

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
РОЗПІЗНАВАННЯ МОВИ ЗА ГОЛОСОМ КОРИСТУВАЧА
RESEARCH AND SOFTWARE IMPLEMENTATION OF
METHODS FOR SPEECH RECOGNITION BY USER'S VOICE

Виконав: студент(ка) 2 курсу, групи КНТ-119м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Драчук Є.А.

(прізвище та ініціали)

Керівник Льовкін В.М.

(прізвище та ініціали)

Рецензент Зеленьова І.Я.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти магістр
 Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “_____” _____ 2020 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Драчука Євгенія Анатолійовича
(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів розпізнавання мови за голосом користувача
Research and Software Implementation of Methods for Speech Recognition by User's Voice

керівник проєкту (роботи) Льовкін Валерій Миколайович, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “3” листопада 2020 року № 301

2. Строк подання студентом проєкту (роботи) 18 грудня 2020 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження, 2. Обґрунтування методу розпізнавання мови за голосом користувача, 3. Проєктування архітектури системи, 4. Основні рішення щодо реалізації компонентів системи, 5. Експлуатація та тестування програми, 6. Економіко-організаційна частина, 7. Охорона праці та безпека в надзвичайних ситуаціях.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Льовкін В.М.доцент		
Економіко-організаційна частина	Пожуєва Т.О., професор		
Охорона праці та безпека в надзвичайних ситуаціях	Коробко О.В., ст. викладач		
Нормоконтролер	Калініна М.В., асистент		

7. Дата видачі завдання “ 7 ”вересня2020року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2–3 тижні	Розділ 1
3	Розробка власного методу.	4–5 тижні	Розділ 2
4	Розробка архітектури програми.	6 тиждень	Розділ 3
5	Розробка програми.	7–8 тижні	Розділ 4
6	Тестування та відлагодження програмного забезпечення.	9 тиждень	Розділ 5
7	Розробка економіко-організаційної частини	10 тиждень	Розділ 6
8	Розробка заходів з охорони праці та безпеки з надзвичайних ситуацій.	11 тиждень	Розділ 7
9	Оформлення пояснювальної записки та документів до неї.	12-13 тиждень	Додатки
10	Нормоконтроль та рецензування.	14–15 тижні	
11	Захист роботи.	16 тиждень	

Студент(ка)

 (підпис) (прізвище та ініціали)

Керівник проєкту (роботи)

 (підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
97 с., 23 рис., 9 табл., 3 дод., 31 джерело.

ГОЛОСОВІ КОМАНДИ, РОЗПІЗНАВАННЯ ОБРАЗІВ, ПРИРОДНЯ
МОВА, РОЗПІЗНАВАННЯ МОВИ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ,
TENSORFLOW, KERAS, PYTHON.

Об'єкт дослідження – процес розпізнавання мови за голосом користувача. Предмет дослідження – методи розпізнавання мови за голосом користувача.

Мета роботи – програмна реалізація методів розпізнавання мови за голосом користувача.

Матеріали, методи та технічні засоби: мова програмування Python, бібліотеки Keras та Tensorflow, згорткові нейронні мережі.

Результати. Представлено результати проєктування програмного забезпечення у вигляді структурної схеми, функціональної схеми та діаграм прецедентів.

Наукова новизна роботи полягає в тому, що в ній запропоновано метод розпізнавання мови за голосом користувача, який ґрунтується на використанні згорткової нейронної мережі для аналізу мовлення користувача з попереднім перетворенням аудіосигналу до спектрограм, що дозволяє виконувати визначення мови за голосом користувача.

Висновки. Розглянуто проблему розпізнавання мови за голосом користувача. Виконано огляд методів розпізнавання мовлення, серед яких зокрема: метод ковзного вікна та приховані марківські моделі, нейронні мережі, метод моделей заповнювачів. Виконано огляд архітектури згорткових нейронних мереж.

Галузь використання – голосові асистенти, розпізнавання мовлення.

ABSTRACT

Explanatory note to the diploma qualifying work of the master:
97 pages, 23 figures, 9 tables, 3 appendixes, 31 sources.

VOICE COMMANDS, PATTERN RECOGNITION, NATURAL LANGUAGE, LANGUAGE RECOGNITION, CONVOLUTIONAL NEURAL NETWORKS, TENSORFLOW, KERAS, PYTHON.

The object of research is the process of speech recognition by the user's voice. The subject of research is methods of speech recognition by user's voice.

The purpose of the work is the software implementation of speech recognition methods by the user's voice.

Materials, methods and technical means: Python programming language, Keras and Tensorflow libraries, convolutional neural networks.

Results. The results of software design in the form of block diagram, functional diagram and diagrams of precedents are presented.

The scientific novelty of the work is that it proposes a method of speech recognition by user's voice, which is based on the use of convolutional neural network for analysis of user's speech with preliminary conversion of audio signal to spectrograms, which allows to determine language by user's voice.

Conclusions. The problem of speech recognition by the user's voice is considered. A review of speech recognition methods, including: sliding window method and hidden Markov models, neural networks, the method of filler models. A review of the architecture of convolutional neural networks is performed.

Field of use – voice assistants, speech recognition.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз проблеми та постановка завдань дослідження	11
1.1 Проблема розпізнавання мовлення	11
1.2 Методи розпізнавання мовлення	14
1.3 Висновки за розділом 1	19
2 Обґрунтування методу розпізнавання мови за голосом користувача	21
2.1 Задача розпізнавання мови за голосом користувача	21
2.2 Аналіз архітектур нейронних мереж.....	21
2.3 Аналіз згорткових нейронних мереж.....	25
2.4 Метод розпізнавання мови за голосом користувача	29
2.5 Висновки за розділом 2	32
3 Проєктування системи.....	33
3.1 Визначення функціональних вимог до програмного забезпечення	33
3.2 Вибір мови програмування	35
3.3 Структурна схема програмного забезпечення	38
3.4 Висновки за розділом 3	39
4 Основні рішення щодо реалізації компонентів системи.....	41
4.1 Алгоритм функціонування системи.....	41
4.2 Основні рішення щодо реалізації компонентів системи.....	42
4.3 Висновки за розділом 4	45
5 Експлуатація та тестування програми	46
5.1 Призначення програми	46
5.2 Умови виконання програми	46
5.3 Виконання програми.....	47
6 Економіко-організаційна частина.....	51
6.1 Планування розробки програмного продукту.....	51
6.2 Розрахунок основної заробітної плати.....	53

6.3 Розрахунок додаткової заробітної плати	54
6.4 Відрахування на єдиний соціальний внесок	54
6.5 Визначення витрат на матеріали	55
6.6 Витрати на спеціальне устаткування	56
6.7 Накладні витрати.....	60
6.8 Оцінка економічної ефективності	60
7 Охорона праці та безпека в надзвичайних ситуаціях.....	62
7.1 Аналіз потенційних небезпек.....	62
7.2 Заходи забезпечення безпеки.....	62
7.3 Заходи з виробничої санітарії та гігієни праці.....	63
7.4 Заходи з пожежної безпеки	68
7.5 Заходи з забезпечення безпеки у надзвичайних ситуаціях.....	70
Висновки	72
Перелік джерел посилання	73
Додаток А Технічне завдання	77
Додаток Б Текст програми	82
Додаток В Слайди презентації.....	90

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ЕОМ – електронно-обчислювальна машина;

ПК – персональний комп'ютер.

ВСТУП

В даний час існують численні технічні засоби, що можуть сприймати (розпізнавати) вимовлені мовні повідомлення: комп'ютери, медичне електронне устаткування, автомобілі, мобільні телефони тощо [1]. Кожний з таких засобів може бути основою системи голосового управління.

Побудова системи голосового управління є в даний час актуальним завданням:

- такі системи здатні істотно полегшити взаємодію користувача з комп'ютерною системою, особливо ця ідея розвинена в концепції так званих «розумних будинків»;
- іноді голосовий інтерфейс є необхідною компонентою, наприклад, коли мова йде про людей з порушеннями опорно-рухового апарату [2].

Завданням таких систем є виділення і розпізнавання з потоку звукового сигналу (як мовного, так і не мовного) заздалегідь визначеного набору мовних команд: прикладом такої команди може служити фраза «Включити світло», при цьому система не повинна реагувати на інші ділянки мовного сигналу, включаючи і ті, які містять окремі слова зумовлених команд [2].

Однак, найчастіше така система повинна підтримувати роботу з голосом користувача різними мовами. Відповідно в процесі виникає завдання розпізнавання мови, якою взаємодіє користувач з системою, що повинна уніфікувати подальшу роботу з командами користувача, уникаючи подальшої необхідності вибирати зі всіх варіантів команд та кожної команди всіма доступними мовами.

Тому можна вважати проблему розпізнавання мови за голосом користувача актуальною.

В якості мети роботи було визначено виконання програмної реалізації методів розпізнавання мови за голосом користувача.

Об'єкт дипломної кваліфікаційної роботи магістра— процес розпізнавання мови за голосом користувача.

Предмет дипломної кваліфікаційної роботи магістра— методи розпізнавання мови за голосом користувача.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Проблема розпізнавання мовлення

Система автоматичного розпізнавання мовлення – це елемент процесу обробки мовлення, призначення якого – забезпечити зручний діалог між користувачем і машиною: в широкому розумінні йдеться про системи, які здійснюють фонемне декодування мовного акустичного сигналу при вимові повідомлень вільним стилем, довільним диктором, без урахування проблемної орієнтації та обмежень на обсяг словника, у вузькому сенсі системи автоматичного розпізнавання мовлення полегшують рішення окремих завдань, накладаючи деякі обмеження на вимоги до розпізнавання природної усної мови в класичному його розумінні, відповідно діапазон різновидів систем автоматичного розпізнавання мовлення простягається від простих автономних пристроїв і дитячих іграшок, які здатні розпізнавати або синтезувати окремо вимовлені слова, цифри, міста, імена, до суперскладних систем розпізнавання природного усного мовлення і його синтезу для використання, наприклад, в якості секретаря-референта: будучи основною складовою будь-якого дружнього інтерфейсу між машиною і людиною, система автоматичного розпізнавання мовлення може бути вбудована в різні додатки, наприклад, в системи голосового контролю, голосового доступу до інформаційних ресурсів, навчання мови за допомогою комп'ютера, допомоги недієздатним, доступу до чого-небудь через системи голосової верифікації/ідентифікації [3].

Розпізнавання мовлення включає в себе два основні етапи.

Перший етап полягає в попередній обробці сигналу: на етапі попередньої обробки вихідний сигнал перетворюється в вектори ознак, на основі яких потім буде проведена класифікація, цей етап може включати в себе наступні кроки:

- перетворення сигналу з аналогової форми в цифрову;
- застосування фільтрів для придушення шумів;

- виділення границь мовлення;
- виділення ознак сигналу: найбільш поширені методи виділення ознак – це метод крейда-частотних кепстральних коефіцієнтів і метод кепстральних коефіцієнтів на основі лінійного передбачення [4].

Другий етап передбачає класифікацію сигналу: після отримання набору ознак необхідно на їх основі визначити, що за звук або слово знаходилося в початковому сигналі, найбільш розповсюджені методи – це приховані марківські моделі і нейронні мережі [4].

Більшість сучасних систем автоматизованого розпізнавання використовують модульну архітектуру з використанням блоку шумоочистки, детектора голосу, перетворювача сигналу в вектори особливостей і головного модуля, що включає алгоритм розпізнавання ключового слова: цифровий сигнал спочатку надходить в модуль шумоочистки, де підвищується якість сигналу внаслідок видалення шумів і внесеного каналом спотворення, потім детектор голосу виділяє ділянки сигналу, що містять мовлення, ці ділянки за допомогою модуля перетворення сигналу в вектори особливостей перетворюються в набори коефіцієнтів, які надходять в головний модуль, в якому відбувається безпосереднє визначення наявності і розпізнавання команди [2].

Позначимо основні області застосування систем розпізнавання мовлення:

- автоматизований призначений для користувача інтерфейс: на сьогоднішній день для багатьох людей спілкування з комп'ютером все ще викликає труднощі, системи розпізнавання мовлення дозволяють долати ці труднощі, величезна перевага систем розпізнавання голосу в тому, що вони набагато швидше будь-яких інших типів інтерфейсів, голосова програма електронної пошти дозволяє включати комп'ютер, диктувати і послати повідомлення, не торкаючись до миші і клавіатури, також люди з фізичними вадами отримають більш ефективний спосіб взаємодії з комп'ютером, найбільш очевидне використання системи розпізнавання мовлення полягає в

створенні систем автоматичного стенографування, які можуть замінити секретарів при диктуванні голосом текстів листів, заміток у щоденник, доповідей, в такому випадку відбувається не тільки економія за рахунок скорочення роботи стенографіста, але і підвищення ступеня конфіденційності інформації;

– управління мобільними пристроями: відомо, наскільки незручно і небезпечно використання мобільних телефонів зі звичайним способом набору номера за кермом, тому останнім часом популярністю користуються мобільні телефони з голосовим набором, що позбавляють користувача від необхідності набирати потрібний номер вручну, досить вимовити ім'я абонента, і з'єднання відбудеться автоматично, аудіосистеми контролю і управління вже застосовуються в автомобілях, при цьому власник автомобіля голосом подає команди управління температурним режимом, радіо, навігаційною системою, які сприймають голос і виконують команди;

– інформаційні послуги: сучасні системи розпізнавання мовлення застосовуються, наприклад, для замовлення авіаквитків, перегляду новин, доступу до баз даних, технологія розпізнавання голосу швидко змінила ринок телефонних послуг, системи, що розпізнають розмовне мовлення, працюють в інформаційних телефонних центрах, ці системи дозволяють автоматизувати діалог з клієнтом, в результаті чого відпадає необхідність у величезній кількості операторів, які приймають телефонні дзвінки, і скорочуються витрати на утримання персоналу, до того ж поліпшується якість обслуговування клієнтів, так як з'єднання з машиною здійснюється практично відразу, позбавляючи клієнтів від тривалого очікування оператора на лінії, що звільниться;

– бізнес і професійна підтримка: уже багато років голосові диктофонні системи, призначені для представників певних професій, наприклад, лікарів і юристів, можна знайти на ринку програмних продуктів, багато представників цих професій використовують системи розпізнавання мовлення в

повсякденній роботі, стали популярні домашні прилади і пристосування, що активуються голосом;

– комбіновані людино-машинні інтерфейси: за останнє десятиліття області застосування таких систем значно розширилися і будуть продовжувати розширюватися, вони застосовуються, зокрема, для контролю обмеженого доступу до об'єкта за допомогою розпізнавання особи й мовлення людини, виконання фінансових операцій за допомогою мови і сенсорних екранів банкоматів [1].

1.2 Методи розпізнавання мовлення

Для виділення інформативних ознак мовленнєвого сигналу використовується спектральне подання мовлення, при цьому на першому етапі здійснюється отримання частотного спектра мовного сигналу за допомогою набору програмних смугових фільтрів (виконуючи так зване дискретне перетворення Фур'є), на другому етапі виконуються перетворення отриманого спектра мовного сигналу:

– логарифмічна зміна масштабу в просторі амплітуд і частот;
 – згладжування спектра з метою виділення його обвідної;
 – кепстральний аналіз, тобто зворотне перетворення Фур'є від логарифма прямого перетворення [5].

