МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій і систем

Кафедра програмного забезпечення автоматизованих систем

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

до кваліфікаційної роботи

«магістр»

на тему: **«Програмне забезпечення комунікатора тварин»**

|  |  |
| --- | --- |
| Виконав: | студент 2 курсу, групи МПЗ-2204 |
| Напряму підготовки: | |
| 121 «Інженерія програмного забезпечення» | |
| (шифр і назва напряму підготовки) | |
|  |  |
| Студент: | Гогулов Я.В. |
|  | (прізвище та ініціали) |
| Керівник: | Голуб С.В. |
|  | (прізвище та ініціали) |
| Рецензент: |  |
|  | (прізвище та ініціали) |

Черкаси 2023**АНОТАЦІЯ**

Гогулов Я.В.

Програмне забезпечення комунікатора тварин

121 «Інженерія програмного забезпечення»

Черкаський державний технологічний університет

Черкаси 2023

Тема кваліфікаційної роботи є вельми актуальною з кількох причин. По-перше, враховуючи зростаючі вимоги людей до розуміння своїх домашніх тварин та отримання персоналізованих рішень, компанії активно інвестують у розробку інновацій, спрямованих на поліпшення комунікації з тваринами.

Мета дослідження

Метою даної роботи є розробка програмного забезпечення для комунікатора тварин, зокрема котів, з використанням нейронної мережі. Досліджуються різні типи нейронних мереж та їх архітектури з метою створення системи, яка здатна розпізнавати емоції кота та зрозуміти приблизно, що він хоче, або що хоче зробити, і перекласти на «людську мову».

**Методологія дослідження.** Для досягнення поставленої мети використовується аналіз різноманітних архітектур нейронних мереж. Розглядаються їхні переваги та недоліки в контексті розпізнавання звуків кота. Особлива увага приділяється вибору оптимальної архітектури для інтеграції в систему.

**Результати дослідження.**

Аналіз показує, що велика кількість архітектур нейронних мереж пропонується для розпізнавання звуків. Однак деякі з них не є ідеальними з точки зору всіх параметрів продуктивності. Робота визначає найбільш популярні архітектури та розглядає їхні переваги та обмеження.

**Висновки**

Розробка програмного забезпечення для комунікатора тварин, зокрема котів, за допомогою нейронної мережі, є перспективною галуззю дослідження. Інтеграція такого програмного забезпечення в системи в різні системи допоможе людям зрозуміти свого домашнього улюбленця, та може значно покращити їхню продуктивність та здатність до розпізнавання звуків. Дана робота слугує внеском у розвиток технологій штучного інтелекту в галузі тваринного спілкування.

Зокрема, в сучасному світі люди пошуково шукають доступні застосунки, які, хоча не завжди відповідають всім їхнім очікуванням, вже є відкритими для користування. Існує активний інтерес інвесторів у розробці застосунків, які допомагають взаєморозумінню між людьми та їхніми домашніми улюбленцями. Такі застосунки мають спрощувати взаємодію із тваринами та сприяти кращому розумінню їхніх потреб.

Відзначається, що хоча існують певні доступні застосунки, вони не завжди відповідають усім очікуванням користувачів. Інвестиції у подальший розвиток та удосконалення цих застосунків можуть сприяти створенню ефективних інструментів для взаємодії та розуміння поведінки домашніх тварин.

Отже, актуальність даної теми у вигляді дослідження визначається реальним прагненням людей отримати засоби комунікації та розуміння своїх домашніх улюбленців через застосунки, які відповідають їхнім очікуванням та потребам.

Інструменти, які були використані для реалізації цього проєкту:

1. Мови програмування: Для розробки програмного забезпечення було використано мови програмування, такі як:
2. Python: Використовується для структуризації.

Машинне навчання та аналіз даних: Для розробки моделі нейромережі для визначення емоції кота через його звук, прогнозів та надання рекомендацій з ефективного використання газу було використано таку бібліотеку як TensorFlow.js - Бібліотека використовується для реалізації моделей штучного інтелекту для розрахунків та прогнозів.

Результатом кваліфікаційної роботи магістра є розроблене програмне забезпечення для користувачів, які хочуть дізнатись, що конкретно хоче кіт. Система є сучасною та зручною, і вона задовольняє всі вимоги з точки зору функціональності, користувальницької зручності та логіки.

**Ключові слова:** розробка програмного забезпечення, нейромережа, класифікація, ідентифікація, розпізнавання звуку, автоматизація, комунікація, звуки кота, дані, ефективність.

**ANNOTATION**

|Hohulov Y.V.

Animal communicator software

121 "Software engineering"

Cherkasy State Technological University

Cherkasy 2023

The topic of the qualification work is very relevant for several reasons. Firstly, given the growing demands of people to understand their pets and receive personalized solutions, companies are actively investing in the development of innovations aimed at improving communication with animals.

Research objective

The aim of this work is to develop software for an animal communicator, in particular cats, using a neural network. Different types of neural networks and their architectures are investigated in order to create a system that is able to recognize the emotions of a cat and understand approximately what it wants or what it wants to do and translate it into "human language".

