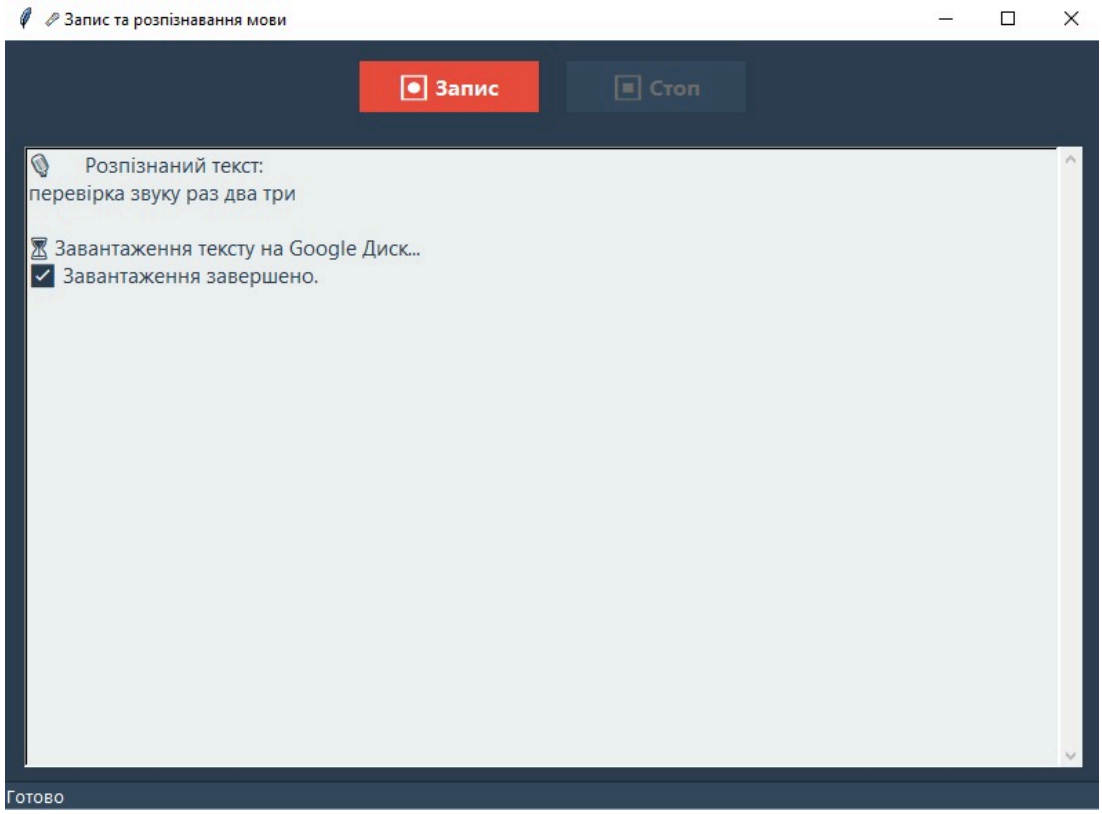

Рисунок 3.3 – Інтерфейс програми з відображенням розпізнаного тексту

Збереження результатів

Отриманий текст зберігається у локальний текстовий файл .txt з іменем, що відповідає аудіофайлу (наприклад, record_YYYYMMDD_HHMMSS.txt). Цей файл також зберігається у спеціальній директорії проекту.

Для кращого розуміння процесу нижче наведено рисунок 3.4, що ілюструє основні етапи роботи системи — від запису голосу до збереження результату розпізнавання тексту.

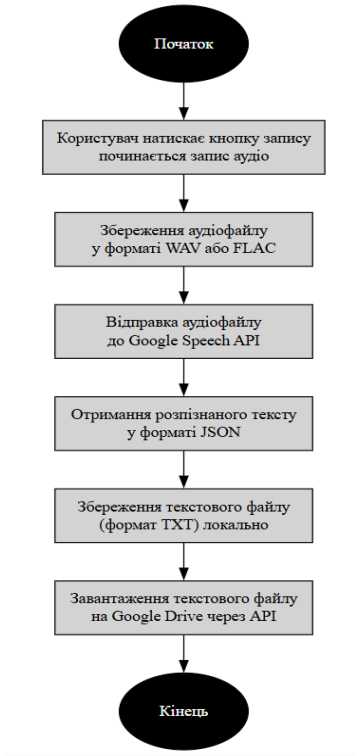

Рисунок 3.4 – Діаграма етапів роботи системи

Ось як саме виглядають етапи роботи системи від самого запису голосу до збереження результату розпізнавання тексту