В даний час центральне місце в статистичних системах автоматичного розпізнавання мовлення належить імовірнісним акустико-фонетичним моделям, які називають також прихованими марківськими моделями:

– вони дозволяють вирішувати задачі розпізнавання мови, а також покращувати якість сигналу, забрудненого шумами і спотвореннями, моделювати джерело мовного сигналу, оптимізувати структуру діалогу та інші: прихована харківська модель, заснована на теорії дискретних випадкових ланцюгів, була вперше введена і вивчена в кінці 60-х- початку 70 - х рр. XX ст., це двічі стохастический процес, при цьому термін «двічі»

використовується для позначення такої пари процесів, один з яких є основним, але прихованим від нас і спостережуваним тільки через інший стохастичний процес, моделі такого типу особливо зручні для опису мовленнєвого сигналу, оскільки в дійсності тиск звукової хвилі, який вимірюється, являє собою лише певний код основного символічного процесу, що протікає в неспостережуваних і повністю недоступних ділянках мозку;

– у акустичному процесі, що спостерігається, виявляються вимірні фізичні кореляції лінгвістичної структури: при побудові моделі розпізнавання мови на основі прихованих марківських моделей вибирають їх основні параметри, що включають тип моделі (ергодична, модель Бекіса, ліво-права модель), розмір моделі (число станів), тип спостережуваних параметрів (дискретні або безперервні щільності розподілу векторів спостережень), при цьому якщо словник, що розпізнається, невеликий, то можна для кожного слова створити вручну топологію прихованих марківських моделей, при цьому зазвичай кількість станів моделі дорівнює кількості звуків (фонем) у слові, для розпізнавання мови з великим словником вкрай важко побудувати і навчити індивідуальну приховану марківську модель для кожного слова, тому кожне слово перетворюється в послідовність вимовлених фонем, і будується модель для кожної фонемі, як правило, модель фонемі має 3 стани (перший описує початок фонемі, другий представляє центральну частину фонемі і третій – закінчення фонемі);

– прихована марківська модель слова виходить шляхом з'єднання в ланцюжок моделей фонем з відповідного фонетичного алфавіту, аналогічним чином з'єднуються моделі слів один з одним, утворюючи моделі фраз і речень;

– системи автоматичного розпізнавання мови, що використовують статистичні методи, включають зазвичай три компонента: мовний словник, модель мови і набір статистичних акустичних моделей звукових одиниць;

– мовний словник (або лексикон) являє собою звичайний словник, в якому для кожного слова дана фонетична або фонемна транскрипція: слово

в лексиконі може мати декілька транскрипцій, що відображають фонетичну варіативність, яка не виводиться з простих комбінаторних і позиційних правил реалізації фонем у слові, у Лексикон можуть включатися не тільки слова, а й окремі словосполучення і навіть фрази;

– модель мови теоретично повинна задавати семантичні і синтаксичні обмеження, яким підкоряється побудова речень в даній мові, подібні обмеження можуть описуватися різними граматичними моделями, у більшості систем автоматичного розпізнавання мови використовуються так звані статистичні марківські граматики, які задають ймовірності для різних послідовностей слів мови, такі граматики називаються *n*-грамними (наприклад, біграмна модель задає ймовірності для всіх пар слів з лексикону, триграмна модель – для всіх трійок слів), на основі ймовірнісних характеристик проводиться оцінка граматичної правдоподібності кожного ланцюжка слів, ідентифікованого до даного моменту процесу розпізнавання;

– граматичні оцінки мають вирішальне значення при розпізнаванні злитого мовлення на основі великих словників, у системах розпізнавання злитого мовлення на базі великих словників основний тип використовуваної фонетичної моделі відповідає контекстній реалізації фонемі або, в термінології розробників таких моделей, трифона: трифон практично збігається з лінгвістичним алофоном, однак трифон, на відміну від традиційних алофонів, виділяється не на основі суб'єктивних слухових або артикуляційних критеріїв, а з урахуванням ступеня акустичних відмінностей і їх впливу на результати розпізнавання, зазвичай набір трифонів включає декілька тисяч контекстних реалізацій, відповідних фонемному інвентарю даної мови [6].

Ще одним методом, який може використовуватися для рішення проблеми, є нейронні мережі. При навчанні мережі з учителем можна навчити мережу розпізнавати об'єкти, що належать заздалегідь визначеному набору класів, а якщо ж мережа навчається без учителя, то вона може групувати об'єкти за класами відповідно до їх цифрових параметрів, таким

чином, на базі нейронних мереж можна створювати системи, що навчаються, та самонавчальні системи, висуваються наступні вимоги до самонавчальних систем:

- розробка системи полягає тільки в побудові архітектури системи: у процесі створення системи розробник створює тільки функціональну частину, але не наповнює (або наповнює в мінімальних обсягах) систему інформацією, основну частину інформації система отримує в процесі навчання;

- можливість контролю своїх дій з подальшим коригуванням: цей принцип говорить про необхідність зворотного зв'язку «Дія – Результат – Корекція» в системі, такі ланцюжки дуже широко поширені в складних біологічних організмах і використовуються на всіх рівнях – від контролю м'язових скорочень на найнижчому рівні до управління складними механізмами поведінки;

- можливість накопичення знань про об'єкти робочої області: знання про об'єкт – це здатність маніпулювати його образом в пам'яті, кількість знань про об'єкт визначається не тільки набором його властивостей, але ще й інформацією про його взаємодії з іншими об'єктами, поведінку при різних впливах, знаходження в різних станах, тобто його поведінку в зовнішньому оточенні: наприклад, знання про геометричний об'єкт передбачає можливість передбачити вид його перспективної проекції при будь-якому повороті і освітленні, ця властивість наділяє систему можливістю абстрагування від реальних об'єктів, тобто можливістю аналізувати об'єкт при його відсутності, відкриваючи тим самим нові можливості в навчанні;

- автономність системи: при інтеграції комплексу дій, які система здатна здійснювати, з комплексом датчиків, що дозволяють контролювати свої дії і зовнішнє середовище, наділена вищенаведеними властивостями система буде здатна взаємодіяти із зовнішнім світом на досить складному рівні, при цьому вона буде адекватно реагувати на зміну зовнішнього оточення (природно, якщо це буде закладено в систему на етапі навчання),

здатність коригувати свою поведінку в залежності від зовнішніх умов дозволить частково або повністю усунути необхідність контролю ззовні, тобто система стане автономною, можливість створення на базі штучних нейронних мереж самонавчальних систем є важливою передумовою для їх застосування в системах розпізнавання (і синтезу) мови [5].

Метод ковзного вікна є ще одним методом вирішення даної задачі. Даний метод характеризується наступним:

- суть методу ковзного вікна полягає у визначенні входження ключового слова за допомогою алгоритму Вітербі, який широко застосовується для розпізнавання злитого мовлення, цей алгоритм вирішує таку задачу: дано вектор спостережень, потрібно визначити найбільш підходящу послідовність прихованих харківських моделей і переходів між їх станами для цього вектора спостережень, далі будемо називати таку послідовність шляхом;

- так як ключове слово може починатися і закінчуватися в будь-якому місці сигналу, то цей метод перебирає всі можливі пари початку і кінця входження ключового слова і знаходить найвірогідніший шлях для ключового слова і цього відрізка, як якщо б ключове слово було присутнє в ньому;

- для кожного знайденого ймовірного шляху ключового слова застосовується функція правдоподібності, заснована на спрацьовуванні, якщо значення шляху, розраховане відповідно до застосовуваного методу оцінки шляху, більше визначеного значення, часто для оцінки шляху використовується значення ймовірності, отримане за допомогою алгоритму Вітербі;

- головним недоліком такого підходу є те, що він перебирає всі можливі варіанти входження ключового слова, що створює велику обчислювальну складність, крім цього, метод розпізнавання команди на основі цього алгоритму полягає в застосуванні його до всієї мовної ділянки для кожної можливої команди зі словника команд, такий підхід має два

суттєві недоліки: велика обчислювальна складність, команди можуть включати слова, які погано розпізнаються за допомогою алгоритму розпізнавання ключового слова [6].

Метод моделей наповнювачів є ще одним рішенням розглядаємої проблеми. Для алгоритмів розпізнавання ключового слова слово для розпізнавання представляється вбудованим в чужорідну мову, на цій підставі методи моделей наповнювачів обробляють цю чужорідну мову за допомогою явного моделювання чужорідної мови за рахунок другорядних моделей, для цього в словник системи розпізнавання додаються «узагальнені» слова, роль цих слів у тому, щоб будь-який сегмент сигналу незнайомого слова або немовної акустичної події був розпізнаний системою як одне слово або ланцюжок з узагальнених слів, для кожного узагальненого слова створюється і навчається акустична модель на корпусі даних з відповідними розміченими сегментами сигналу, на виході з декодера видається ланцюжок, що складається зі слів словника (ключових слів) і узагальнених слів, узагальнені слова потім відкидаються, і решта ланцюжка вважається результатом розпізнавання, при цьому недоліком підходу з використанням слів-заповнювачів є висока ймовірність помилки, коли ключові слова розпізнаються як узагальнені, крім цього, постає і питання про оптимальний вибір алфавіту узагальнених слів [6].

1.3 Висновки за розділом 1

Виконано огляд проблеми розпізнавання мовлення. Основних понять, етапів даного процесу, а також напрямків практичного застосування.

Виконано огляд методів розпізнавання мовлення, серед яких зокрема: метод ковзного вікна та приховані марківські моделі, нейронні мережі, метод моделей заповнювачів.

В якості завдань дипломної кваліфікаційної роботи магістравизначено наступні:

- дослідження методів розпізнавання мови за голосом користувача;
- розробка технічного завдання;
- проєктування програмного забезпечення;
- розробка програмного забезпечення;
- тестування та налагодження програми;
- економічне обґрунтування доцільності розробки.

2 ОБГРУНТУВАННЯ МЕТОДУ РОЗПІЗНАВАННЯ МОВИ ЗА ГОЛОСОМ КОРИСТУВАЧА

2.1 Задача розпізнавання мови за голосом користувача

Нехай ϵ множина аудіофрагментів, що представляють мовлення користувача, $M = \{M_i\}$, $i = 1..n$, де M_i – фрагмент мовлення користувача (аудіоповідомлення) під номером i , i множина $Y = \{Y_i\}$, $i = 1..n$, де Y_i – мова, якою відбувається мовлення.

Задача розпізнавання мови за голосом користувача полягає в тому, щоб для кожного повідомлення M_i , визначеного аудіофрагментом заданої довжини, визначити клас, до якого воно належить, представлений мовою повідомлення, на основі використання функції $F(b)$, яка виражена на основі формули (2.1):

$$F(M_i) = \{Y_i\}. \quad (2.1)$$

Відповідно для розв'язання задачі розпізнавання мови за голосом користувача потрібно визначити вигляд функції $F(M_i)$. У такому випадку можна буде автоматизувати процес розв'язання задачі за допомогою математичного і програмного забезпечення.

Одним зі способів визначення даної залежності є створення нейронної мережі.

2.2 Аналіз архітектур нейронних мереж

У більшості нейронних моделей тип нейрона пов'язаний з його розташуванням в мережі (якщо нейрон має тільки вихідні зв'язки, то це вхідний нейрон, якщо навпаки – вихідний нейрон), однак можливий випадок, коли вихід топологічно внутрішнього нейрона розглядається як частина виходу мережі, в процесі функціонування мережі здійснюється перетворення

вихідного вектора у вихідний, деяка переробка інформації, конкретний вид виконуваного мережею перетворення даних обумовлюється не тільки характеристиками нейроподібних елементів, а й особливостями її архітектури, а саме топологією міжнейронних зв'язків, вибором певних підмножин нейроподібних елементів для введення і виведення інформації, способами навчання мережі, наявністю або відсутністю конкуренції між нейронами, напрямком і способами управління і синхронізації передачі інформації між нейронами, з точки зору топології можна виділити три основні типи нейронних мереж:

- повнозв'язні: у повнозв'язних нейронних мережах кожен нейрон передає свій вихідний сигнал іншим нейронам, в тому числі і самому собі, всі вхідні сигнали подаються всім нейронам, вихідними сигналами мережі можуть бути всі або деякі вихідні сигнали нейронів після декількох тактів функціонування мережі;

- багат шарові: у багат шарових нейронних мережах нейрони об'єднуються в шари, шар містить сукупність нейронів з єдиними вхідними сигналами, число нейронів в шарі може бути будь-яким і не залежить від кількості нейронів в інших шарах, у загальному випадку мережа складається з шарів, пронумерованих зліва направо, зовнішні вхідні сигнали подаються на входи нейронів вхідного шару (його часто нумерують як нульовий), а виходами мережі є вихідні сигнали останнього шару, крім вхідного і вихідного шарів у багат шаровій нейронній мережі є один або декілька прихованих шарів, зв'язки від виходів нейронів деякого шару до входів нейронів наступного шару називаються послідовними, у свою чергу;

- слабозв'язані: у слабозв'язаних нейронних мережах нейрони розташовуються у вузлах прямокутної або гексагональної решітки, кожен нейрон пов'язаний з чотирма (околиця фон Неймана), шістьма (околиця Голі) або вісьмома (околиця Мура) своїми найближчими сусідами [7].

Серед багат шарових нейронних мереж виділяють наступні типи:

– монотонні – це окремий випадок шаруватих мереж з додатковими умовами на зв'язки і нейрони: кожен шар крім останнього (вихідного) розбитий на два блоки (збудливий і гальмуючий), зв'язки між блоками теж поділяються на ті, які гальмують, і збуджуючі: якщо від нейронів блоку А до нейронів блоку В ведуть тільки збуджуючі зв'язки, то це означає, що будь-який вихідний сигнал блоку є монотонною неубивающою функцією будь-якого вихідного сигналу блоку А. Якщо ж ці зв'язки тільки гальмують, то будь-який вихідний сигнал блоку В є незростаючою функцією будь-якого вихідного сигналу блоку А. Для нейронів монотонних мереж необхідна монотонна залежність вихідного сигналу нейрона від параметрів вхідних сигналів;

– мережі без зворотних зв'язків: у таких мережах нейрони вхідного шару отримують вхідні сигнали, перетворюють їх і передають нейронам першого прихованого шару, і так далі аж до вихідного, який видає сигнали для інтерпретатора і користувача, якщо не визначено інше, то кожен вихідний сигнал шару подається на вхід всіх нейронів шару, проте можливий варіант з'єднання шару з довільним шаром, серед багат шарових мереж без зворотних зв'язків розрізняють повнозв'язні (вихід кожного нейрона шару пов'язаний з входом кожного нейрона шару) і частково повнозв'язні, класичним варіантом даних мереж є повнозв'язні мережі прямого поширення;

– мережі зі зворотними зв'язками: у мережах із зворотними зв'язками інформація з наступних шарів передається на попередні, серед них, в свою чергу, виділяють наступні: шарувато-циклічні (відрізняються тим, що шари замкнуті в кільце, останній шар передає свої вихідні сигнали першому, всі верстви рівноправні і можуть як отримувати вхідні сигнали, так і видавати вихідні), шарувато-повнозв'язні (складаються з шарів, кожен з яких представляє собою повнозв'язну мережу, а сигнали передаються як від шару до шару, так і всередині шару; в кожному шарі цикл роботи розпадається на три частини, прийом сигналів з попереднього шару, обмін сигналами

всередині шару, вироблення вихідного сигналу і передача до подальшого шару), повнозв'язно-шаруваті (за своєю структурою аналогічні шарувато-повнозв'язним, але функціонують по-іншому: в них не розділяються фази обміну всередині шару і передачі наступному, на кожному такті нейрони всіх верств приймають сигнали від нейронів як свого шару, так і наступних) [7].

За результатами проведеного аналізу було сформульовано таблицю для порівняння різних архітектур нейронних мереж, серед яких було виділено ті, які можуть бути актуальними для вибору в даній задачі (табл. 2.1).

Таблиця 2.1 – Порівняння архітектур нейронних мереж

Критерій порівняння	Багатошаровий перцептрон	Згорткові нейронні мережі	Рекурентні нейронні мережі
Тип даних	Універсальна	Зображення	Послідовності
Задача класифікації	Придатна	Придатна	Придатна

Як видно, основні сучасні архітектури нейронних мереж придатні для розв'язання задачі класифікації (2.1), хоча при цьому кожна з них має певні вимоги щодо вхідних даних. За цим критерієм можна виділити наступне:

– аудіофрагменти відносно саме розпізнавання мови не є задачею класифікації за послідовністю загалом, оскільки в даному випадку від всієї сукупності залежить остаточний результат, на відміну від задачі розпізнавання мовлення, тобто сенсу сказаного, коли саме послідовність слів є важливою для отримання результату;

– для виконання класифікації аудіоданих можуть використовуватися багатошарові перцептрони, але при цьому дана модель є базовою і не враховує особливостей даних сама по собі;

– згорткові нейронні мережі вимагають роботи з зображеннями, відповідно для цього потрібно випрацювати механізм перетворення аудіоданих у зображення.