Research methodology. To achieve this goal, we analyze various neural network architectures. Their advantages and disadvantages in the context of cat sound recognition are considered. Particular attention is paid to choosing the optimal architecture for integration into the system.

Research results.

The analysis shows that a large number of neural network architectures are proposed for sound recognition. However, some of them are not ideal in terms of all performance parameters. The paper identifies the most popular architectures and discusses their advantages and limitations.

Conclusions.

The development of software for communicating with animals, in particular cats, using a neural network is a promising area of research. Integrating such software into various systems will help people understand their pets and can significantly improve their performance and sound recognition abilities. This work contributes to the development of artificial intelligence technologies in the field of animal communication.

In particular, in the modern world, people are searching for affordable applications that, although they do not always meet all their expectations, are already open for use. There is an active interest of investors in the development of applications that help people and their pets to understand each other. Such applications should simplify interaction with animals and contribute to a better understanding of their needs.

It is noted that although there are some available applications, they do not always meet all user expectations. Investing in the further development and improvement of these applications can help create effective tools for interacting with and understanding pet behavior.

Thus, the relevance of this topic in the form of a study is determined by the real desire of people to get the means to communicate and understand their pets through applications that meet their expectations and needs.

The tools used to implement this project are:

1. Programming languages: Programming languages such as:

1) Python: Used for structuring.

Machine learning and data analysis: A library such as TensorFlow.js was used to develop a neural network model to determine the emotion of a cat through its sound, make predictions and provide recommendations for efficient gas use - The library is used to implement artificial intelligence models for calculations and predictions.

The result of the master's qualification work is the developed software for users who want to know what exactly the cat wants. The system is modern and user-friendly, and it meets all the requirements in terms of functionality, user-friendliness, and logic.

Keywords: software development, neural network, classification, identification, sound recognition, automation, communication, cat sounds, data, efficiency.

**ЗМІСТ**

[ВСТУП 6](#_Toc156386610)

[РОЗДІЛ 1. ІСНУЮЧІ МЕТОДИ ТА ЗАСОБИ РОЗВ’ЯЗАННЯ ПОСТАВЛЕНИХ ЗАВДАНЬ 12](#_Toc156386611)

[1.1. Визначення актуальних проблем 12](#_Toc156386612)

[1.2. Аналіз існуючих методів та засобів 13](#_Toc156386613)

[1.3. Аналіз існуючих аналогів програмного забезпечення 17](#_Toc156386614)

[Висновок до першого розділу 25](#_Toc156386615)

[РОЗДІЛ 2. ТЕОРЕТИЧНІ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ 27](#_Toc156386616)

[2.1. Теоретичні дослідження 27](#_Toc156386617)

[2.2 Експериментальні дослідження 34](#_Toc156386618)

[Висновок до розділу: 41](#_Toc156386619)

[РОЗДІЛ 3. ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ У ПРАКТИКУ ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ 42](#_Toc156386620)

[3.1 Моделювання предметної області 42](#_Toc156386621)

[3.1.1. Предметна область моделювання. Модель предметної області. Словник предметної області. 44](#_Toc156386622)

[3.1.2. Елементи моделювання предметної області 44](#_Toc156386623)

[3.1.3. Робоча область моделювання 47](#_Toc156386624)

[3.2 Формування та аналіз вимог 49](#_Toc156386625)

[3.2.1. Формування вимог до програмного забезпечення. Первинні і детальні вимоги. Вимоги замовника і розробника. Функціональні та нефункціональні вимоги 49](#_Toc156386626)

[3.2.2. Формування вимог за допомогою діаграми прецедентів 52](#_Toc156386627)

[3.2.3. Проектування логічної структури програмного комплексу 54](#_Toc156386628)

[3.2.3.1. Діаграми класів 54](#_Toc156386629)

[3.2.3.2. Діаграми пакетів 58](#_Toc156386630)

[3.2.4 Архітектурне проектування 60](#_Toc156386631)

[3.2.4.1 Діаграма компонентів 61](#_Toc156386632)

[3.2.4.2 Розгортання програмної системи на апаратних засобах. Діаграма розгортання 63](#_Toc156386633)

[3.2.5 Моделювання поведінки системи. 64](#_Toc156386634)

[3.2.5.1 Діаграма діяльності 64](#_Toc156386635)

[3.2.5.2 Діаграма послідовності 66](#_Toc156386636)

[3.2.5.3 Діаграма комунікації 69](#_Toc156386637)

[3.2.5.4 Діаграма скінченного автомату. 70](#_Toc156386638)

[**Висновок до третього розділу** 72](#_Toc156386639)

[РОЗДІЛ 4. РОЗРОБКА ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 73](#_Toc156386640)

[4.1. Розробка програмного комплексу 73](#_Toc156386641)

[4.1.1. Обґрунтування вибору засобів реалізації 73](#_Toc156386642)

[4.1.2. Опис структурної (функціональної) схеми системи 74](#_Toc156386643)

[4.1.3. Опис логічної схеми системи 74](#_Toc156386644)

[4.1.4. Розробка бази даних 75](#_Toc156386645)

[4.1.5. Розробка інтерфейсу користувача 75](#_Toc156386646)