3.3 Автоматизація збереження та інтеграція з Google Drive

Підключення до Google Drive API. Для автоматичного збереження результатів у хмарне сховище реалізовано інтеграцію з Google Drive API. Авторизація користувача відбувається через OAuth 2.0, де токени і конфіденційні дані зберігаються у .env. Це дає змогу безпечно завантажувати файли з локальної машини на Google Drive без повторного введення логіну.

Механізм роботи. Після завершення розпізнавання та створення текстового файлу запускається окремий процес, який виконує:

Завантаження текстового файлу та відповідного аудіофайлу на Google Drive у заздалегідь визначену папку.

Обробку помилок мережі або авторизації з автоматичними спробами повтору. [9]

Підтвердження успішного завантаження користувачу через інтерфейс.

На рисунку 3.5 показано приклад вмісту Google Drive після автоматичного завантаження — збережено аудіофайл та відповідний текстовий документ.

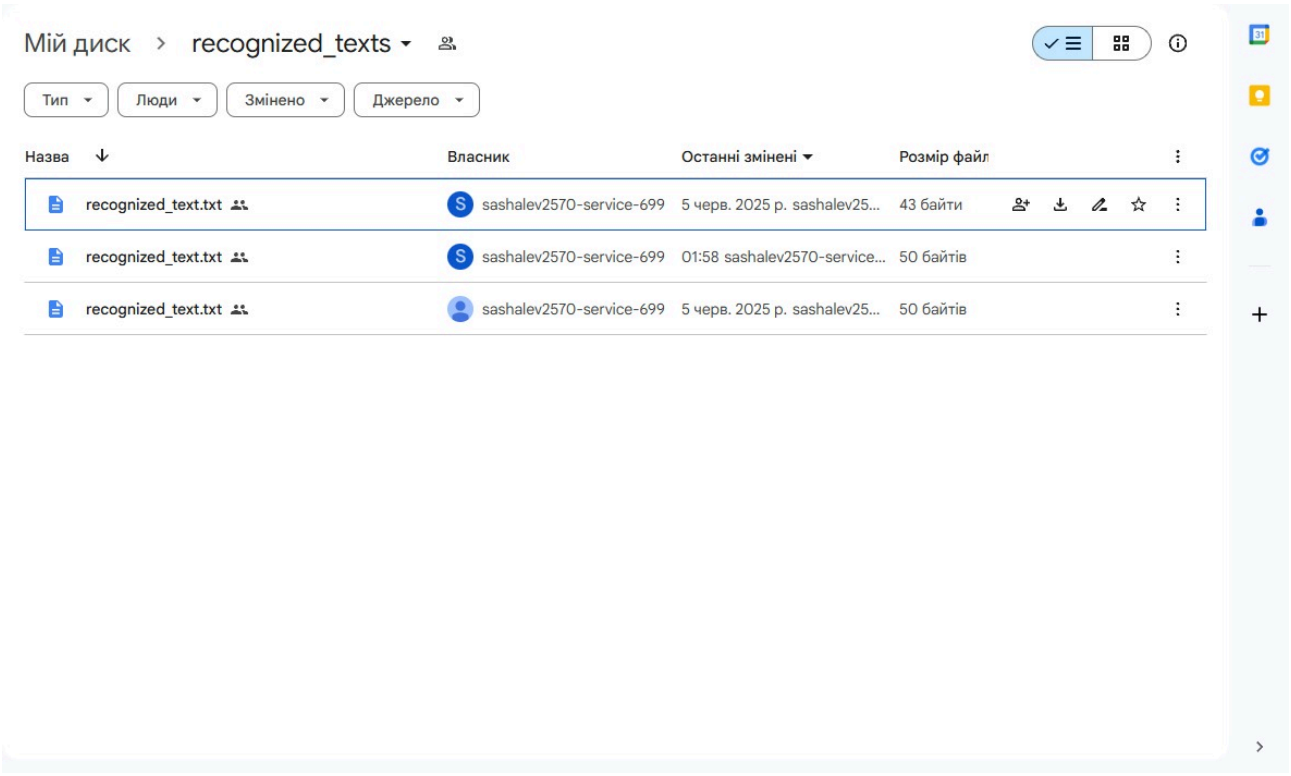

Рисунок 3.5 – Вміст Google Drive з файлами розпізнаних текстів

Використання багатопоточності. Щоб інтерфейс програми залишався чуйним і не блокувався під час очікування відповіді від сервера Google, весь процес розпізнавання і завантаження файлів винесено в окремий потік (через threading або concurrent.futures).

Формати файлів

Аудіофайли зберігаються у форматі .wav з параметрами, які забезпечують сумісність із Google Speech API.

Текстові транскрипти зберігаються у форматі .txt — простому текстовому файлі без форматування, що забезпечує сумісність із більшістю текстових редакторів [3].

РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Аналіз структури та функцій технологічного процесу

Своєчасне виявлення небезпек має вирішальне значення для розробки та впровадження ефективних заходів щодо запобігання нещасним випадкам і травмам. Оскільки не завжди можливо виявити небезпечні умови заздалегідь, а вивчення небезпечних дій може вимагати значного часу для збору відповідних даних, методи виявлення цих небезпек повинні бути відповідним чином диференційовані

На основі аналізу небезпечних умов, присутніх у виробничому процесі, виділяють наступні групи відповідно до їх впливу на працівників:

1. Вплив на обладнання, де оцінюється стан або рівень небезпеки, пов’язаний з використовуваним обладнанням.