2.3 Аналіз згорткових нейронних мереж

Багатошарові нейронні мережі є надзвичайно ефективними для вирішення декількох категорій проблем, таких як класифікація, оцінка, наближення, розпізнавання тощо, однак вони можуть швидко досягти обмеження, особливо з дуже високовимірними вхідними даними:

– наприклад, потрібно створити модель глибокого навчання для виконання завдання класифікації котів чи собак;

– модель навчається на наборі даних із зображеннями у відтинках сірого кольору розміру 420x420 пікселів;

– після навчання потрібно, щоб модель могла сказати, що це зображення собаки чи kota;

– першим підходом було б, наприклад, використання багатошарового персептрона з вхідним шаром 128 нейронів, оскільки кожен нейрон шару отримує значення 420x420 пікселів як вхідне і присвоює вагу значенню пікселя, буде отримано на нейрон $420 \times 420 = 176400$ ваг, помножених на загальну кількість 128 нейронів, до яких потрібно ще додати 128 зміщень (одне зміщення на нейрон), відповідно даній моделі доведеться вивчити понад 22 мільйони параметрів, при цьому маючи тільки 128 нейронів і один проміжний шар, якщо потрібна більш потужна модель, це займе декілька додаткових шарів і більше нейронів на шар, кількість параметрів збільшується, наприклад, тришарова модель 128–256–1 вимагає майже 23 мільйони параметрів, така модель потребує величезної кількості даних для навчання;

– оскільки параметри, що підлягають коригуванню, набагато численніші, фаза навчання триватиме довше, а продуктивність падає, при

цьому навіть якщо була б достатня обчислювальна потужність, неможливо було б правильно навчити цю модель;

– навчання багатошарового перцептрона вимагає надання вхідних даних у вигляді вектора (одновимірний масив): у випадку, якщо вхідні дані є матричним зображенням, його доведеться вирівнювати, щоб отримати вектор, однак у природних зображеннях є дуже міцні зв'язки між сусідніми пік селями, відповідно вирівнюючи зображення, втрачається ця інформація, мережі стає складніше інтерпретувати дані під час навчального етапу, це виправдовує інтерес до іншої архітектури нейронної мережі для вирішення подібних проблем – згорткових нейронних мереж: згорткова нейронна мережа може успішно фіксувати просторові та часові залежності в зображенні за допомогою застосування відповідних фільтрів, архітектура забезпечує краще пристосування до набору даних зображення за рахунок зменшення кількості задіяних параметрів та повторного використання ваг, іншими словами, мережу можна навчити краще розуміти зображення (рис. 2.1) [8].

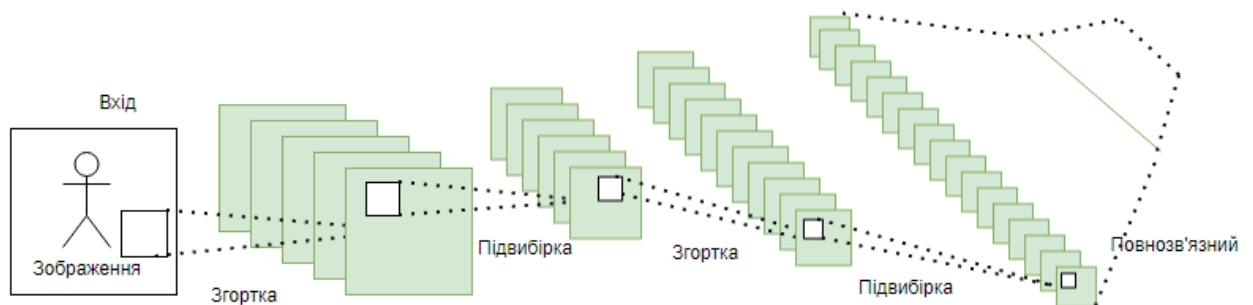


Рисунок 2.1 – Структура згорткової нейронної мережі

Основними будівельними блоками згорткової нейронної мережі є:

– шар згортки – фільтр, який іноді називають ядром, рухається по зображенню, переглядаючи декілька пікселів одночасно (наприклад, 3x3 або 5x5), результати виконання згортки підсумовуються в одне число, яке представляє всі пікселі, які фільтр спостерігав;

– шар активації: шар згортки генерує матрицю, яка значно менших розмірів, ніж вихідне зображення, ця матриця запускається через рівень активації, який вводить нелінійність, щоб дозволити мережі навчатися через зворотне розповсюдження, при цьому під час навчання коефіцієнти фільтрів оновлюються, вони можуть бути негативними, тоді виходить, що коефіцієнти карти ознак можуть мати велике негативне значення під час навчання, оскільки відомо, що ці значення являють собою рівні пікселів, а отже позитивні, можна застосувати функцію для заміни негативних значень нулями та збереження позитивних значень такими, якими вони є, це функція активації, яка називається ReLU;

– шар об'єднання (пулінгу) – це процес подальшого зменшення розміру матриці, при цьому фільтр рухається результатами попереднього шару і вибирає одне число з кожної групи значень (як правило, максимальне, це називається максимальним об'єднанням – maxpooling), це дозволяє мережі навчатися набагато швидше, зосереджуючись на найважливішій інформації в кожній особливості зображення (рис. 2.2): у природному зображенні існує сильна локальна кореляція між пікселями, це означає, що для зображення, якщо відомо, що піксель червоний, дуже ймовірно, що його чотири найближчі пікселі також мають відтінки червоного, а що стосується зображення у відтінках сірого, наприклад, якщо інтенсивність пікселя становить 180, то його найближчі пікселі також матимуть інтенсивність близько 180, отже, можна зменшити розміри зображення, зберігаючи лише одного представника на локальний блок пікселів, що називається об'єднанням;

– повністю пов'язаний шар – традиційна багат шарова структура перцептрона: його вхід – це одновимірний вектор, що представляє вихід попередніх шарів, його вихід – це перелік ймовірностей для різних можливих міток, прикріплених до зображення (наприклад, собака, кішка, птах), мітка, яка отримує найбільшу ймовірність, є рішенням задачі про класифікацію [8]-[10].

3	4	1	2				
5	2	3	2			5	3
7	4	5	3			8	6
8	5	4	6				

Рисунок 2.2 – Виконання операції максимального об'єднання

На рис. 2.2 приведено приклад виконання операції максимального об'єднання на фрагменті зображення.

У даному випадку зображення (набір комірок) було розбито на квадрати розміру 2×2 , у кожному такому квадраті було знайдено найбільше значення, отриманий результат було занесено в результуючу таблицю. Це дозволяє виділяти найбільш характерні значення властивостей і прибирати шум.

Приклад виконання згортки на фрагменті зображення наведено на рис. 2.3.

1_{x1}	1_{x0}	1_{x1}	0	0			
0_{x0}	1_{x1}	1_{x0}	1	0		3	
0_{x1}	0_{x0}	1_{x0}	1	1			
0	0	1	1	0			
0	1	1	0	0			

Рисунок 2.3 – Виконання операції згортки

Фільтр у даному випадку має розмір 3×3 . Значення фільтру записано в нижніх правих кутах клітинок. Відповідно виконання операцій відбувається

шляхом множення значень клітинок на значення відповідних позицій фільтру, в результаті чого отримані значення складаються і заносяться в результуючу таблицю.

Згортку виконано на помаранчевій зоні, жовті позиції – інші позиції в зображенні, які не підпали під фільтр, а результат виконання операції занесено в зелену зону.

Відповідно фільтр визначає ті властивості, які мають бути перевірені під час реалізації операції, тобто властивості, на які має бути звернено особливу увагу.

Згортка – це лінійне перетворення вхідних даних окремого вигляду, що передбачає отримання компоненту ($q1, q2$) наступного рівня, при цьому виконується скалярне множення пікселів вікна на вектор згортки, формула (2.2):

$$y_{q1,q2}^t = \sum_{-d \leq a, b \leq d} W_{a,b} x_{i+a, j+b}^t, \quad (2.2)$$

де $y_{q1,q2}^t$ – результат виконання згортки на рівні t ;

$x_{q1,q2}^t$ – вхід згортки, який одночасно є виходом всього попереднього шару, карта ознак в шарі під номером t ;

W – матриця ваг [11]-[12].

2.4 Метод розпізнавання мови за голосом користувача

Метод розпізнавання мови за голосом користувача дозволяє розв'язати задачу (2.1), яка за своєю суттю є задачею класифікації, адже визначає віднесення аудіофрагменту до одного з класів, яким є одна з множини підтримуваних навченою моделлю природніх мов.

Першим завданням, яке має бути вирішено в процесі роботи методу розпізнавання мови за голосом користувача, є завдання визначення вхідних даних. За своєю суттю вхідні дані є аудіофайлами.

Використання згорткових нейронних мереж потребує роботи з зображеннями замість аудіофайлів, тому спочатку потрібно перетворити аудіодані на зображення. Для цього має використовуватись спектрограма.

З аудіоданими працювати з нейронними мережами достатньо складно, адже вони потребують певного оброблення. Це може виконуватися або виділенням певних характеристик з цих даних, або, як запропоновано в моїй роботі, виконуватися на основі перетворення аудіоданих в зображення, адже для роботи з зображеннями існує достатньо адекватний механізм – згорткові нейронні мережі.

Відповідно на першому етапі має виконуватися перетворення аудіоданих в спектрограми.

По суті спектрограма є графічним представленням даних, в якому за абсцисою відкладається час відтворення, тобто кожна мить аудіофайлу, а за ординатою – частота. Інтентивність кольору при цьому визначає потужність частоти.

Такий формат дозволяє по суті візуалізувати інтенсивність і специфічність мовлення, що достатньо зручно саме для визначення припальної мови.

Для створення моделі використовується згорткова нейронна мережа, що складається з трьох згорткових шарів та повнозв'язного шару, на якому формується вихід мережі.

Тобто згорткова нейронна мережа має виділити специфічні ознаки, характерні для звукового представлення кожної відповідної мови і за виділеними цими ознаками на виході визначити, до якої мови належить цей фрагмент.

Після створення мережі реалізується підготовка даних шляхом створення набору спектрограм. Тобто всі наявні файли мають бути перетворені таким чином.

В якості основи використовувалися спектрограми для фрагментів по 10 секунд. На основі цього фрагменту виконувалось прийняття рішення. Відповідно будь-який присутній аудіофайл, для якого визначено вихідний клас, тобто мову, до якої належить цей аудіофайл, має бути розбито на фрагменти по 10 секунд, а для кожного фрагмента якості мітки (вихідного класу) має бути встановлено задану мову.

Вся множина фрагментів формує в результаті вибірку даних. Дана вибірка розбивається на навчальну і тестову вибірки. Розбиття відбувається за співвідношенням: 80 % навчальна вибірка, 20 % – тестова вибірка.

На наступному етапі забезпечується навчання нейронної мережі на навчальній вибірці даних. Тобто для кожного екземпляру вибірки реалізується навчання на розмічених даних до досягнення встановленого рівня помилки.

Після цього навчена нейронна мережа може використовуватися для завдання розпізнавання, тобто класифікації сигналу (віднесення до відповідної природної мови).

Для цього знову ж таки має бути сформовано з отриманого повідомлення набір фрагментів, кожен з яких має довжину 10 секунд. Після цього відповідний фрагмент має бути перетворений на окрему створену спектрограму.

Спектрограма кожного фрагменту подається на вхід навченої нейронної мережі, а на виході отримується клас (природня мова, до якої віднесено фрагмент).

Після визначення для кожного фрагмента його мови, відбувається голосування за вибір загального рішення. Клас, який отримує найбільшу кількість голосів, стає результуючим класом і видається в якості результату роботи методу.

2.5 Висновки за розділом 2

Виконано постановку задачі розпізнавання мови за голосом користувача.

Виконано аналіз згорткових нейронних мереж.

Запропоновано метод розпізнавання мови за голосом користувача, який базується на використанні згорткової нейронної мережі для аналізу мовлення користувача з попереднім перетворенням аудіосигналу до спектрограм.

3 ПРОЄКТУВАННЯ СИСТЕМИ

3.1 Визначення функціональних вимог до програмного забезпечення

Програмне забезпечення, що реалізується, повинно виконувати наступні функції:

- створення спектрограм на основі заданого користувачем аудіофрагмента;
- побудова нейронної мережі;
- навчання нейронної мережі;
- збереження результатів навчання;
- завантаження нейронної мережі;
- запис мовлення користувача, що має реалізовуватися з підтримкою графічного інтерфейсу;
- розпізнавання мови на основі заданого аудіофрагменту.

Дані функції мають використовуватися окремо, оскільки безпосередньо створення моделі і її навчання може виконуватися один раз і не повторюватися до виникнення такої необхідності, тому функціональність має бути виділено в окремий застосунок. Відповідно в даному випадку, зокрема і через рідкість виконання таких дій, достатньо консольного застосунку. Ці дії має виконувати адміністратор.

Використання готової моделі для розпізнавання мови має виконуватися в іншому застосунку. Для цього потрібен стандартний застосунок з графічним інтерфейсом, оскільки ці дії може виконувати звичайний користувач.

Адміністратор може використовувати напряду або через інші функції наступні можливості:

- створення спектрограм;
- побудова нейронної мережі;
- навчання нейронної мережі;

– збереження результатів навчання.

При цьому створення спектрограми є додатковою, допоміжною, дією, яка виконується під час навчання нейронної мережі.

Діаграму прецедентів для роботи адміністратора з програмним забезпеченням наведено на рис. 3.1.

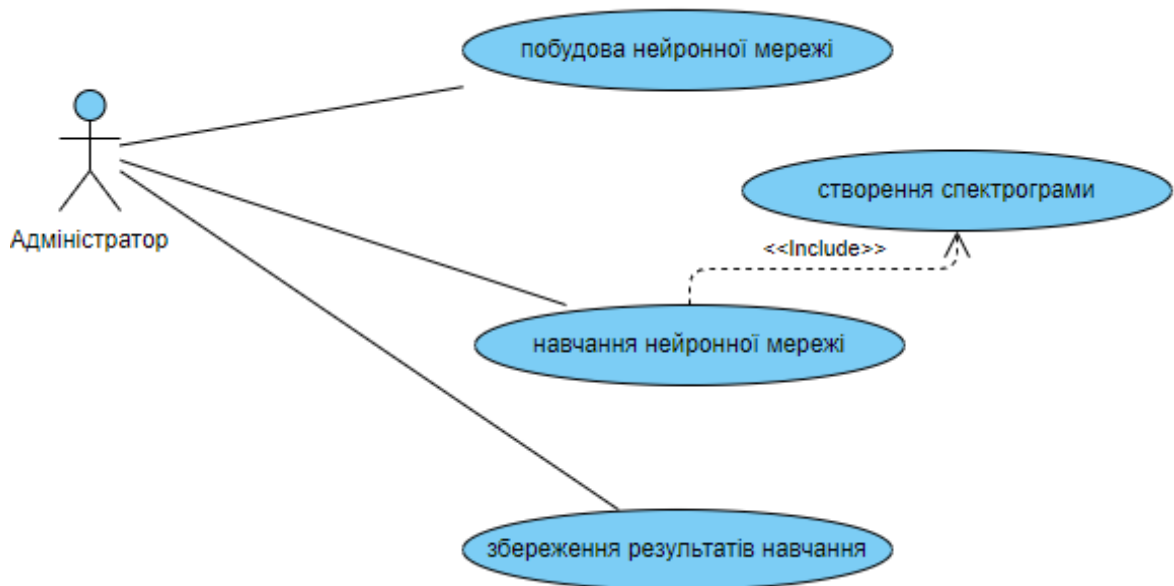


Рисунок 3.1 – Діаграма прецедентів роботи адміністратора

Користувач може використовувати напряду або через інші функції наступні можливості:

- завантаження нейронної мережі;
- запис мовлення користувача;
- розпізнавання мови.

Під завантаженням нейронної мережі мається на увазі завантаження параметрів мережі.

Діаграму прецедентів для роботи користувача з програмним забезпеченням наведено на рис. 3.2.