[4.1.6. Опис розробки програмних компонентів 76](#_Toc156386647)

[4.2 Тестування системи 77](#_Toc156386648)

[4.2.1. Модульне тестування 77](#_Toc156386649)

[4.2.2. Інтеграційне тестування 79](#_Toc156386650)

[4.2.3. Системне тестування 80](#_Toc156386651)

[4.2.4. Приймальне тестування 82](#_Toc156386652)

[4.3 Приклади впровадженого програмного комплексу 83](#_Toc156386653)

[ВИСНОВКИ 86](#_Toc156386654)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 87](#_Toc156386655)

[ДОДАТОК Б 113](#_Toc156386656)

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

|  |  |  |
| --- | --- | --- |
| Програмне забезпечення | − | сукупність програм, інструментів, документації та даних, які забезпечують функціонування комп'ютерної системи. Це включає в себе різноманітні типи програм, починаючи від операційних систем і закінчуючи застосунками для різних завдань. |
| ШІ  НМ  DAL  ML | −  −  −  − | штучний інтелект;  нейронна мережа;  dataaccesslayer;  machinelearning; |
| Tensorflow | − | Відкрите програмне забезпечення(бібліотека) для розробки та навчання моделей глибокого навчання (deeplearning). Це один із найпопулярніших фреймворків для створення та тренування штучних нейронних мереж. |
|  | − | Організована колекція даних, яка зберігається та доступна для обробки за допомогою комп'ютерних систем. |
| IT  (інформаційні технології) | − | Галузь знань, яка включає в себе розробку, використання та управління комп'ютерною технологією для обробки та передачі інформації. |
|  | − |  |

# ВСТУП

**Актуальність теми.** В сучасному світі швидкого технологічного розвитку виникає необхідність впровадження інноваційних підходів для поліпшення взаємодії людини з тваринами в різноманітних сценаріях, починаючи від домашніх улюбленців до тварин у сільському господарстві та лісовому господарстві. Таким чином, актуальність створення програмного забезпечення для комунікації з тваринами виходить за межі побутових та розважальних застосувань, охоплюючи широкий спектр важливих областей.

Ця кваліфікаційна робота присвячена розробці програмного забезпечення, яке слугує в якості комунікатора тварин, надаючи можливість взаємодії та розуміння потреб тварин у відповідності з їхньою природною мовою та поведінковими виявами. Застосування програмного комунікатора тварин відкриває нові перспективи для більш ефективного взаєморозуміння та сприяє покращенню якості життя тварин, підвищенню їхнього комфорту та забезпеченню необхідних потреб.

У цьому контексті, дослідження взаємодії людини та тварин через програмне забезпечення комунікатора стає важливою складовою розвитку інтерфейсів для комунікації, що враховує особливості сприйняття та вираження тваринами інформації. Кваліфікаційна робота спрямована на розробку та дослідження функціоналу програмного забезпечення, яке не лише дозволяє виявити потреби та стан тварин, але й сприяє взаєморозумінню та покращує комунікацію між людиною та твариною.

Взаємодія із тваринами завжди викликала цікавість та бажання кращого розуміння їхньої мови і емоцій. Коти, зокрема, часто виявляють свої почуття через вираз обличчя, тіло, та звукові сигнали. Однак розкодування цих сигналів та розуміння емоцій котів залишається великим викликом.

Ця магістерська робота ставить перед собою завдання розробки та застосування програмного забезпечення для комунікації з котами на основі їхніх емоційних виявів. Для досягнення цієї мети, ми використовуємо мову

програмування Python та ряд бібліотек, зокрема TensorFlow та scikit-learn.

**Мета та завдання дослідження.** Метою нашого дослідження є підвищення адекватності програмної реалізації процесів класифікації звуків тварин за рахунок використання агентного підходу та агентного програмування. Є яке здатне розпізнавати та класифікувати емоційні стани, звуків котів та надавати інтерпретацію їхнього поведінкового виразу. Для досягнення цього ми використовуємо методи машинного навчання, зокрема глибокі нейронні мережі, інтегровані у фреймворк TensorFlow.

Завдання дослідження включають:

Збір та аналіз аудіо- та відеозаписів для виокремлення емоційних виявів котів.

Розробка та навчання моделей глибокого навчання для розпізнавання емоційних станів на основі вхідних даних.

Інтеграція програмного забезпечення у практичні сценарії взаємодії з котами та оцінка ефективності розробленої системи.

Використані технології та інструменти

Для реалізації нашого дослідження ми обираємо мову програмування Python як основний інструмент розробки. Використання бібліотеки TensorFlow дозволяє нам будувати та навчати складні моделі глибокого навчання для аналізу емоційних виявів у великому обсязі даних. Додатково, ми використовуємо бібліотеку scikit-learn для проведення аналізу та оцінки результатів.

Ця магістерська робота покликана розширити наше розуміння мови та емоцій котів, використовуючи передові методи обробки сигналів та машинного навчання. Розроблене програмне забезпечення може знайти застосування у вивченні поведінки тварин, ветеринарній медицині, та у сфері взаємодії людей та котів у різних сценаріях.