2. Створення умов, що сприяють виникненню технологічних помилок у обслуговуючого персоналу під час виробничого процесу.
3. Створення умов і можливостей для проникнення працівників у небезпечні зони.
4. Причини небезпечних дій, що виникають через недостатню підготовку працівників та організацію навчання з охорони праці.

Моделі, що ілюструють формування та виникнення травмонебезпечних та аварійних ситуацій в комп’ютерному залі, представлені у вигляді моделі, що зображає процес їх формування та виникнення [14].

Відповідно до аналізу небезпечних умов, які існують у виробничому процесі виокремлено такі наступні за характером дії на працівника їх групи [14]:

- характеризують стан або рівень небезпеки обладнання, які використовуються.

- сприяють виникненню технологічних помилок обслуговуючого персоналу впродовж виробничого процесу;

- створювати умови та можливість проникнення працівника в небезпечну зону;

- приводять до виникнення небезпечних дій (внаслідок низького рівня професійної підготовки працівників та організації навчання з охорони праці).

Таблиця 4.1 – Моделі становлення та виникнення небезпечних ситуацій

Вид робіт, виробничий підрозділ, робоче місце, виробниче обладнання, склад агрегату	Виробнича небезпека			Можливі наслідки	Заходи запобігання небезпечним ситуаціям
	Небезпечна умова (НУ)	Небезпечна дія (НД)	Небезпечна ситуація (НС)		
Виконання робіт із електрообладнанням	Не вимкнено живлення. Відсутність заземлення.	Нехтування правилами ТБ	Ураження струмом	Травма (Т)	Проведення повторного інструктажу з ТБ. Розробка нових способів захисту. Встановлення заземлення.
НД НУ → НС → Т					

Моделі формування та виникнення травмонебезпечних і аварійних ситуацій в комп’ютерному кабінеті представлено у вигляді моделі формування та виникнення травмонебезпечних і аварійних ситуацій – табл. 4.1.

4.2 Електробезпека в офісі чи закладі

Промислові приміщення можна освітлювати штучним або природним світлом. Природне світло, якщо воно належним чином організоване, є найбільш сприятливим для людини. Основні вимоги до освітлення включають:

1Достатня освітленість для швидкого і легкого розпізнавання об'єктів роботи.

2Рівномірне освітлення без різких тіней.

3Відсутність сліпучого світла, що може шкодити працівнику.