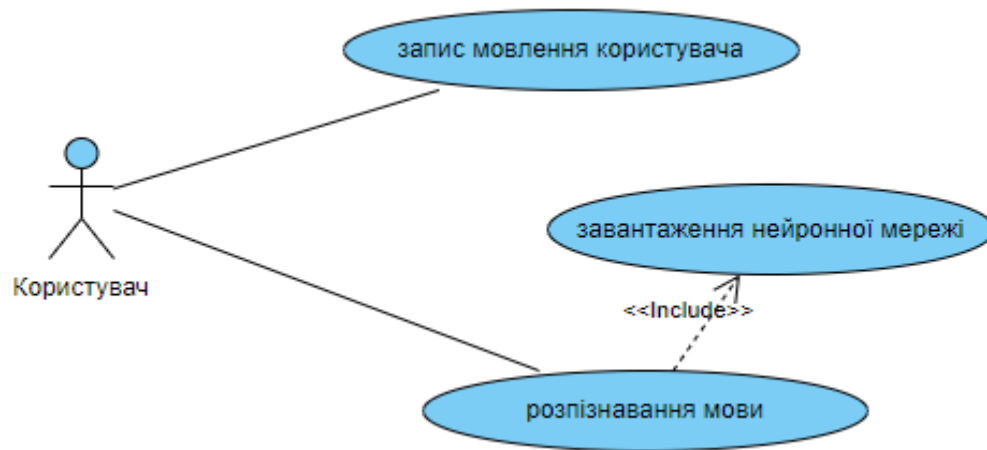


Рисунок 3.2 – Діаграма прецедентів роботи користувача

3.2 Вибір мови програмування

Мова Python на даний момент є активно використовуваною мовою, що є найпопулярнішим засобом для роботи з нейронними мережами на даний момент. Оскільки в основі функціональності, що реалізується, лежить метод, який ґрунтується на нейронних мережах, то логічно в якості базового варіанта для вибору мови програмування для реалізації програми розглянути саме цей варіант.

До переваг мови Python належать:

- розширені бібліотеки: Python завантажується з великою бібліотекою, містить код для різних цілей, таких як регулярні вирази, створення документації, тестування модулів, веб-браузери, робота з потоками даних, бази даних, електронна пошта, маніпулювання зображеннями тощо, відповідно не потрібно писати повний код для цього вручну;
- розширювана: засоби Python можна поширити на інші мови, можна написати свій код на таких мовах, як C++ або С та отримати доступ до нього з Python;
- вбудовувана: безкоштовно розширювана, мова Python також є вбудованою, можна помістити код Python у вихідний код іншої мови,

наприклад, C++, це дозволяє додати можливості сценарію до коду іншою мовою;

- підвищення продуктивності праці: простота мови та великі бібліотеки роблять програмістів продуктивнішими, ніж використовуючи такі мови, як Java та C++;

- читабельна: Python легко зрозуміти та використовувати для кодування, для визначення блоків не потрібні фігурні дужки, а відступ є обов'язковим, що додатково сприяє читабельності коду;

- об'єктно-орієнтована: ця мова підтримує як процедурні, так і об'єктно-орієнтовані парадигми програмування, хоча функції допомагають у використанні коду, класи та об'єкти дозволяють моделювати реальний світ, клас дозволяє інкапсулювати дані та функції в одне ціле;

- безкоштовна та з відкритим кодом: Python є у вільному доступі, але можна не тільки безкоштовно завантажити Python, але також можна завантажити його вихідний код, внести зміни до нього і навіть поширити його, він завантажує велику колекцію бібліотек, щоб допомогти у виконанні завдань;

- переносна: наприклад, на C++, можливо, доведеться внести деякі зміни до проєкту, якщо потрібно запустити його на іншій платформі, а на Python потрібно кодувати лише один раз, і можна запустити проєкт будь-де, однак потрібно бути обережним, щоб не включати будь-які функції, що залежать від системи;

- інтерпретована: оскільки оператори виконуються один за одним, налагодження простіше, ніж у складених мовах;

- вимагає менше кодування: практично всі завдання, виконані за допомогою Python, вимагають менше кодування порівняно з тим, коли те саме завдання виконується іншими мовами, Python має широкую стандартну бібліотеку, тому не потрібно шукати жодних сторонніх бібліотек, щоб виконати свою роботу;

– доступна: Python – безкоштовна мова, тому приватні особи, невеликі компанії чи великі організації можуть використовувати вільні наявні ресурси для створення додатків, Python популярна і широко використовується, тому він забезпечує кращу підтримку [13]-[15].

Оскільки мова програмування R використовується також для наукових обчислень, її було розглянуто в якості альтернативи. Результати порівняння мов програмування зведено до табл. 3.1.

Таблиця 3.1 – Порівняння мов програмування

Критерій порівняння	Python	R
Робота з даними	Зручна	Зручна
Машинне навчання	В повній мірі	Наявне
Готові бібліотеки	Велика кількість для різних задач	Наявні

На основі описаної логіки для розроблення програмного забезпечення було обрано мову Python.

Для реалізації роботи з моделлю було розглянуто дві бібліотеки – Tensorflow і Keras.

Щодо їх порівняння між собою можна виділити наступне.

До переваг бібліотеки Tensorflow належать:

– підвищена функціональність: хоча Keras має багато загальних функцій для глибокого навчання, Tensorflow є більш досконалим, особливо в операціях високого рівня, таких потоки, черги та налагодження;

– підвищений контроль: не завжди потрібно багато контролю, але деякі нейронні мережі можуть вимагати цього, особливо при роботі з такими операціями як ваги або градієнти [16]-[18].

До переваг бібліотеки Keras належать:

- простота використання: простий, послідовний користувацький досвід в Keras оптимізований для випадків використання, тому можна отримати чіткі, діючі відгуки щодо більшості помилок;
- модульна композиція: моделі Keras з'єднують налаштовані конструктивні блоки з невеликими обмеженнями;
- висока гнучкість і розширюваність: можна написати налаштовувані блоки для нових досліджень та створити нові шари, функції втрат, метрики та цілі моделі;
- Keras працює з наборами даних Pandas, створюючи тензори, в свою чергу TensorFlow вимагає від користувача, щоб він написав весь код, щоб створити тензор, самостійно;
- менше NumPy: NumPy складний, в коді багато зусиль витрачається на роботу з масивами NumPy, код Keras використовує його лише один раз, Keras дозволяє працювати з фреймами даних або NumPy масивами взаємозамінно;
- Keras уникає деталей низького рівня, у TensorFlow потрібно сказати це чітко;
- розширені функції TensorFlow: код Keras імпортує TensorFlow, тому можна запрограмувати функції TensorFlow безпосередньо;
- Keras також підтримує інші засоби: можна працювати з додатковими бібліотеками [16]-[18].

В результаті для роботи з нейронними мережами було обрано саме бібліотеку Keras, але в основі її роботи при цьому використовується бібліотека Tensorflow.

3.3 Структурна схема програмного забезпечення

У відповідності з логікою, яку було описано під час вибору функціональних вимог до програми, було виконано проєктування структури програми.

Програмне забезпечення побудовано на основі виділення двох складових:

- графічний інтерфейс користувача;
- підпрограма для роботи з моделлю.

Кожну складову виділено в окрему програму, взаємодія з якими відбувається окремо.

Пов'язані програми через спосіб збереження і роботи з даними. Для цього використовуються файли формату H5. Тобто база даних не створюється, оскільки структура даних, з якими доводиться працювати, достатньо проста. Вфайлах формату H5 зберігаються параметри навченої нейронної мережі для їх подальшого завантаження та роботи з ними. Саме збереження параметрів мережі у файл з розширенням H5 і мається на увазі під збереженням, а читання – відповідно під завантаженням нейронної мережі.

На рис. 3.3 представлено відповідну структуру програмного забезпечення.

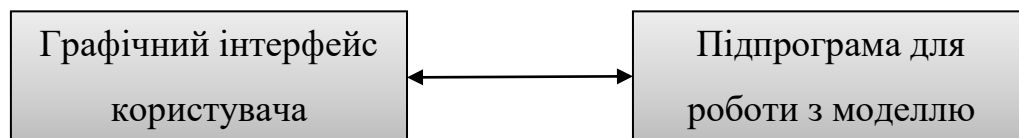


Рисунок 3.3 – Структура програми

3.4 Висновки за розділом 3

Представлено результати проєктування програмного забезпечення у вигляді структурної схеми, функціональної схеми та діаграм прецедентів для роботи користувача та адміністратора з програмним забезпеченням. Описано прийняті рішення щодо проєктування.

Програмне забезпечення вирішено розробляти мовою програмування Python, оскільки вона є зручним засобом для машинного навчання, та на

основі бібліотек Keras та Tensorflow, оскільки вони разом створюють достатньо простий у використанні інтерфейс для роботи з нейронними мережами.

4 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

4.1 Алгоритм функціонування системи

На рис. 4.1 наведено функціональну схему роботи програми.



Рисунок 4.1 – Схема функціонування програми

Робота програми, що представляє графічний інтерфейс користувача, ґрунтується на циклі, який повторюється до того моменту, як користувач забажає завершити роботу з програмою.

В циклі спочатку користувач записує повідомлення, потім реалізується завантаження раніше навченої моделі, а далі виконується розпізнавання.

4.2 Основні рішення щодо реалізації компонентів системи

Файл `model.py` організовує роботу з моделлю. Даний файл реалізує звернення до скрипта створення моделі та її навчання.

За створення моделі відповідає функція `create` з файлу `createmodel.py`.

Робота даної функції розпочинається зі створення моделі на основі стеку шарів:

```
nn= Sequential()
```

Далі додаються три згорткові шари, кожен з яких визначається наступним чином, при цьому фільтр визначається в даному прикладі розміром 3x3, задаються страйди, тобто крок переміщення фільтра, за допомогою параметру `strides`:

```
nn.add(Conv2D(  
8,  
(3, 3),  
strides=(1, 1),  
padding='same',  
l2reg=regularizers.l2(0.001),  
params=params))
```

В якості функції активації визначено функцію ReLU:

```
nn.add(Activation('relu'))
```

Виконання пулінгу забезпечується на основі функції `maxpool`:

```
nn.add(MaxPooling2D(pool_size=(3, 3), strides =(2, 2), padding ='same'))
```

Визначення повнозв'язного шару реалізується за допомогою наступного виклику, забезпечується також виконання дропауту:

```
nn.add(Dense(
    16,
    activation='relu',
    l2reg=regularizers.l2(0.001)))
nn.add(Dropout(0.5))
nn.add(Dense(len(RecogL)))
```

Безпосередньо навчання моделі реалізується через основну функцію, де керування відбувається за допомогою ключів.

Збереження моделі після кожної епохи виконується наступним чином:

```
ModelCheckpoint(
    'info.h5',
    monitor='val_loss',
    verbose=0,
    save_best_only=True,
    mode='min')
```

Умови завершення навчання визначаються за допомогою функції EarlyStopping, при цьому завершення навчання виконується тоді, коли протягом трьох епох (параметр patience) зміна функції втрат (параметр min_delta) дорівнює 0:

```
treas = EarlyStopping(
    monitor='val_loss',
    min_delta=0,
    patience=3,
```

```
verbose=0,
mode='auto')
```

Оцінювання результатів розпізнавання виконується на тестових даних (attrst, outtst):

```
classes = nn.evaluate(attrst, outtst, verbose=0)
```

Саме навчання виконується протягом 400 епох, в якості навчальної вибірки використовується набір вхідних і вихідних даних (attrtr, outtr), в якості валідаційної вибірки використовується пара (attrst, outtst):

```
nn.fit(
attrtr,
outtr,
epochs=400,
callbacks=[mc, treas],
verbose=1,
validation_data=(attrst, outtst),
smpllen=60)
```

Функція `spectr` виконує створення спектрограм на основі переданого аудіофайлу або набору їх.

Функція `createdata` безпосередньо забезпечує розрахунки для побудови спектрограми.

Завантаження навченої моделі реалізується наступним чином:

```
load_model('info.h5')
```

4.3 Висновки за розділом 4

Описано алгоритм функціонування програми з представленням схеми функціонування.

Описано процес створення та навчання нейронної мережі, а також роботи з нею за допомогою бібліотеки Keras.

5 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

5.1 Призначення програми

Програмне забезпечення призначене для розпізнавання мови за голосом користувача.

Для цього надаються засоби для роботи з моделлю, що охоплюють побудову нейронної мережі, навчання нейронної мережі, збереження результатів навчання, створення спектрограм, та засоби роботи з користувачем, що надають йому можливість записувати його мовлення, розпізнавати мову та забезпечують для цього завантаження нейронної мережі.

5.2 Умови виконання програми

Для експлуатації програмного забезпечення необхідні наступні технічні засоби:

- процесор з 4 ядрами, при цьому частота кожного має бути не менше 3 ГГц;
- оперативна пам'ять – не менше 2 Гб;
- жорсткий диск – вільний простір не менше 1 Гб.

Дані вимоги висувуються для необхідності навчання моделі. Якщо модель вже є готова і навчання не передбачається, то вимоги можуть бути знижені.

Для роботи програмного забезпечення необхідно на серверній частині інсталиувати:

- інтерпретатор мови програмування Python версії 3;
- пакети Keras, Tensorflow та NumPy.

5.3 Виконання програми

Програмне забезпечення складається з двох програмних застосунків: консольного застосунку та застосунку з графічним інтерфейсом.

Для розв'язання задачі спочатку повинна бути підготовлена модель. Тому спочатку потрібно працювати з консольним застосунком. Його реалізовано через файл `model.py`. Отримані результати під час роботи виводяться в файл `info.txt`.

Запуск даного застосунку виконується наступним чином за замовчуванням:

```
pythonmodel.py
```

Застосунок підтримує роботу через ключі. Зокрема передбачаються наступні ключі:

- `train` – навчання моделі;
- `test` – тестування.

Для роботи з готовою моделлю передбачено застосунок з графічним інтерфейсом користувача. Його роботу організовано через файл `user.py`. Відповідно для його запуску потрібно виконати команду:

```
pythonuser.py
```

Після цього відкривається вікно з графічним інтерфейсом користувача (рис. 5.1).

У даному вікні представлено дві кнопки, що дозволяють керувати роботою застосунку:

- кнопка «Записати», що дозволяє розпочати процес запису мовлення користувача в файл;

– кнопка «Зупинити», що відповідно дозволяє завершити даний процес і зберегти отримані результати в файл.