**Об'єкт дослідження:** є процес класифікації звукових повідомлень. Комунікації із тваринами.

**Предметом дослідження:** є процес створення програмного забезпечення інформаційної технології звуків тварин, зокрема котів.

**Методи дослідження:** розпізнавання та класифікація моделей для аналізу емоцій та звуків котів.

**Наукова новизна одержаних результатів**. Удосконалено метод агентного проектування програмного забезпечення і класифікатора звуків тварин, зокрема котів, за рахунок розробки нових процесів поєднання програмних модулів шляхом платформі Windows, що підвищує адекватність реалізації розроблених алгоритмів.

**Практичне значення одержаних результатів.** Програмно реалізовано метод аналізу і класифікації звукових даних котів на платформі Windows.

Покращена комунікація: Програмне забезпечення для спілкування з тваринами, яке використовує TensorFlow, потенційно може подолати розрив у спілкуванні між людьми та котами, аналізуючи та інтерпретуючи їх вокалізацію, мову тіла та вираз обличчя. Точно визначаючи та розуміючи котячі емоції, таке програмне забезпечення може допомогти людям належним чином реагувати на їхні потреби, що призведе до кращого спілкування та міцніших зв'язків між людьми та котами.

Покращене самопочуття: Розуміння емоційного стану кота має вирішальне значення для забезпечення його благополуччя. Програмне забезпечення для спілкування з тваринами може надати цінну інформацію про емоційний стан кота, допомагаючи власникам виявити ознаки стресу, тривоги або дискомфорту. Ці знання дозволять власникам вжити відповідних заходів для полегшення цих негативних емоцій і створити більш позитивне та комфортне середовище для своїх котячих компаньйонів.

Поведінковий аналіз: Програмне забезпечення для комунікації з тваринами, що використовує TensorFlow, може зробити свій внесок у сферу аналізу поведінки тварин. Точно визначаючи та інтерпретуючи котячі емоції, дослідники можуть глибше зрозуміти, як емоції впливають на поведінку котів. Ці знання можуть бути застосовані в різних сферах, таких як добробут тварин, ветеринарна медицина та психологія тварин, що призведе до покращення стратегій догляду та лікування котів.

Дослідження та освіта: Програмне забезпечення для спілкування з тваринами може слугувати цінним інструментом для дослідників та викладачів, які вивчають поведінку та емоції котів. Надаючи засоби для аналізу та інтерпретації котячих емоцій, це програмне забезпечення може сприяти дослідженням на такі теми, як вплив факторів навколишнього середовища на котячі емоції, вплив взаємодії людини і кота на емоційне благополуччя та розвиток емоційного інтелекту у котів. Такі дослідження можуть сприяти поглибленню знань у сфері поведінки та добробуту тварин.

Практичне застосування: Практичне значення програмного забезпечення для спілкування з тваринами виходить за рамки власників домашніх тварин. Наприклад, притулки для тварин і рятувальні організації можуть скористатися цією технологією, отримуючи інформацію про емоційний стан котів, які перебувають під їхньою опікою, допомагаючи їм забезпечити належний догляд і знайти відповідні домівки для цих тварин. Крім того, ветеринари можуть використовувати це програмне забезпечення, щоб краще розуміти і задовольняти емоційні потреби своїх котячих пацієнтів, що призведе до покращення результатів лікування.

Таким чином, практичне значення програми-комунікатора на Python (TensorFlow) для виявлення емоцій у котів полягає в тому, що вона може покращити комунікацію, поліпшити самопочуття, уможливити аналіз поведінки, сприяти дослідженням та освіті, а також знайти практичне застосування в різних сферах, пов'язаних з доглядом за котами та їхнім благополуччям. Хоча конкретних наукових досліджень або публікацій на цю тему може бути небагато, ширше застосування програмного забезпечення для комунікації з тваринами і пов'язаних з ним технологій підкреслює практичну значущість цієї концепції.

Особистий внесок автора. Всі відображені в роботі результати представлені автором особисто.

Особистий внесок автора в цю роботу полягає в розробці та вдосконаленні алгоритмів нейронної мережі для розпізнавання звуків на відео, або за допомогою мікрофона. Автор провів аналіз різних архітектур нейронних мереж, визначив їхні переваги та недоліки з точки зору поставленого завдання – розпізнавання об'єктів та істот. Додатково, автор вніс вагомий внесок у вибір та оптимізацію підходів до інтеграції розробленої нейронної мережі в системи розумного будинку. Його дослідження висвітлює актуальність та важливість використання штучного інтелекту для покращення ефективності сучасних технологій у сфері систем розумного будинку.

Апробація результатів. Апробація результатів магістерської роботи включатиме в себе серію етапів для перевірки та підтвердження ефективності розробленого програмного забезпечення і нейронної мережі.

1. Тестування на синтетичних данних:

* проведення тестів на синтетичних даних для оцінки точності та чутливості нейронної мережі у різних умовах.

1. Валідація на реальних даних:

* застосування розробленого програмного забезпечення до реальних даних, що генеруються системами розумного будинку.

1. Експертна оцінка:

* залучення фахівців у галузі штучного інтелекту, систем розумного будинку або ветеринарії для оцінки результатів та визначення їхньої придатності та доцільності.