4Неприпустимість обмеження рівня освітленості в залежності від часу.

Розрахунок штучного освітлення починається з визначення висоти розташування світильників і їх кількості. Висоту (м) розташування світильників над робочим місцем знаходимо за допомогою такої формули:

$$h_н = H - \left(h_1 + h_2\right), (4.1)$$

де H – висота приміщення, м; h_1 – відстань від підлоги до поверхні освітлення, м; h_2 – віддаль від стелі до світильника, м.

$$h_н = 4,5 - \left(2,2 + 1,5\right) = 0,8 \text{м}.$$

Якщо світильники розташовані симетрично на вершинах квадратів, їх кількість можна визначити за допомогою наступної формули:

$$n_c = \frac{S_n}{l^2}, (4.2)$$

де l – віддаль між світильниками, м.

Підставивши значення отримаємо:

$$n_c = \frac{36}{9} = 4 \text{од}.$$

Тоді світловий потік буде становити:

$$F_л = \frac{1,3 \cdot 36 \cdot 300}{4 \cdot 0,25 \cdot 0,545} = 2576,2 \text{ Лк}.$$

З урахуванням світлового потоку 2576,2 Лк, ми обираємо тип лампи – LED-трубка T8 (G13) довжиною 1200 мм, яка замінює люмінесцентну 40 Вт. Як варінат для вибору може бути – Philips CorePro LEDtube 1200mm 16.5W 840, або Osram SubstiTUBE Advanced T8 1200mm 16.8W.

4.3 Забезпечення безпеки в умовах надзвичайних ситуацій

Забезпечення безпеки населення та території в умовах загроз та надзвичайних ситуацій є одним з найважливіших завдань держави. Захист населення передбачає систему масштабних заходів, які реалізуються центральними та місцевими органами виконавчої влади, органами виконавчої влади, управлінням надзвичайних ситуацій та цивільного захисту, підпорядкованими системами та підприємствами, які забезпечують широкий спектр організаційних, технічних, санітарних, гігієнічних, епідеміологічних та інших заходів у сфері запобігання та ліквідації наслідків надзвичайних ситуацій [14].

Загрози життєвим інтересам громадян, держави та суспільства поділяються на зовнішні та внутрішні. Вони можуть виникати внаслідок різноманітних природних катастроф, техногенних аварій та військових конфліктів. Принципи захисту базуються на основних положеннях Женевської конвенції про захист жертв воєнних дій та її Додаткових протоколів з урахуванням специфіки можливих бойових дій та реальних можливостей держави забезпечити матеріальну базу для захисту. Приймаються системні заходи з метою ефективного захисту населення та зменшення збитків у разі виникнення надзвичайних ситуацій [14].

Одним із важливих аспектів є система попередження та інформування населення, яка досягається через попереднє створення та постійну готовність національних, територіальних та об'єктових систем попередження.

ВИСНОВКИ

У ході виконання дипломної роботи було проведено глибоке дослідження та розробку системи розпізнавання мовлення, що використовує сучасні технології глибокого навчання. Основною метою дослідження було створення ефективної системи, здатної точно перетворювати аудіосигнали у текстовий формат, що має важливе значення для автоматизації процесів обробки мовної інформації.

У процесі роботи було проведено детальний аналіз предметної області, що дозволило визначити ключові аспекти та виклики, пов'язані з розпізнаванням мовлення. Особлива увага приділялася аналізу форматів аудіофайлів та їх впливу на якість розпізнавання, що є критично важливим для забезпечення точності системи. Було досліджено різні методи попередньої обробки аудіосигналів, включаючи фільтрацію шумів, нормалізацію амплітуди та сегментацію сигналу, які значно покращують якість вхідних даних і, відповідно, точність розпізнавання.

Для реалізації системи розпізнавання мовлення було обрано Google Speech-to-Text API, що дозволило ефективно обробляти аудіосигнали та отримувати точні текстові транскрипції. Використання цього API забезпечило високу якість розпізнавання та підтримку української мови, що є важливим для розвитку національних мовних технологій. Розроблена програмна система включає запис аудіо, обробку аудіосигналів та інтеграцію з Google Speech-to-Text API, що дозволяє автоматично зберігати результати розпізнавання у текстові файли та завантажувати їх на Google Drive. Це забезпечує зручність та безпеку зберігання даних, що є важливим для практичного застосування системи.