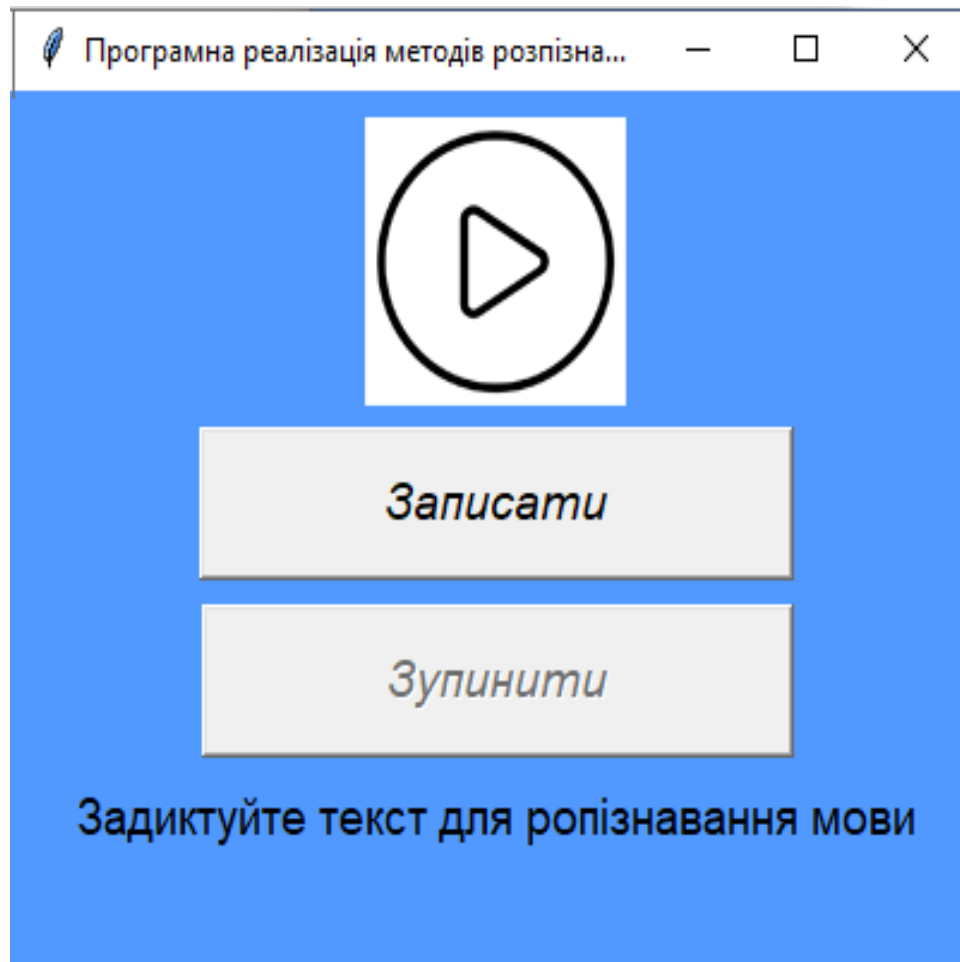


Рисунок 5.1 – Початкова форма програмного застосунку

Логіка роботи організована у відповідності з задачею (2.1), на розв’язання якої спрямовано розроблений метод і програмне забезпечення. Тобто спочатку має бути створено аудіофрагмент, який потім програма має віднести до однієї з природніх мов. Для створення такого фрагменту достатньо визначити його початок і кінець, і зберегти отримані дані для подальшої класифікації.

Спочатку в програмі активною є тільки кнопка «Записати», тобто натиснути на кнопку «Зупинити» не вдасться. При цьому іконка зверху відображає можливість розпочати процес запису. Повідомлення внизу відображає пропозицію розпочати запис.

Після того, як процес запису розпочався, іконка змінює свій вигляд на такий, що відображає можливість завершити запис (рис. 5.2). Повідомлення внизу змінюється на інформування про те, що процес запису триває. Кнопка «Зупинити» стає активною, а кнопка «Записати» навпаки неактивною, тобто одразу розпочати запис іншого фрагмента, допоки не завершився запис першого, неможливо.

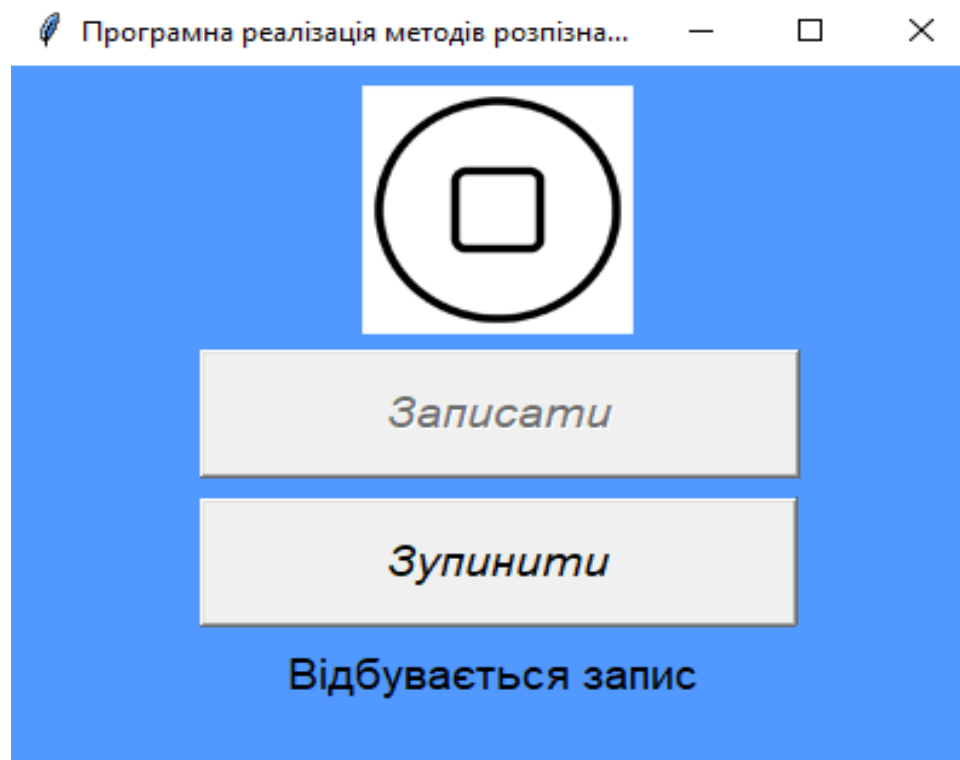


Рисунок 5.2 – Процес запису аудіоповідомлення

Як тільки користувач натисне на кнопку «Зупинити», то знову ж відбудеться перехід до першого описаного вигляду. Активною буде тільки кнопка «Записати». Знизу буде відображено повідомлення про те, яку саме мову було розпізнано в записаному повідомленні (рис. 5.3). Це і є результатом класифікації.

Розпізнавання в програмі, тобто на практиці, реалізовано для англійської та німецької мов. Для розширення бази мов необхідно отримати відповідну вибірку даних, на якій має бути реалізовано навчання моделі.

Потенційно може підтримуватися будь-яка припродня мова, але за наявності відповідної вибірки даних.

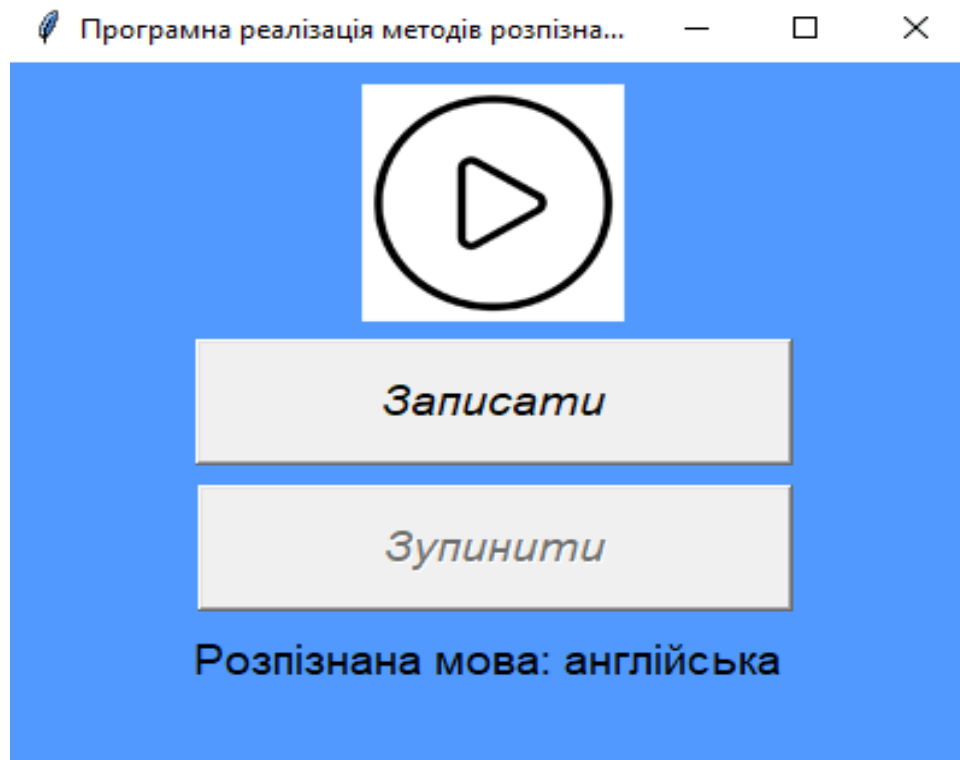


Рисунок 5.3 – Відображення результатів розпізнавання

6 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

Мета роботи полягала в розробленні програмного забезпечення розпізнавання мови за голосом користувача.

Використання запропонованого методу дозволить автоматизувати даний етап під час створення інтелектуального помічника, який зможе в результаті обирати відповідну схему оброблення природної мови на основі результатів, отриманих за допомогою відповідного розробленого модуля. Подібний модуль коштує 600 гривень на місяць.

Розроблене програмне забезпечення як відповідний застосунок або окремий вбудований модуль заплановано продавати приблизно за суму, еквівалентну 426 гривням на місяць, що є придатним, адже розробка має наукову новизну. Кількість очікуваних покупців – 20.

6.1 Планування розробки програмного продукту

Основні роботи, необхідні для розробки програмного продукту, і їх трудомісткість представлені в табл. 6.1.

Необхідні початкові дані:

- час на розробку – до 2 місяців;
- програміст працює – до 1,8 місяці (40днів) із заробітною платою 11000 грн./міс.;
- консультант працює – до 0,25 місяця (5 днів) із заробітною платою 7000 грн./місяць.

Завершуючим етапом планування розробки програмного продукту є звідний стрічковий графік розробки програмного продукту, побудований за даними табл. 6.1 [19].

Таблиця 6.1 – Характеристика робіт по розробці програми

Найменування роботи	Трудомісткість		Виконавці		Трива- лість
	люд.- дн.	% до підсумку	спеціаль- ність	кіл. люд.	днів
1. Дослідження вимог до проєктованого програмного продукту	10	22.22	Програміст	2	5
			Консультант		
2. Розробка технічного завдання	5	11.11	Програміст	1	5
3. Проєктування загальної структури програмного забезпечення	5	11.11	Програміст	1	5
4. Розробка схеми функціонування програмного забезпечення	5	11.11	Програміст	1	5
5. Створення програмного коду	14	31.11	Програміст	1	14
6. Тестування та налагодження програмного продукту	3	6.66	Програміст	1	3
7. Складання програмної документації	3	6.66	Програміст	1	3
Всього	45	100.0			40

Лінійний графік Ганта представлений в табл. 6.2.

Чорним кольором позначено роботу програміста, сірим – консультанта.

Таблиця 6.2 – Лінійний графік Гантарозробки програмного продукту

Найменування роботи	Календарний період, дні							
	07.09 – 11.09	14.09 – 18.09	21.09 – 25.09	28.09 – 02.10	05.10 – 09.10	12.09 – 16.10	19.10 – 23.10	26.10 – 30.10
1. Дослідження і аналіз вимог до проєктованого продукту	■							
2. Розробка технічного завдання		■						
3. Проєктування загальної структури програмного забезпечення			■					
4. Розробка схеми функціонування програми				■				
5. Створення програмного коду					■	■	■	
6. Тестування та налагодження програмного продукту							■	
7. Складання програмної документації								■

6.2 Розрахунок основної заробітної плати

Витрати по цій статті складаються з планового фонду заробітної плати усіх категорій працівників, зайнятих в розробці програмного продукту, розрахунок заробітної плати ведеться на основі відомостей про трудомісткість (табл.6.1)[19].

Результати розрахунків зведені в табл. 6.3.

Таблиця 6.3 – Розрахунок основної заробітної плати

Посада виконавця	Кількість, люд.	Місячний оклад	Кількість місяців роботи	Премії і доплати, грн.	Середня місячна зарплата, грн.	Сумарна зарплата, грн.
Програміст	1	11000	1,8	–	12222	22000
Консультант	1	7000	0,25	–	7000	1750
Всього	2					23750

6.3 Розрахунок додаткової заробітної плати

Для розрахунку додаткової заробітної плати проводяться окремі обчислення, вони базуються на попередніх, встановлюючи рівень у 10 % основної зарплати [19]. Отримані результати розрахунку заробітної плати приведені в табл. 6.4.

Таблиця 6.4 – Основна і додаткова заробітна плата

Посада виконавця	Додаткова заробітна плата, грн.	Сума основної і додат. заробітної плати, грн.
Програміст	2200	24200
Консультант	175	1925
Разом	2375	26125

6.4 Відрахування на єдиний соціальний внесок

Розрахунок відрахувань до соціального внеску виконується на основі формули:

$$\text{ВІД} = (\text{ЗП}_{\text{осн}} + \text{ЗП}_{\text{дод}}) \cdot 0,22, \quad (6.1)$$

де $ЗП_{\text{осн}}$ – основна заробітна плата, грн.;

$ЗП_{\text{дод}}$ – додаткова заробітна плата, грн. [19].

Результат:

$$ВІД = (23750 + 2375) \cdot 0,22 = 5747,5 \text{ грн.}$$

6.5 Визначення витрат на матеріали

Матеріали, які знадобилися для розробки, представлені в табл.6.5.

Таблиця 6.5 – Витратні матеріали

Матеріал	Кількість	Вартість, грн.
Папір формату А4	150 аркушів	40
CD-RW диск	1 шт.	5,0
Тонер для принтера	1 шт.	120

Витрати на матеріали розраховуються за формулою (6.2):

$$M = \sum_{i=1}^n (C_i \cdot N_i \cdot (1 + K_{\text{т.з.}}) - C_{io} \cdot N_{io}), \quad (6.2)$$

де M – витрати на матеріали, покупні напівфабрикати і комплектуючі вироби, грн.;

$K_{\text{т.з.}}$ – коефіцієнт, що враховує транспортно-заготівельні витрати;

C_i – ціна i -го найменування матеріалу, напівфабрикату, комплектуючого, грн.;

N_i – ціна в i -му матеріалі, напівфабрикаті, комплектуючого;

C_{io} – ціна поворотних відходів i -го найменування матеріалу, грн.;

N_{io} – кількість зворотних відходів i -го найменування;

n – кількість найменувань матеріалів, напівфабрикатів, комплектуючих [19].

Розрахуємо витрати на матеріали:

$$M = (40 + 5,0 + 120) \cdot (1 + 0,05) = 173,25 \text{ грн.}$$

Витрати на матеріали дорівнюють 173,25 грн.

6.6 Витрати на спеціальне устаткування

Витрати на оплату машинного часу електронно-обчислювальної машини (ЕОМ) для написання і відладки цієї програми визначають по формулі (6.3):

$$C_{\text{мо}} = P_{\text{екс}} \cdot t_{\text{мо}}, \quad (6.3)$$

де $C_{\text{мо}}$ – витрати на оплату машинного часу, грн.;

$P_{\text{екс}}$ – експлуатаційні витрати на одну годину машинного часу цієї цифрової ЕОМ, грн./машино-год.;

$t_{\text{мо}}$ – машинний час цифрової ЕОМ для написання і налагодження даного програмного продукту, машино-год. [19].

Визначимо кошторис витрат на утримання і експлуатацію устаткування.

Амортизаційні відрахування визначають по формулі (6.4):

$$A = \Phi_6 \cdot N_a, \quad (6.4)$$

де Φ_6 – балансова вартість обчислювальної техніки, грн.;

N_a – норма амортизаційних відрахувань на повне відновлення обчислювальної техніки, % [19].

Розрахуємо амортизаційні відрахування:

$$A = 12000 \cdot \frac{25}{100} = 3000 \text{ грн.}$$

Балансова вартість машини визначається як вартість машини плюс 5% на транспортні витрати і монтаж.

Витрати на електроенергію визначаються за формулою (6.5):

$$C_v = N_n \cdot \Phi_{\text{еф}} \cdot K_{\text{зч}} \cdot K_{\text{зп}} \cdot C_{\text{ел.в}}, \quad (6.5)$$

де N_n – номінальна потужність ЕОМ, кВт;

$\Phi_{\text{еф}}$ – річний ефективний фонд часу роботи ЕОМ, машино–год;

$K_{\text{зч}}$ – середній коефіцієнт завантаження по часу;

$K_{\text{зп}}$ – коефіцієнт завантаження по потужності;

$C_{\text{ел.в}}$ – ціна одного кВт·год електроенергії, грн./(кВт·год) [19].