1. Тестування в реальних умовах:

* розгортання розробленого рішення в реальних умовах інтеграції з системами розумного будинку для перевірки його функціональності та ефективності в реальному використанні.

1. Залучення користувачів:

* організація пілотного тестування залученням користувачів (можливо, власників домашніх тварин), збір їхніх вражень і фідбеку щодо зручності та ефективності комунікації з тваринами через розроблену систему.

1. Аналіз результатів та виправлення недоліків:

* оцінка отриманих результатів, виявлення можливих недоліків та внесення виправлень для подальшого вдосконалення системи.

# РОЗДІЛ 1. ІСНУЮЧІ МЕТОДИ ТА ЗАСОБИ РОЗВ’ЯЗАННЯ ПОСТАВЛЕНИХ ЗАВДАНЬ

## 1.1. Визначення актуальних проблем

У процесі розробки системи класифікації звуків тварин, зокрема котів, на платформі Windows з метою ідентифікації звуку кота як , стикаємося з численними важливими аспектами, які потребують ретельного вивчення та визначення.

Багато власників тварин відчувають дефіцит засобів для ефективного спілкування зі своїми домашніми улюбленцями, зокрема з котами.

Відсутність чіткої класифікації звуків, які видають тварини, обмежує розуміння їхнього спілкування та потреб.

Відсутність ефективних методів перекладу звуків тварин, зокрема котів, на зрозумілу людську мову ускладнює взаєморозуміння між людьми і їхніми домашніми улюбленцями.

Не вистачає точних засобів для розуміння поведінки тварин, зокрема котів, що ускладнює взаємодію та навчання.

Формула Фур'є-перетворення:

Фур'є-перетворення звукового сигналу визначається наступним чином:

(1.1)

де:

* - Фур'є-перетворення,
* - звуковий сигнал,
* - частота.

Формула кількості гармонічних компонентів:

Кількість гармонічних компонентів у звуковому сигналі можна

визначити через аналіз його Фур'є-спектра.

(1.2)

де:

​ - кількість гармонік,

- ширина Фур'є-спектра сигналу,

- ширина одного піку на Фур'є-спектрі.

Загалом, чим більше гармонічних компонентів у Фур'є-спектрі, тим більше може бути розмаїття класифікації звуків у програмному забезпеченні "Комунікатор тварин".

## 1.2. Аналіз існуючих методів та засобів

У сучасному світі існує попит на розробку систем, які можуть класифікувати та ідентифікувати звуки тварин, зокрема котів, щоб зрозуміти, що він хоче. Один з ключових етапів цього процесу - аналіз існуючих методів та засобів програмного забезпечення для створення ефективної системи класифікації звуків.

Першим кроком є огляд різноманітних підходів та методів, які вже застосовуються у сфері акустичного аналізу та класифікації звуків тварин. Деякі з важливих напрямків включають:

Застосування методів машинного навчання:

Використання алгоритмів машинного навчання, зокрема нейронних мереж, для розпізнавання та класифікації звукових сигналів. Формули, які використовуються для розрахунку втрат чи точності моделей, можуть бути виражені за допомогою функцій втрат та метрик ефективності.

Обробка сигналів та акустичний аналіз:

Використання методів обробки сигналів, таких як швидке перетворення Фур'є (FFT) та акустичний аналіз, для виділення характерних особливостей звуків тварин. Формули для обчислення спектральних характеристик можуть

включати в себе математичні вирази для перетворень.

Важливим етапом є визначення конкретних вимог до системи класифікації. Це включає в себе розгляд технічних, функціональних та ергономічних аспектів, таких як швидкість обробки, точність класифікації, можливості розширення та зручність використання.

Закінчений аналіз існуючих методів та програмних засобів надає основу для подальшого вдосконалення системи класифікації звуків котів та вибору оптимального напрямку для розробки нового програмного забезпечення

Таблиця 1.1

**Переваги та недоліки машинного навчання**

|  |  |  |
| --- | --- | --- |
| **Метод** | **Вагомі переваги** | **Обмеження та недоліки** |
| Машинне навчання | - Висока точність класифікації | - Вимагає великого обсягу навчальних даних |
|  | - Здатність до автоматичного оновлення моделі | - Висока обчислювальна складність |
| Обробка cигналів | - Відмінна роздільна здатність в часі та частоті | - Залежність від якості вхідних сигналів |
|  | - Ефективність у виокремленні акустичних особливостей | - Складність у роботі з шумами |

Формули для оцінки точності:

1. Точність класифікації (Accuracy):

(1.3)

1. Чутливість (Sensitivity):

= (1.4)

1. Специфічність (Specificity):

= (1.5)

Тут:

* - true positives (правильно класифіковані позитивні),
* - true negatives (правильно класифіковані негативні),
* - false positives (неправильно класифіковані позитивні),
* - false negatives (неправильно класифіковані негативні).

Ці формули допоможуть числово оцінити ефективність методів та підходів до класифікації.

Такий підхід забезпечить конкретні та кількісні результати порівняння, що полегшить вибір оптимального методу для системи класифікації звуків котів.