Оцінка якості розпізнавання мовлення проводилася за допомогою метрик Word Error Rate (WER) та Character Error Rate (CER), які дозволили визначити ефективність запропонованих методів та алгоритмів. Ці метрики є стандартом у галузі розпізнавання мовлення та дозволяють оцінити точність системи на рівні

слів та символів. Проведена оцінка показала, що розроблена система має високу точність розпізнавання, що підтверджує ефективність використаних методів та технологій.

Крім того, у роботі було розглянуто питання охорони праці та безпеки в надзвичайних ситуаціях, що є важливим аспектом при роботі з комп'ютерною технікою та програмним забезпеченням. Запропоновані заходи забезпечують безпечні умови праці та захист даних у системах мовного розпізнавання, що є необхідним для успішного впровадження системи в реальних умовах.

Розроблена система розпізнавання мовлення має широке практичне значення та може бути використана в різних сферах, включаючи освіту, медицину, бізнес та державне управління. Вона дозволяє автоматизувати процеси обробки аудіоданих та покращити взаємодію людини з комп'ютером, що є особливо важливим для осіб з обмеженими фізичними можливостями. Система може бути інтегрована в різні програмні продукти та сервіси, що розширює її функціональні можливості та сфери застосування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Характеристика ASR. 2024. URL: https://uk.shaip.com/blog/automatic-speech-recognitiona-complete-overview/ (дата звернення 04.05.2025)

2.Розпізнавання голосу URL: https://uk.shaip.com/blog/voice-recognition-overview-and-applications/ (дата звернення 04.01.2025)

3.Попередня обробка аудіо URL: https://milvus.io/ai-quick-reference/how-do-speech-recognition-systems-manage-audio-preprocessing?utm_source=chatgpt.com (дата звернення 03.01.2025)

4. Формати аудіофайлів та їх вплив на якість розпізнавання: URL: https://userua.org.ua/yaki-formaty-muzyky-zaraz-isnuyut (дата звернення: 22.04.2025).

5. Огляд існуючих speech-to-text моделей та сервісів URL: https://sendpulse.ua/blog/transcribe-audio-into-text (дата звернення: 01.03.2025).

6. Аналіз якості розпізнавання мовлення URL: https://galileo.ai/blog/character-error-rate-cer-metric (дата звернення: 04.03.2025).

7. Precision, Recall та F1-score URL: https://www.v7labs.com/blog/f1-score-guide (дата звернення: 08.03.2025).

8. Реалізація інтеграції Google Speech API URL: https://inbase.com.ua/integracziya-za-dopomogoyu-api-shho-cze-take-yiyi-funkcziyi-ta-mozhlyvosti-u-produktah-inbase/ (дата звернення 05.02.2025)

9. Google Cloud Speech-to-Text | Google Cloud Skills Boost URL:https://www.cloudskillsboost.google/course_templates/700 (дата звернення 04.05.2025)

10. Обгрунтування вибору та актуальність теми розпізнавання мовлення URL: https://eitca.org/artificial-intelligence/eitc-ai-gcml-google-cloud-machine-learning/introduction/what-is-machine-learning/what-is-text-to-speech-tts-and-how-it-works-with-ai/ (дата звернення 02.04.2025)

11. Пояснення WER та коефіцієнт помилок слів URL: https://www.rev.com/resources/what-is-wer-what-does-word-error-rate-mean (дата звернення 04.01.2025)

12. Зниження шуму URL: https://uk.wikipedia.org/wiki/%D0%97%D0%BD%D0%B8%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F_%D1%88%D1%83%D0%BC%D1%83 (дата звернення 14.04.2025)

13. Інтеграція голосового помічника у CRM систему URL: https://krayincrm.com/integrating-voice-assistance-into-your-crm/?utm_source=chatgpt.com (дата звернення 10.05.2025)