Відповідно:

$$C_v = 0,2 \cdot 1800 \cdot 0,75 \cdot 0,7 \cdot 2,11 = 400,31 \text{ грн.}$$

Зарплата обслуговуючого персоналу розраховується по формулі (6.6):

$$ЗП_{\text{обсл}} = ФЗП_r \cdot (1 + K_{\text{відр}}) \cdot \frac{t_{\text{обсл}}}{\Phi_{\text{еф.обсл}}}, \quad (6.6)$$

де $ФЗП_r$ – річний фонд заробітної плати (основної та додаткової) обслуговуючих робітників, грн;

$K_{\text{відр}}$ – коефіцієнт, що враховує відрахування на соціальне страхування й в інші фонди;

$t_{\text{обсл}}$ – час протягом року, необхідне на технічне обслуговування ЕОМ, год/рік;

$\Phi_{\text{еф.обсл}}$ – річний ефективний фонд часу обслуговуючого персоналу, год/рік [19].

Машину обслуговує один електронщик з основною заробітною платою 6400 грн./міс. Відповідно за 12 місяців річний фонд заробітної обслуговуючого робітника буде складати:

$$\Phi ЗП_p = 6400 \cdot 12 = 76800 \text{ грн.}$$

При цьому $K_{\text{відр}} = 0,22$.

Час, який протягом року цей робітник має витратити на технічне обслуговування ЕОМ для підтримки роботи програми, протягом року складає 5 годин, тобто $t_{\text{обсл}} = 5 \text{ год./рік}$.

Річний ефективний фонд часу обслуговуючого персоналу розраховується на основі формули (6.7):

$$\Phi_{\text{еф.обсл.}} = (D_p \cdot F_c - r \cdot m) \cdot K, \quad (6.7)$$

$$K = (100 - C_n)/100,$$

де D_p – кількість робочих днів в році;

F_c – тривалість робочої зміни;

r – кількість скорочених годин в передсвяткові дні;

m – пересвяткові дні;

C_n – втрати від номінального робочого фонду через невихід на роботу [19].

У 2020 році кількість робочих днів складає 251 день, а кількість передсвяткових днів дорівнює 6. Кількість скорочених годин в передсвяткові дні дорівнює 1 годині. Тривалість робочої зміни дорівнює 8 годинам.

C_n дорівнює 12,5 %, виходячи з наступного: 9 % – на відпуску, 2 % – на лікування, 0.5 % – на виконання сіуспільних обов'язків, 1 % – на учбові відпустки.

Тоді на основі формули (6.7) розрахуємо:

$$\Phi_{\text{эф.обсл}} = (251 \cdot 8 - 1 \cdot 6) \cdot 0,875 = 1751,75 \text{ год.}$$

Відповідно зарплата обслуговуючого персоналу буде дорівнювати:

$$ЗП_{\text{обсл}} = 76800 \cdot (1 + 0,22) \cdot 5 / 1751,75 = 267,44 \text{ грн.}$$

Сума витрат по статті "Поточний ремонт устаткування" дорівнює 3% від балансової вартості устаткування, що дорівнює 360,00 грн.

Інші витрати складають 5% від суми попередніх.

Зведемо отримані дані в табл. 6.6.

Таблиця 6.6 – Кошторис витрат на зміст і експлуатацію устаткування

Найменування статей витрат	Сума, грн.
Амортизація устаткування	3000,00
Експлуатація устаткування	400,31
Зарплата обслуговуючого персоналу з відрахуваннями	267,44
Поточний ремонт устаткування	360,00
Інші витрати	201,41
Всього	4229,16

Експлуатаційні витрати на одну годину машинного часу складають:

$$P_{\text{екс}} = 4229,16 / 1800 = 2,35 \text{ грн./год.}$$

Розробка і відладка програмного продукту займає 320 годин машинного часу (40 днів по 8 робочих годин), тобто $t_{\text{мо}} = 320$ год.

Витрати на оплату машинного часу склали:

$$C_{\text{мо}} = 2,35 \cdot 320 = 752,00 \text{ грн.}$$

6.7 Накладні витрати

В даному випадку накладні витрати складають 40% до основної заробітної плати розробників, що складає 9500 грн.

Отримані результати зведено до табл. 6.7.

Таблиця 6.7 – Калькуляція кошторисної вартості робіт по розробці програми і ціна програмного продукту

Найменування статей витрат	Сума, грн.	Питома вага до підсумку, %
1. Основна заробітна плата	23750,00	56,15
2. Додаткова заробітна плата	2375,00	5,61
3. Соціальні відрахування	5747,50	13,59
4. Витрати на матеріали	173,25	0,41
5. Витрати на оплату машинного часу	752,00	1,78
6. Накладні витрати	9500,00	22,46
7. Разом витрати	42297,25	100,00
8. Прибуток	12689,18	-
9. Всього	54986,43	-
10. ПДВ (20%)	10997,29	-
Всього	65983,72	-

6.8 Оцінка економічної ефективності

Планується продаж 20 копій по 426 гривень за копію на місяць. Відповідно очікується отримати надходження від продажів програми в розмірі 102 240 гривень за рік.

Оскільки фактично може бути отримано інші результати продажів, то це враховується шляхом визначення рівня ризику. Ризик встановлено на рівні 50%. Відповідно в такому випадку від продажів надійде 51 120 гривень.

Витрати окупляться за 1,29 року. Отримане значення менше за нормативний термін окупності, який визначається для соціальної сфери на рівні 6,67 року.

Розробка даної програми є економічно доцільною.

7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

7.1 Аналіз потенційних небезпек

Під час проведення дослідження та програмної реалізації методів розпізнавання мови за голосом користувача основні роботи по обробці отриманої інформації виконуються за допомогою персонального комп'ютеру (ПК).

При роботі з комп'ютерною технікою можуть бути виявленні наступні небезпечні чинники:

- мерехтіння екрану або періодична зміна його яскравості;
- підвищена зовнішня освітленість екрану;
- нестабільність зображення на екрані;
- підвищений рівень статичної електрики;
- висока температура, вологість і рухливість повітря робочої зони;
- підвищене значення напруги в електричному ланцюзі, замикання яке може пройти через тіло людини;
- пожежа [20].

7.2 Заходи забезпечення безпеки

Офісні центри відносяться до приміщень без підвищеної небезпеки: при проєктуванні систем електропостачання, монтажу силового електрообладнання та електричного освітлення будівель та споруд для комп'ютерної техніки необхідно дотримуватись вимог Правил улаштування електроустановок [20].

Згідно ДСТУ Б В.2.5-82:2016 «Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом» комп'ютерна техніка, периферійні пристрої ПК та устаткування для обслуговування, ремонту та налагодження ПК, інше устаткування, електропроводка та кабелі за виконанням та ступенем захисту відповідають

класу згідно Правил улаштування електроустановок, за якого повинні мати апаратуру захисту від струму короткого замикання та інших аварійних режимів[20].

Відеотермінали, комп'ютерна техніка, спеціальні периферійні пристрої ПК та устаткування для обслуговування, ремонту та налагодження ПК відповідають вимогам діючих в Україні стандартів, нормативних актів з охорони праці і правил: відеотермінали, комп'ютерна техніка, спеціальні периферійні пристрої ПК закордонного виробництва додатково відповідають вимогам національних стандартів держав-виробників і мати відповідну позначку на корпусі, в паспорті або іншій експлуатаційній документації [20].

За способом захисту людини від ураження електричним струмом відеотермінали, комп'ютерна техніка та обладнання для обслуговування, ремонту та налагодження ПК відповідають I класу захисту або заземлені відповідно до Нормативного акту про охорону праці 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроям» [21].

7.3 Заходи з виробничої санітарії та гігієни праці

Для підтримки нормованих значень концентрації негативних і позитивних іонів передбачена штучна іонізація, у приміщеннях з дисплеями проводиться щоденне вологе прибирання і регулярні провітрювання протягом робочої зміни, видалення пилу з екранів дисплеїв проводиться не рідше 1 разу за зміну, у періоди зміни, коли робота не виконується, дисплеї вимкнені [22].

Висока чи низька температура повітря в приміщенні з ПК негативно впливає на функціональний стан користувача: недостатня вологість в приміщенні призводить до надмірного висихання слизових оболонок та до

нагромадження зарядів статичної електрики, що утворюється в процесі роботи ПК [23].

Разом з тим недопустима вологість повітря більше 75%: на робочих місцях користувачів ПК параметри мікроклімату мають відповідати вимогам ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» [23], Державних санітарних норм і правил 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [24].

У залах з обчислювальною технікою і інших виробничих приміщеннях при виконанні робіт операторського типу, пов'язаних з нервово-емоційним напруженням, дотримуються оптимальних величин температури повітря 22-24°C при його відносній вологості 40-60% і швидкості руху повітря не більше 0,1 м/с відповідно до Державних будівельних норм В.2.5-67:2013 «Опалення, вентиляція, кондиціонування» [25].

Згідно ДБН В.2.5-28-2018 «Природне і штучне освітлення» для штучного освітлення приміщень, в яких використовують дисплеї, запроєктована система комбінованого освітлення, передбачено пристрій робочого і евакуаційного освітлення [26].

При влаштуванні місцевого освітлення застосовуються лампи на пересувних шарнірах, світло падає по ліву сторону, потужність ламп, яка рекомендується – не менше 40-70 Вт, при цьому одне місцеве освітлення неприпустимо [25].

Нижче наведено розрахунок загального освітлення штучним світлом у обчислювальному центрі розміром 10 x 6 x 3,5 м, III розряду зорових робіт, потрібно створити освітленість $E_n = 300$ лк, коефіцієнт відбиття стелі $\rho_{cm} = 70\%$ і стін $\rho_c = 50\%$, для освітлення використовуються люмінесцентні лампи типу ЛБ 40 [26].

Знаходимо індекс приміщення по формулі (7.1):

$$i = \frac{A \cdot B}{H_p (A + B)}, \quad (7.1)$$

де A, B – розміри обчислювального центру (довжина, ширина), м;

H_p – висота світильників над робочою областю, м.

Обчислимо:

$$i = \frac{10 \cdot 6}{(3,5 - 0,8) \cdot (10 + 6)} \approx 1,4.$$

Приймаємо коефіцієнт запасу $k_z = 1,6$ і коефіцієнт нерівномірності освітлення $z = 1,1$

При індексі $i = 1,4$ отримаємо $\eta = 51\%$.

Визначаємо необхідний світловий потік ламп в кожному ряді ($N_p = 2$) приміщення з площею S по формулі (7.2):

$$\Phi_p = \frac{E_n \cdot A \cdot B \cdot k_z \cdot z}{\eta}, \quad (7.2)$$

де E_n – рівень нормованого загального освітлення, лк;

A – довжина приміщення, м;

B – ширина приміщення, м;

k_z – коефіцієнт запасу (для люмінесцентних ламп $k_z = 1,4 \dots 1,7$);

z – коефіцієнт нерівномірності освітленості (відношення середньої до мінімальної освітленості), як правило дорівнює (для люмінесцентних ламп $z = 1,1$);

η – коефіцієнт використання світлового потоку.

Обчислимо світловий потік ламп в кожному ряді:

$$\Phi_p = \frac{300 \cdot 60 \cdot 1,1 \cdot 1,6}{2 \cdot 0,51} \approx 31058 \text{ лм.}$$

Якщо в світильники встановити по дві лампи ЛБ 40 ($n = 2$) потужністю 40 Вт і світловим потоком $\Phi_{\text{л}} = 3200$ лм, то необхідне число світильників в ряді складе:

$$N = \frac{\Phi_p}{n \cdot \Phi_{\text{л}}}. \quad (7.3)$$

Обчислимо кількість світильників в ряді:

$$N = \frac{31058}{2 \cdot 3200} \approx 5 \text{ світильників.}$$

Робочі місця програмістів обчислювальних машин не мають граничних рівнів звуку вище 50 дБ, а в приміщеннях для розміщення гучних агрегатів обчислювальних машин вище 75 дБ, при виконанні робіт з вироблення концепцій нових програм рівні звуку не перевищують 40 дБ; при виконанні операторської роботи – 65 дБ згідно Державних санітарних норм 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку [27].

Ергономічна організація робочого місця, оснащеного дисплеєм, враховує специфіку конкретного робочого місця і особливості виконуваних користувачами завдань із застосуванням відео-терміналу комп'ютера [27].

Правильна організація робочих місць сприяє: усуненню загального дискомфорту, зменшенню втомлюваності працівника, підвищенню його продуктивності, проведені дослідження показують, що при раціональній організації робочих місць продуктивність, праці зростає на 15–25% [26].

Площа, виділена для одного робочого місця з візуальними дисплейними терміналами або ПК, повинна складати не менше 6 м², а об'єм – не менше 20 м³, при розміщенні робочих місць необхідно дотримуватись таких вимог:

- робочі місця з візуальними дисплейними терміналами розміщуються на відстані не менше 1,2 м від стін зі світловими прорізами;
- відстань між бічною поверхнею одного з відео терміналів має бути не меншою за 1,2 м;
- відстань від екрана дисплея до оператора (при розмірі екрану по діагоналі): 14 " -15 " – 600-700 мм; 17 ' – 700-800 мм; 19 ' – 800-900 мм; 21 " – 900-1000 мм;
- відстань між тильною поверхнею одного відеотермінала та екраном іншого не повинна бути меншою 2,5 м;
- прохід між рядами робочих місць має бути не меншим 1 м [27].

Вимоги до конструкції робочого столу, стільця, підставки для ніг на робочих місцях з візуальними дисплейними терміналами визначаються Нормативно-правовим актом з охорони праці 0.00-7.15-18 [21].

Конструкція робочого столу має відповідати сучасним вимогам ергономіки і забезпечувати оптимальне розміщення на робочій поверхні всього обладнання та пристосувань, що використовуються: висота робочої поверхні столу для ВДТ відповідно має бути в межах 680 – 800 мм, а ширина і глибина – забезпечувати можливість виконання операцій у зоні досяжності моторного поля, рекомендовані розміри столу: висота – 725 мм, ширина – 600 – 1400 мм, глибина – 800 – 1000 мм, робочий стіл відповідно повинен мати простір для ніг висотою не менше 600 мм, шириною не менше 500 мм, глибиною на рівні колін не менше 450 мм, на рівні витягнутої ноги – не менше 650 мм [26].

У приміщеннях, обладнаних відеотерміналами, стіни пофарбовані фарбами пастельних тонів з коефіцієнтом відображення 40-60%, коефіцієнт відбиття стелі 70-80%, підлоги 30-50%, поверхня підлоги рівна, неслизька, з антистатичними властивостями [27].

У разі виникнення надмірного зорового навантаження, що супроводжується погіршенням самопочуття, незважаючи на хорошу ергономічну організацію роботи і дотримання режимних моментів,

проводиться індивідуальний підхід в обмеженні часу роботи, корекція тривалості перерв для відпочинку або зміна діяльності, не пов'язаної з комп'ютером[28].

Для підтримання високого рівня працездатності та збереження здоров'я працівників щоденно виконується комплекс фізичних вправ, що сприяють функціональному стану [28].

7.4 Заходи з пожежної безпеки

В приміщенні обчислювального центру джерелом загоряння можуть бути електронні схеми ЕОМ, прилади для технічного обслуговування, електроустаткування, кондиціонери повітря, де за рахунок різних порушень відбувається перегрів тих чи інших елементів, електричні іскри, дуга, що можуть визвати загоряння горючих матеріалів, дерев'яні та пластикові стільці та столи, папір, перевантаження мережі живлення, людські чинники тощо [29].

Заходи щодо протипожежного захисту розділяються на:

- організаційні заходи: навчання робітників та службовців правилам пожежної безпеки, організація пожежної охорони, проведення бесід, лекцій, плакатів і т.п.;

- технічні заходи передбачають дотримання протипожежних правил і норм при устрої систем опалення, вентиляції, кондиціюванні повітря, блисковкозахисті при спорудженні будинків, установці технологічного устаткування й ін.;

- експлуатаційні заходи передбачають правильну експлуатацію систем опалення, вентиляції й кондиціювання повітря, блисковкозахисту технологічних машин і устаткування, правильне утримання будинків і територій і т.п.;

- режимні заходи передбачають заборону або обмеження застосування відкритого вогню в вогнебезпечних місцях, паління в

невстановлених місцях, обов'язкове дотримання норм і правил при роботі з вогнебезпечними й вибухонебезпечними речовинами [29]-[30].