При зміні швидкості відтворення звукових повідомлень, можливі різні ефекти, такі як зміна висоти та темпу звуку. Ці ефекти можуть впливати на сприйняття та класифікацію звуків, особливо у випадках, коли зберігається чи втрачається частина аудіосигналу.

Формула ефекту зміни швидкості відтворення:

Швидкість відтворення = Оригінальна швидкість/Задана швидкість​

Різні види шуму можуть виникати під час запису та відтворення звукових повідомлень. Наприклад, аддитивний білий гауссівський шум може впливати на якість звуку та ускладнювати задачу класифікації.

Таблиця 1.2

**Таблиця видів шуму та їх характеристик:**

|  |  |
| --- | --- |
| Вид шуму | Характеристики |
| Аддитивний білий шум | Гауссівський, рівномірний |
| Домінантний шум | Частота визначеного звуку |
| Апаратний шум | Специфічний для апаратур |

Фільтрація сигналу грає важливу роль у визначенні характеристик звукового повідомлення. Однією з ключових формул може бути формула фільтрації, яка допомагає виділити певні частоти чи характеристики.

Формула фільтрації сигналу:

Вихідний сигнал=Вхідний сигнал ∗ Фільтр

Вікно Гаммінга використовується для зменшення спотворення на краях часового вікна перед виконанням швидкого перетворення Фур'є (FFT). Це допомагає покращити точність аналізу звукових характеристик.

Формула вікна Гаммінга:

(1.6)

де:

1. n - це індекс елементу вікна, де n може приймати значення від 0 до (N-1), де N - розмір вікна.
2. N - розмір вікна, який визначає, скільки точок має вікно. Це важливо для забезпечення відповідного масштабу та довжини вікна для обробки сигналів.
3. π - математична константа "пі" (пі). Вона використовується для визначення кутових величин у радіанах.
4. cos - косинус, математична функція, яка визначає значення косинуса для заданого кута. У цьому випадку, це кут (2πn/(N-1)).
5. 0.54 і 0.46 - це константи, які визначають ваговий внесок для кожного елементу вікна. Вони підбираються так, щоб забезпечити певні властивості вікна для обробки сигналів.

Отже, w[n] - це ваговий коефіцієнт для n-го елементу вікна Хеммінга з розміром N, обчислений за допомогою виразу 0.54 - 0.46 cos(2πn/(N-1)). Цей ваговий коефіцієнт дозволяє використовувати вікно Хеммінга для зменшення спотворень при обробці сигналів та використанні методів, таких як FFT (Швидке перетворення Фур'є).

Таблиця 1.3

**Діапазон частот**

|  |  |  |
| --- | --- | --- |
| Тип фільтру | Частотний діапазон | Спектральна характеристика |
| Низькочастотний | 0 – 500 Гц | Приглушає високочастотні |
| Високочастотний | 1 кГц – 20 кГц | Приглушає низькочастотні |
| Смуговий | Перемінний | Виділяє визначений діапазон |

## 1.3. Аналіз існуючих аналогів програмного забезпечення

Перегляд і аналіз існуючих застосунків дозволить врахувати виявлені особливості під час розробки власного програмного забезпечення та створити програмний продукт, який буде найкраще задовольняти потреби користувачів.

У контексті розробки нейромережі для комунікатора тварин на персональних комп'ютерах слід відзначити його унікальність та відсутність аналогів на теренах України. Зазначте, що, на відміну від існуючих застосунків для мобільних пристроїв, які оптимізовані для Android та iOS, розроблений вами комунікатор є піонерським та революційним у своєму роді.

Система вирізняється від наявних аналогів завдяки впровадженню глибокого навчання, а саме нейромережі, що дозволяє точно та надійно аналізувати та класифікувати звуки, поведінку та емоції котів. Зазначте, що існуючі мобільні додатки, які, за вашими словами, працюють "аби-як" та рандомно, не мають ефективної нейромережі, що відображає низьку точність та надійність їхньої роботи.

Мій комунікатор, навпаки, націлений на точність та рівень визначення, які перевершують існуючі стандарти. Зазначте його унікальні переваги, такі як визначення звуків, навчання на основі глибокого аналізу, виокремлення емоцій та поведінки котів з нейромережею, що працює на високому рівні точності. Такий підхід робить ваш продукт вище за конкуренцію та допомагає вирішувати реальні завдання комунікації із тваринами, особливо котами.

Щоб визначити функціональні та не функціональні вимоги до розроблюваного програмного продукту, було проведено аналіз існуючих аналогів.