14. Інструкція з охорони праці при роботі на персональному комп'ютері URL: https://services.uteka.ua/ua/publication/zrazky-34-trudovi-vidnosyny-ta-oplata-pratsi-138-instrukciya-po-oxrane-truda-pri-rabote-na-personalnom-kompyutere-obrazec (дата звернення 04.04.2025)

15.Ознайомлення з Python URL: https://aws.amazon.com/what-is/python/ (дата звернення 24.04.2025)

16. Використання API для перетворення мовлення в текст від Google для транскрибування аудіо в Python https://www.assemblyai.com/blog/google-speech-to-text-api-python (дата звернення 02.05.2025)

17. CER пояснення URL: https://galileo.ai/blog/character-error-rate-cer-metric (дата звернення 09.04.2025)

18. Принципи побудови нейронних мереж URL: https://itmaster.biz.ua/programming/vision/neural-networks-principles.html (дата звернення 24.04.2025)

19. Охорона праці та безпека експлуатації ПК URL: https://opcb.kpi.ua/?p=3590 (дата звернення 01.05.2025)

20. Deepgram. Benchmarking top open-source speech models URL: https://deepgram.com/learn/benchmarking-top-open-source-speech-models?utm_source=chatgpt.com (дата звернення: 16.04.2025)

21. Голосові помічники URL: https://hurma.work/blog/voice-assistants-shho-cze-i-navishho-voni-hr-ok-google-alexa-siri-cortana/ (дата звернення 10.04.2025)

ДОДАТКИ

Додаток А

App_gui.py

```
from upload_drive import upload_to_drive
import tkinter as tk
from tkinter import scrolledtext, messagebox
import threading
import numpy as np
import sounddevice as sd
import soundfile as sf
import speech_recognition as sr
class Recorder:
def __init__(self, filename="output.wav", samplerate=16000, channels=1):
self.filename = filename
self.samplerate = samplerate
self.channels = channels
self.recording = False
self.frames = []
def start(self):
self.recording = True
self.frames = []
self.thread = threading.Thread(target=self._record)
```

```
self.thread.start()

def _record(self):

def callback(indata, frames, time, status):

if self.recording:

self.frames.append(indata.copy())

else:

raise sd.CallbackStop()

with sd.InputStream(samplerate=self.samplerate, channels=self.channels,
callback=callback):

while self.recording:

sd.sleep(100)

def stop(self):

self.recording = False

self.thread.join()

data = np.concatenate(self.frames, axis=0)

sf.write(self.filename, data, self.samplerate)

class App:

def __init__(self, root):

self.root = root

self.root.title(" 🎤 Запис та розпізнавання мови")

self.root.geometry("650x500")

self.root.configure(bg="#2c3e50") # Темний фон

self.recorder = Recorder()

self.recognizer = sr.Recognizer()

btn_font = ("Segoe UI", 12, "bold")
```

```
text_font = ("Segoe UI", 11)

btn_frame = tk.Frame(root, bg="#2c3e50")

btn_frame.pack(pady=15)

self.btn_record = tk.Button(

btn_frame, text=" Запис", command=self.start_recording,

bg="#e74c3c", fg="white", activebackground="#c0392b",

font=btn_font, width=12, relief=tk.FLAT

)

self.btn_record.grid(row=0, column=0, padx=10)

self.btn_stop = tk.Button(

btn_frame, text=" Стоп", command=self.stop_recording, state=tk.DISABLED,

bg="#34495e", fg="white", activebackground="#2c3e50",

font=btn_font, width=12, relief=tk.FLAT

)

self.btn_stop.grid(row=0, column=1, padx=10)

self.text_area = scrolledtext.ScrolledText(

root, wrap=tk.WORD, width=70, height=20,

font=text_font, bg="#ecf0f1", fg="#2c3e50", relief=tk.SUNKEN, borderwidth=2

)

self.text_area.pack(padx=15, pady=10, fill=tk.BOTH, expand=True)

self.status_var = tk.StringVar()

self.status_var.set("Готово")

status_bar = tk.Label(root, textvariable=self.status_var, bd=1, relief=tk.SUNKEN,
anchor=tk.W, bg="#34495e", fg="white", font=("Segoe UI", 10))

status_bar.pack(side=tk.BOTTOM, fill=tk.X)
```

```
def start_recording(self):

self.text_area.delete('1.0', tk.END)

self.status_var.set("Запис розпочато...")

self.recorder.start()

self.btn_record.config(state=tk.DISABLED, bg="#7f8c8d")

self.btn_stop.config(state=tk.NORMAL, bg="#e74c3c")

def stop_recording(self):

self.recorder.stop()

self.status_var.set("Запис зупинено. Розпізнавання...")

self.btn_record.config(state=tk.NORMAL, bg="#e74c3c")

self.btn_stop.config(state=tk.DISABLED, bg="#34495e")

threading.Thread(target=self.recognize_audio).start()

def recognize_audio(self):

try:

with sr.AudioFile(self.recorder.filename) as source:

audio = self.recognizer.record(source)

text = self.recognizer.recognize_google(audio, language="uk-UA")

self.text_area.insert(tk.END, f"🎙️ Розпізнаний текст:\n{text}\n")

txt_filename = "recognized_text.txt"

with open(txt_filename, "w", encoding="utf-8") as f:

f.write(text)

self.status_var.set("Завантаження тексту на Google Диск...")

self.text_area.insert(tk.END, "\n⏳ Завантаження тексту на Google
Диск...\n")

upload_to_drive(txt_filename)
```

```
self.text_area.insert(tk.END, "✅ Завантаження завершено.\n")

self.status_var.set("Готово")

except sr.UnknownValueError:

self.text_area.insert(tk.END, "\n❌ Не вдалося розпізнати мову.\n")

self.status_var.set("Готово")

except sr.RequestError as e:

self.text_area.insert(tk.END, f"\n❌ Помилка сервісу розпізнавання: {e}\n")

self.status_var.set("Готово")

except Exception as e:

self.text_area.insert(tk.END, f"\n❌ Помилка під час завантаження на Google Диск: {e}\n")

self.status_var.set("Готово")

if __name__ == "__main__":

root = tk.Tk()

app = App(root)

root.mainloop()
```