Обчислювальний центр відноситься до категорії «П-Па» згідно ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою», згідно Державних будівельних норм В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги», а клас можливої пожежі визначається, як «А» [29]-[30].

Прийнято також наступні рішення:

- для ліквідації пожежі у початковій стадії її розвитку силами персоналу об'єктів застосовуються первинні засоби пожежогасіння: вогнегасники, пожежний інвентар (покривала з негорючого теплоізоляційного полотна, ящики з піском тощо), системи автоматичного пожежогасіння;

- так як приміщення (дослідницької лабораторії, конструкторського бюро, тощо) що обладнане ПК з візуальними дисплейними терміналами має площу 60 м², тому відповідно до вимог п. 3.8 розділу «Типові норми належності вогнегасників»;

- ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги» [28] для гасіння електроустановок, що знаходяться під напругою, передбачені вуглекислотні вогнегасники типу ВВК-3,5 у кількості 3 штук (з розрахунку один вогнегасник с величиною заряду вогнегасної речовини 3 кг. і більше, на 20 м² площі приміщення), відстань між вогнегасниками та місцями можливих загорянь не перевищує 15 м [29].

Під час прийняття на роботу і в процесі праці усі працівники обчислювального центру повинні проходити протипожежний інструктаж, евакуаційні виходи і проходи утримуються постійно вільними та легкодоступними:

- приміщення обчислювального центру обладнано автоматичними пожежними сповіщувачами, що реагують на підвищення температури, дим, полум'я;

- для відводу надлишкового тепла від ЕОМ задіяні кондиціонування повітря і система вентиляції, що обладнана пристроєм, який забезпечує автоматичне відключення вентиляції, в разі виникнення пожежі;

- напруга до електроустановок подається по кабельним мережам, що прокладені під технологічною з'ємною підлогою з рівнем вогнестійкості не менше 0,5 год. [30].

7.5 Заходи з забезпечення безпеки у надзвичайних ситуаціях

Ядерні вибухи в атмосфері й більш високих шарах призводять до виникнення потужних електромагнітних полів з довжиною хвиль від 1 до 1000 м і більше, ці поля називають електромагнітним імпульсом, електромагнітний імпульс виникає при ядерному вибусі у воєнний час, у мирний час – при випробуванні ядерної зброї і катастрофах в атмосфері і космосі [31].

Особливо чутливими до впливу електромагнітного імпульсу є 6 основних груп об'єктів і систем, серед них можна окремо виділити наступні:

- системи виробництва, перетворення і накопичення енергії: електростанції, генератори постійного і змінного струму, трансформатори, комутатори і розподільні пристрої, електричні батареї, паливні, сонячні й термоелементи;

- системи радіозв'язку, передачі, зберігання і накопичення інформації: антени, хвилеводи, коаксильні кабелі, електронні прилади, радіопередавачі, радіоприймачі, телефонні апарати, телеграфні установки, заземлені кабелі й проводи [31].

Сучасний рівень знань про природу і властивості електромагнітного імпульсу дає можливість розробити захист від нього і впровадити заходи захисту до яких входять схеми, стійкі до електромагнітної інтерференції,

радіоелектронні елементи стійкі до електромагнітного імпульсу, екранування окремих пристроїв або цілих електронних систем [31].

ВИСНОВКИ

У роботі розглянуто проблему розпізнавання мови за голосом користувача. Виконано огляд методів розпізнавання мовлення, серед яких зокрема: метод ковзного вікна та приховані марківські моделі, нейронні мережі, метод моделей заповнювачів.

Виконано огляд архітектур нейронних мереж та архітектури згорткових нейронних мереж зокрема. Запропоновано використовувати її для роботи з мовленням.

Представлено результати проєктування програмного забезпечення у вигляді структурної схеми, функціональної схеми та діаграм прецедентів для роботи користувача та адміністратора з програмним забезпеченням.

Програмне забезпечення розроблено мовою програмування Python на основі бібліотек Keras та Tensorflow.

Програмне забезпечення складається з програмного застосунку з графічним інтерфейсом, який дозволяє записувати мовлення користувача та працювати з готовою моделлю, та окремого консольного застосунку, який дозволяє керувати нейронною мережею, зокрема створювати модель, виконувати її навчання та зберегати результати.

Наукова новизна роботи полягає в тому, що в ній запропоновано метод розпізнавання мови за голосом користувача, який ґрунтується на використанні згорткової нейронної мережі для аналізу мовлення користувача з попереднім перетворенням аудіосигналу до спектрограм, що дозволяє виконувати визначення мови за голосом користувача.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Бондаренко, М. Ф. Распознавание речи [Текст] : этапы развития, современные технологии и перспективы их применения / М. Ф. Бондаренко, А. В. Работягов, С. В. Щепковский // Бионика интеллекта. – 2010. – № 2 (73). – С. 164-168.
2. Гребнов С. В. Аналитический обзор методов распознавания речи в системах голосового управления [Текст] / С. В. Гребнов // Вестник ИГЭУ. – 2009. – Вып. 3. – С. 113-115.
3. Прохоров, А. Системы автоматического распознавания речи [Электрон. ресурс] / А. Прохоров. – Компьютер Пресс. – 2003. – № 7. – Режим доступа : <https://compress.ru/article.aspx?id=11331>.
4. Воробьева, С. А. Методы распознавания речи [Текст] / С. А. Воробьева // Молодой ученый. – 2016. – №26. – С. 136-141.
5. Фролов, А. В. Синтез и распознавание речи. Современные решения [Электрон. ресурс] / А. В. Фролов, Г. В. Фролов. – Режим доступа : <http://www.frolov-lib.ru/books/hi/ch05.html>.
6. Грачев А. М. Статистические подходы к автоматическому распознаванию речи [Текст] / А. М. Грачев // Вестник Нижегородского университета им. Н. И. Лобачевского. – 2015. – № 2 (2). – С. 376–379.
7. Круглов В. В. Искусственные нейронные сети. Теория и практика [Текст] / В. В. Круглов, В. В. Борисов. – 2-е изд., стереотип. – М. : Горячая линия-Телеком, 2002. – 382 с.
8. Kone C. Introducing Convolutional Neural Networks in Deep Learning [Electronic resource] / Cyrille Kone. – Access mode : <https://towardsdatascience.com/introducing-convolutional-neural-networks-in-deep-learning-400f9c3ad5e9>.
9. Saha, S. A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way [Electronic resource] / S. Saha. – Access mode : <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural->

networks-the-eli5-way-3bd2b1164a53.

10. Convolutional Neural Network Tutorial [Electronic resource] : From Basic to Advanced. – Access mode : <https://missinglink.ai/guides/convolutional-neural-networks/convolutional-neural-network-tutorial-basic-advanced/>.

11. Николенко, С. Глубокое обучение [Текст] / С. Николенко, А. Кадури, Е. Архангельская. – СПб. : Питер, 2018. – 480 с.

12. PIA22925 [Electronic resource] : Vibrations from InSight's First 20 Minutes on Mars. – Access mode : <https://photojournal.jpl.nasa.gov/catalog/PIA22925>.

13. Advantages and Disadvantages of Python – How it is dominating programming world [Electronic resource]. – Access mode : <https://data-flair.training/blogs/advantages-and-disadvantages-of-python/>.

14. Златопольский Д. М. Основы программирования на языке Python [Текст]/ Д.М. Златопольский. – М.: ДМК Пресс, 2017. – 284 с.

15. Гэддис Т. Начинаем программировать на Python[Текст]/ Т. Гэддис. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.

16. TensorFlow vs Keras [Electronic resource] : Introduction to Machine Learning. – Access mode : <https://www.bmc.com/blogs/tensorflow-vs-keras/>.

17. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow [Текст]/ О. Жерон. – К. : Диалектика, 2020. – 688 с.

18. Джулли А. Библиотека Keras – инструмент глубокого обучения [Текст] / А. Джулли, С. Пал. – М. : ДМК Пресс, 2018. – 294 с.

19. Методичні вказівки по економічному обґрунтуванню дипломних проектів для студентів спеціальностей 8.05010301 «Програмне забезпечення автоматизованих систем» та 8.05010201 «Інформаційні технології проектування» для усіх форм навчання [Текст] / В. В. Остапенко, Є. М. Сердюк. – Запоріжжя : ЗНТУ, 2015. – 22 с.

20. Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом [Текст] : ДСТУ Б В.2.5-82:2016. – На заміну ДБН В.2.5-27-2006 ; чинний від 2017-04-01. – К. : ДП УкрНДНЦ,

2016. – 109 с. – (Державний Стандарт України).

21. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроям [Електрон. ресурс] : НПАОП 0.00-7.15-18. – На заміну НПАОП 0.00-1.28-10 ; чинний від 2018-05-18. – К. : Мінсоцполітики України, 2018. – 6 с. – Режим доступу: <http://zakon.rada.gov.ua/laws/show/z0508-18>.

22. Охорона праці [Текст] : навч. посіб. / З. М. Яремко, С. В. Тимошук, О. І. Третяк, Р. М. Ковтун ; за ред. проф. З. М. Яремка. – Львів : Видавничий центр ЛНУ імені Івана Франка, 2010. – 374 с.

23. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень [Текст]. – Харків : Форт, 2000. – 24 с.

24. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин [Електронний ресурс] : ДСанПіН 3.3.2.007-98. – Чинний від 1998-12-10. – К. : МОЗ України, 1998. – Режим доступу : <http://mozdocs.kiev.ua/view.php?id=2445>.

25. ДБН В.2.5-67:2013 Опалення, вентиляція, кондиціонування [Текст]. – К. : Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2017. – 149 с.

26. ДБН В.2.5-28:2018. Природне та штучне освітлення. Посібник [Текст] : проблеми природного і штучного освітлення. – К. : Етін, 2019. – 180 с.

27. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку. Законодавство про охорону праці. Том 1 [Текст] / Харків : Форт, 2013. – 368 с.

28. Жидецький, В. Ц. Основи охорони праці [Текст] : підручник / В. Ц. Жидецький. – 5-те вид., доп. – К. : Знання, 2014. – 373 с.

29. Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою [Текст] : ДСТУ Б В.1.1-36:2016. – На заміну НАПБ Б.03.002-2007 ; чинний від 2017-01-01. –

К. : Мінрегіонбуд України, 2016. – 66 с.

30. ДБН В.1.1-7:2016. Пожежна безпека об'єктів будівництва. Загальні вимоги [Текст]. – К. : Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2016. – 39 с.

31. Шоботов В. М. Цивільна оборона [Текст] : Навчальний посібник / В. М. Шоботов. – Вид. 2-ге, перероб. – К. : Центр навчальної літератури, 2006. – 438 с.

ДОДАТОК А
Технічне завдання

Вступ

У дипломній кваліфікаційній роботі магістринеобхідно виконати дослідження проблеми розпізнавання мови за голосом користувача, для чого повинен бути виконаний аналіз проблеми на даний момент, методів, які використовуються для розпізнавання мовлення, а на основі виконаного аналізу повинен бути запропонований власний метод і виконана його програмна реалізація.

A.1 Підстави для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу магістра темою “Дослідження та програмна реалізація методів розпізнавання мови за голосом користувача”, затверджене наказом № 301 від 3 листопада 2020 р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

Програмне забезпечення має використовуватись для розпізнавання мови за голосом користувача.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Програма повинна забезпечувати виконання наступних функцій:

- створення спектрограм;
- побудова нейронної мережі;
- навчання нейронної мережі;
- збереження результатів навчання;

- завантаження нейронної мережі;
- запис мовлення користувача;
- розпізнавання мови.

Вхідними даними є голос користувача і відповідне повідомлення у вигляді аудіофайлу.

Вихідними даними є навчена модель розпізнавання та мова, яку було розпізнано з повідомлення.

А.3.2 Умови експлуатації

Для роботи програмного забезпечення необхідно на серверній частині інсталиювати:

- інтерпретатор мови програмування Python версії 3;
- пакети Keras, Tensorflow та NumPy.

А.3.3 Вимоги до складу та параметрів технічних засобів

Для експлуатації програмного забезпечення необхідні наступні технічні засоби:

- процесор з 4 ядрами, при цьому частота кожного має бути не менше 3 ГГц;
- оперативна пам'ять – не менше 2 Гб;
- жорсткий диск – вільний простір не менше 1 Гб.

А.3.4 Вимоги до транспортування та збереження

Вимоги до транспортування і зберігання аналогічні вимогам, пропонованим до носія, на якому зберігається програма. Додаткових вимог не висувається.

A.3.5 Вимоги до програмної документації

Програма повинна поставлятися з наступними документами:

- текстом програми;
- описом експлуатації та тестування програми.

A.3.6 Вимоги до маркування та упакування

Програма може бути записана на оптичному, гнучкому або жорсткому диску. У випадку розповсюдження програмного забезпечення на відповідному носії має бути вказано маркування «Дослідження та програмна реалізація методів розпізнавання мови за голосом користувача».

A.4 Стадії та етапи розробки

Виконання дипломної кваліфікаційної роботи магістра повинно бути реалізовано на основі наступних етапів:

- розроблення методу розпізнавання автоматично згенерованих кодів;
- аналіз вимог до програмного забезпечення;
- проєктування програмного забезпечення;
- реалізація програмного забезпечення;
- тестування і відлагодження програмного забезпечення;
- оформлення документації.

Програмний продукт повинен бути виконаний в терміни, визначені календарним планом, наведеним у завданні на роботу.

A.5 Порядок контролю та приймання

Дипломна кваліфікаційна робота магістра повинна бути виконана у відповідності з календарним планом, зазначеним у завданні на роботу.

Дипломна кваліфікаційна робота магістра має бути прийнята консультантами відповідних розділів, після чого затверджена завідуючим кафедрою, а на 13-й тиждень захищена на засіданні екзаменаційної комісії за відповідним графіком.