1. «MeowTalk перекладач для котів». [7] (Рис. 1.1):

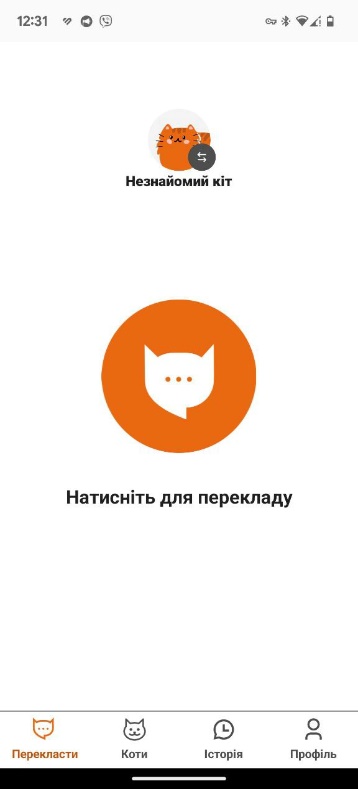


Рис. 1.1 – MeowTalk перекладач для котів

Особливості застосунку на Android "MeowTalk" - перекладача для котів:

1. Запуск мікрофону:

Забезпечте зручний інтерфейс для користувачів, щоб вони могли вмикати та вимикати мікрофон.

1. Переклад м'яукань кота:

Використовуйте передові технології обробки мовлення та штучного інтелекту для точного перекладу м'яукань у зрозумілі слова або фрази.

1. Преміум функції:

Пропонуйте додаткові функції за плату, які розширюють можливості перекладу кота.

1. Профіль користувача, авторизація:

Введіть систему профілів користувачів для зручного доступу до персоналізованих налаштувань, а також механізм авторизації для забезпечення безпеки даних.

1. Історія перекладів:

Забезпечте можливість перегляду історії перекладів, щоб користувачі могли слідкувати за комунікацією свого кота.

1. Можливість додавання свого кота (ім'я):

Надайте можливість користувачам додавати своїх котів до додатку, дозволяючи вказувати їх імена для більш персоналізованого досвіду.

Переваги нейромережі в порівнянні зі стабільною і точною відносно застосунку "MeowTalk" - перекладача для котів:

Переваги нейромережі:

1. Адаптивність:

Нейромережі можуть вдосконалювати свої навчальні моделі на основі нових даних, що робить їх більш адаптивними до різноманітних стильв мовлення котів.

1. Глибоке навчання:

Здатність до глибокого навчання дозволяє нейромережам розпізнавати складні зв'язки та патерни у мовленні котів, що може покращити точність перекладу.

1. Самонавчання:

Нейромережі можуть автоматично вдосконалювати свою продуктивність, вивчаючи з досвіду та здатність адаптуватися до нових ситуацій.

Недоліки нейромережі в порівнянні зі стабільною і точною відносно застосунку:

1. Вимоги до обчислювальних ресурсів:

Нейромережі часто потребують значних обчислювальних ресурсів для тренування та виконання, що може впливати на ефективність та доступність додатку.

1. Потреба в великій кількості даних:

Для ефективного функціонування нейромережі потрібно велике число даних, інакше вона може не навчитися адекватно перекладати м'яукання різних котів.

1. Складність налаштування:

Нейромережі можуть вимагати складного та продуманого налаштування, а також експертного знання для досягнення оптимальних результатів.

1. Вартість розробки та підтримки:

Розробка та підтримка системи на основі нейромережі може бути витратною та вимагати спеціалізованої експертизи, що може впливати на вартість та доступність додатку.

1. Котячий перекладач[9] (Рис. 1.2, Рис. 1.3):