Audio_recognition.py

```
import speech_recognition as sr

def record_audio_stream():

r = sr.Recognizer()

mic = sr.Microphone()

mic_stream = mic.listen_in_background(mic, lambda x, y: None)

return r, mic, mic_stream
```

```
def record_audio_start():
r = sr.Recognizer()
mic = sr.Microphone()
mic.adjust_for_ambient_noise(mic)
stream = None
print("Готовий до запису")
return r, mic, stream
def record_audio_stop(r, mic, source, timeout=None):
print("Запис іде... Говоріть")
audio = r.listen(source, timeout=timeout)
filename = "output.wav"
with open(filename, "wb") as f:
f.write(audio.get_wav_data())
print(f"Запис збережено у файл {filename}")
return filename
def recognize_audio(file_path):
r = sr.Recognizer()
with sr.AudioFile(file_path) as source:
audio = r.record(source)
try:
text = r.recognize_google(audio, language="uk-UA")
print("Розпізнаний текст (українською):", text)
return text
except sr.UnknownValueError:
```

```
print("Не вдалося розпізнати аудіо")
return ""
except sr.RequestError as e:
print(f"Помилка сервісу розпізнавання: {e}")
return ""
```

Upload_drive.py

```
from googleapiclient.discovery import build
from googleapiclient.http import MediaFileUpload
from google.oauth2 import service_account
SERVICE_ACCOUNT_FILE = 'service_account.json' # файл із ключами
SCOPES = ['https://www.googleapis.com/auth/drive.file']
def upload_to_drive(filename):
credentials = service_account.Credentials.from_service_account_file(
SERVICE_ACCOUNT_FILE, scopes=SCOPES)
drive_service = build('drive', 'v3', credentials=credentials)
folder_id = '1knerSXWVln2MU5RPamX0zs-Gcy8vJVIM' # встав свій ID папки Google Drive
file_metadata = {
'name': filename,
'parents': [folder_id]
}
media = MediaFileUpload(filename, mimetype='text/plain')
file = drive_service.files().create(
```

```
body=file_metadata,
media_body=media,
fields='id'
).execute()
print('Файл завантажено, ID:', file.get('id'))
if __name__ == "__main__":
upload_to_drive('recognized_text.txt')
```