ДОДАТОК Б
Текст програми

```
#підключення модуля передобробки даних з бібліотеки машинного
навчання Scikit-learn
from sklearn import preprocessing

#підключення класу для побудови моделей з нейромережевої
бібліотеки keras
from keras.models import Model

#підключення класу для групування даних з нейромережевої
бібліотеки keras
from keras.models import Sequential

#підключення класу для побудови згорткового шару з нейромережевої
бібліотеки keras
from keras.layers import Conv2D

#підключення класу для забезпечення пулінгу з нейромережевої
бібліотеки keras
from keras.layers import MaxPooling2D

#підключення класу для забезпечення середнього пулінгу з
нейромережевої бібліотеки keras
from keras.layers import AveragePooling2D

#підключення класу для побудови повнозв'язного шару з
нейромережевої бібліотеки keras
from keras.layers import Dense

#підключення класу для виконання згладжування даних з
нейромережевої бібліотеки keras
from keras.layers import Flatten

#підключення дропауту для уникнення проблеми перенавчання з
нейромережевої бібліотеки keras
from keras.layers import Dropout

#підключення засобів введення з нейромережевої бібліотеки keras
```

```

from keras.layers import Input

#підключення засобів організації функцій активації з нейромережевої
бібліотеки keras
from keras.layers import Activation

#підключення засобів створення оптимізатора Адама з
нейромережевої бібліотеки keras
from keras.optimizers import Adam

#підключення засобів завершення навчання з нейромережевої
бібліотеки keras
from keras.callbacks import EarlyStopping

#підключення засобів збереження отриманих результатів навчання з
нейромережевої бібліотеки keras
from keras.callbacks import ModelCheckpoint

#підключення засобів завантаження моделі з H5-файлів
нейромережевої бібліотеки keras
from keras.models import load_model

#підключення засобів штрафування з нейромережевої бібліотеки keras
from keras import regularizers

#підключення засобів для роботи з масивами та переліками
import np_utils

import tensorflow
from keras import backend as K

#визначення параметрів сесії
infos = tensorflow.ConfigProto(
    intra_op_parallelism_threads=1,

```

```
inter_op_parallelism_threads=1)
```

```
tensorflow.set_random_seed(2100)
```

```
#створення сесії
```

```
ts = tensorflow.Session
```

```
(graph=tensorflow.get_default_graph(),
```

```
config=infos)
```

```
#робота реалізується через Keras з використанням Tensorflow в якості  
бекенду
```

```
K.set_session(ts)
```

```
#функція створення моделі
```

```
def create(params):
```

```
#створення моделі на основі стеку шарів
```

```
nn = Sequential()
```

```
#створення першого згорткового шару
```

```
nn.add(Conv2D(
```

```
#кількість нейронів
```

```
8,
```

```
#розмір фільтру
```

```
(3, 3),
```

```
#крок переміщення фільтру
```

```
strides=(1, 1),
```

```
#створення паддингу
```

```
padding='same',
```

```

#визначення параметрів штрафування
    l2reg=regularizers.l2(0.001),
#визначення вхідних параметрів
    params=params))

#встановлення функції активації
    nn.add(Activation('relu'))

#визначення пулінгу
    nn.add(
#визначення пулінгу
    MaxPooling2D(
#визначення розміру області пулінгу
    pool_size=(3, 3),
#визначення кроку
    strides =(2, 2),
#визначення паддингу
    padding ='same'))

#створення другого згорткового шару
    nn.add(Conv2D(
#кількість нейронів
    16,
#розмір фільтру
    (3, 3),
#крок переміщення фільтру
    strides =(1, 1),
#створення паддингу
    padding ='same',
#визначення параметрів штрафування

```

```

l2reg=regularizers.l2(0.001)))

#встановлення функції активації
nn.add(Activation('relu'))

#визначення пулінгу
nn.add(MaxPooling2D(pool_size=(3, 3), strides =(2, 2), padding
='same'))

#створення третього згорткового шару
nn.add(Conv2D(
#кількість нейронів
64,
#розмір фільтру
(5, 5),
#крок переміщення фільтру
strides =(1, 1),
#створення паддингу
padding ='same',
#визначення параметрів штрафування
l2reg=regularizers.l2(0.001)))

#встановлення функції активації
nn.add(Activation('relu'))

#визначення пулінгу
nn.add(MaxPooling2D(pool_size=(3, 5), strides =(1, 5), padding
='same'))

```

```

#визначення пулінгу
    nn.add(AveragePooling2D(
#розмір області
        pool_size=(5, 5),
#крок
        strides =(5, 5),
#визначення паддингу
        padding ='valid'))

#визначення згладжування даних
    nn.add(Flatten())

#створення повнозв'язного шару
    nn.add(Dense(
#кількість нейронів
    16,
#встановлення функції активації
        activation='relu',
#визначення параметрів штрафування
        l2reg=regularizers.l2(0.001)))

#визначення дропауту для уникнення перенавчання
    nn.add(Dropout(0.5))

#визначення даних повнозв'язного шару
    nn.add(Dense(len(RecogL)))

#визначення функції активації
    nn.add(Activation('softmax'))

```

```
#визначення функції оптимізації
```

```
dt = Adam(lr=0.01)
```

```
#компіляція моделі
```

```
nn.compile(  
    operl='categorical_crossentropy',  
    opero=dt,
```

```
#визначення метрики результатів навчання
```

```
metrics=['accuracy'])
```

```
return nn
```

ДОДАТОК В
Слайди презентації

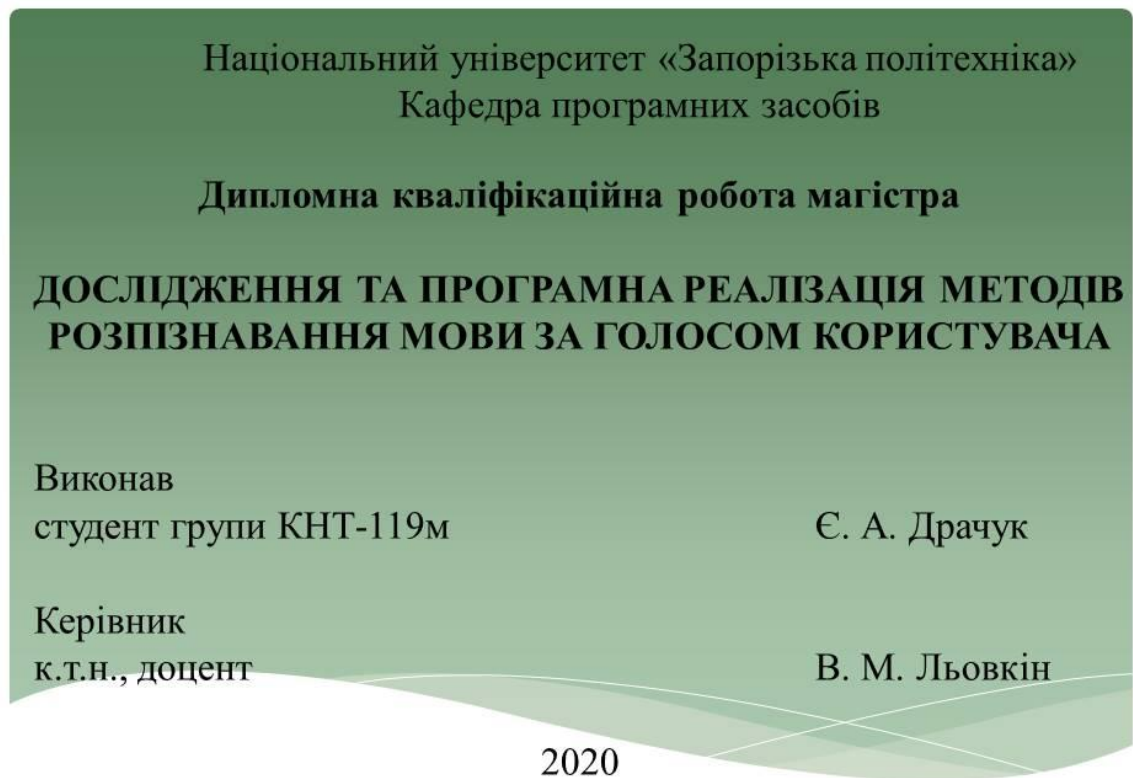


Рисунок В.1 – Слайд 1

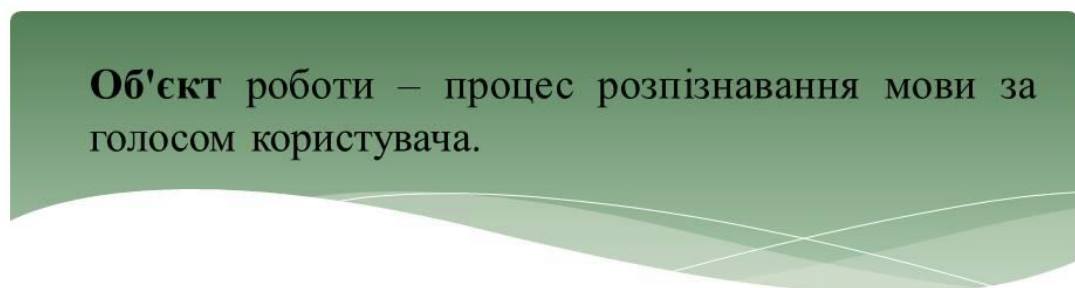


Рисунок В.2 – Слайд 2

Завдання

- ✓ дослідження методів розпізнавання мови за голосом користувача;
- ✓ розробка технічного завдання;
- ✓ проектування програмного забезпечення;
- ✓ розробка програмного забезпечення;
- ✓ тестування та налагодження програми;
- ✓ економічне обґрунтування доцільності розробки.

3

Рисунок В.3 – Слайд 3

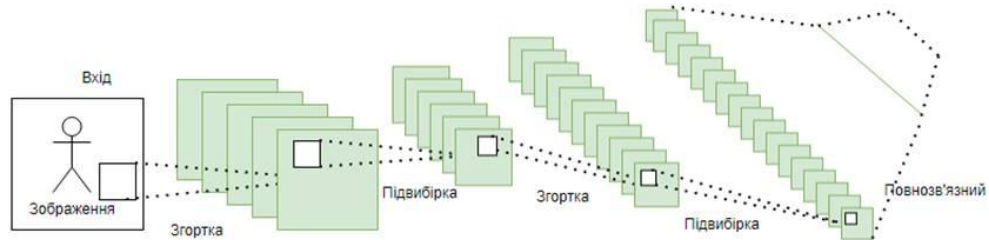
Порівняння архітектур нейронних мереж

Критерій порівняння	Багатошаровий перцептрон	Згорткові нейронні мережі	Рекурентні нейронні мережі
Тип даних	Універсальна	Зображення	Послідовності
Задача класифікації	Придатна	Придатна	Придатна

4

Рисунок В.4 – Слайд 4

Згорткова нейронна мережа



5

Рисунок В.5 – Слайд 5

Задача розпізнавання мови за голосом користувача

Нехай є множина аудіофрагментів, що представляють мовлення користувача, $M = \{M_i\}$, $i = 1..n$, де M_i – фрагмент мовлення користувача (аудіоповідомлення) під номером i , і множина $Y = \{Y_i\}$, $i = 1..n$, де Y_i – мова, якою відбувається мовлення.

Задача розпізнавання мови за голосом користувача полягає в тому, щоб для кожного повідомлення M_i , визначеного аудіофрагментом, визначити клас, до якого воно належить, представлений мовою повідомлення, на основі використання функції $F(b)$, яка виражена на основі формули:

$$F(M_i) = \{Y_i\}.$$

Для розв'язання задачі розпізнавання мови за голосом користувача потрібно визначити вигляд функції $F(M_i)$.

6

Рисунок В.6 – Слайд 6

Метод розпізнавання мови за голосом користувача

Етап 1. Створення моделі на основі згорткової нейронної мережі, що складається з 4 шарів:

- 3 згорткові шари;
- 1 повнозв'язний шар.

Етап 2. Формування вибірок даних зі створенням спектрограм для фрагментів аудіофайлів з тривалістю 10 секунд.

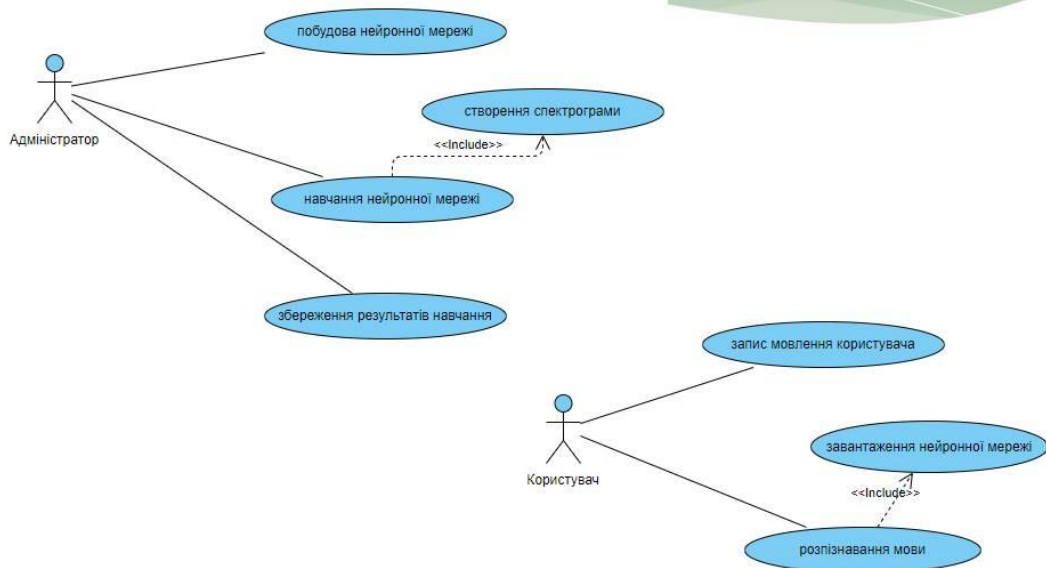
Етап 3. Навчання моделі.

Етап 4. Розпізнавання за заданим аудіоповідомленням шляхом голосування на фрагментах.

7

Рисунок В.7 – Слайд 7

Функціональні вимоги до програми



8

Рисунок В.8 – Слайд 8

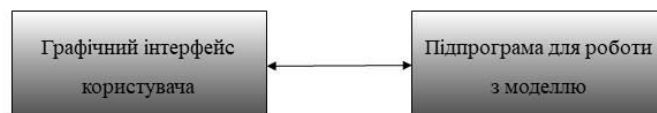
Порівняння мов програмування

Критерій порівняння	Python	R
Робота з даними	Зручна	Зручна
Машинне навчання	В повній мірі	Наявне
Готові бібліотеки	Велика кількість для різних задач	Наявні

9

Рисунок В.9 – Слайд 9

Структура програми



10

Рисунок В.10 – Слайд 10



Рисунок В.11 – Слайд 11

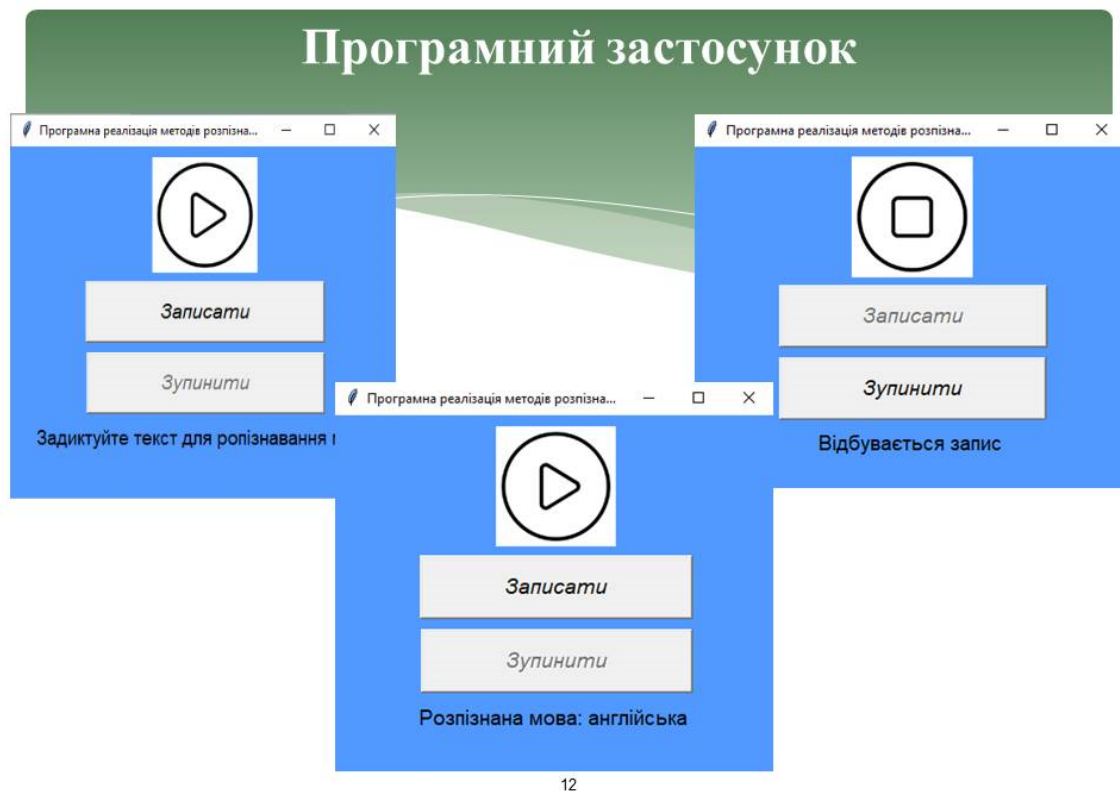


Рисунок В.12 – Слайд 12

Висновки

Програмне забезпечення складається з програмного застосунку з графічним інтерфейсом, який дозволяє записувати мовлення користувача та працювати з готовою моделлю, та окремого консольного застосунку, який дозволяє керувати нейронною мережею, зокрема створювати модель, виконувати її навчання та зберігати результати.

Наукова новизна роботи полягає в тому, що в ній запропоновано метод розпізнавання мови за голосом користувача, який ґрунтується на використанні згорткової нейронної мережі для аналізу мовлення користувача з попереднім перетворенням аудіо-сигналу до спектрограм, що дозволяє виконувати визначення мови за голосом користувача.