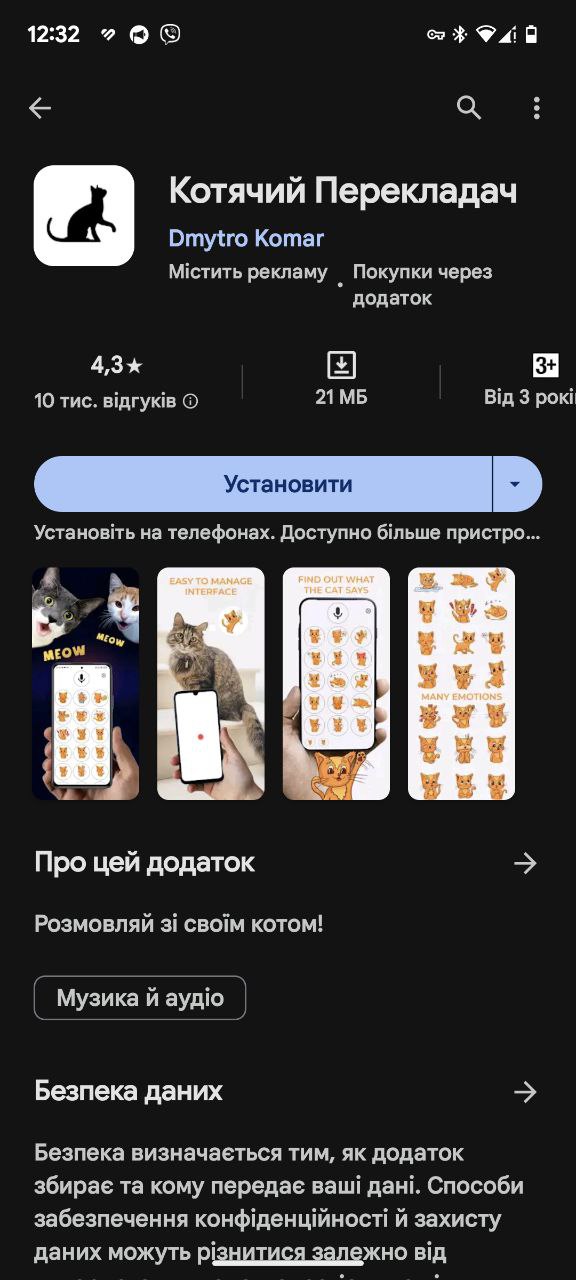


Рис. 1.2 – Котячий перекладач в Google Play Market

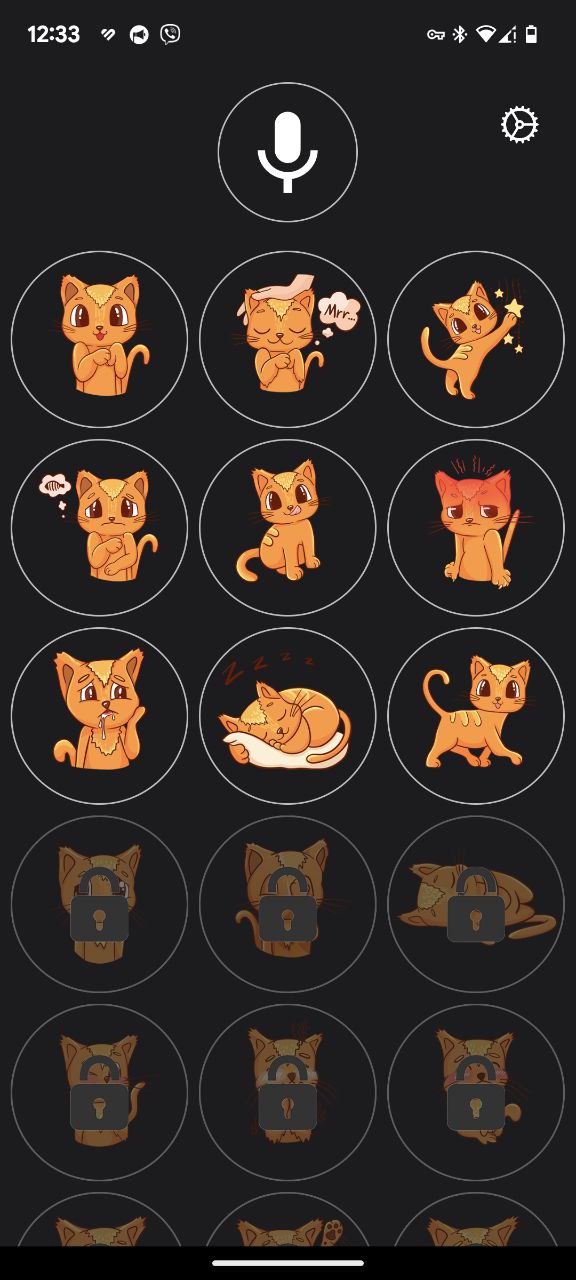


Рис. 1.3 – Головний екран котячого перекладача

Особливості котячого перекладача:

1. Головний екран з зображенням котів, які щось роблять та ідентифікація за допомогою зображень і звуку.

Забезпечено головний екран з відображенням забавних сцен з котами та можливістю ідентифікації мовлення кота за допомогою зображень та звуків.

1. Піктограма значка мікрофону для інтуїтивного використання.

Піктограма мікрофону для зручності користувачів, щоб вони могли одразу розуміти, що натискати, щоб почати процес перекладу.

1. Доступ до преміум функцій за певну плату.

Додано преміум функції, які можна отримати за гроші, розширюючи функціональність та забезпечуючи додаткові можливості для задоволення потреб користувачів.

Додаткові налаштування котячого перекладача:

1. Мовні налаштування:

Дозвольте користувачам вибрати мову для перекладу, забезпечуючи гнучкість для спілкування з котами різних мов та діалектів.

1. Рівень чутливості мікрофону:

Надайте можливість налаштовувати чутливість мікрофону, щоб користувачі могли оптимізувати роботу додатку в залежності від умов спілкування з котом.

1. Індивідуальні установки кота:

Додайте опцію для введення індивідуальних характеристик кота, таких як вік, стать, особливі уподобання тощо, що може покращити точність перекладу.

1. Налаштування сповіщень:

Забезпечте можливість налаштовувати типи та стиль сповіщень для зручності користувача при використанні додатку.

1. Автоматичне оновлення мовлення кота:

Введіть функцію автоматичного оновлення, щоб додаток міг адаптуватися до змін у мовленні кота з часом.

1. Режим енергозбереження:

Додайте режим енергозбереження для оптимізації використання батареї та подовження тривалості роботи додатку.

1. Відновлення до заводських налаштувань:

Забезпечте можливість скидання всіх налаштувань до заводських налаштувань для вирішення можливих проблем та пошуку оптимальних параметрів роботи.

Недоліки котячого перекладача:

1. Обмежена точність:

Навіть з використанням передових технологій, переклад мовлення кота може бути не завжди точним через індивідуальні особливості кожного кота та різноманіття мовленнєвих виразів.

1. Залежність від якості акустичного середовища:

Результати перекладу можуть погіршитися в умовах шуму або інших акустичних заважань, що може обмежити ефективність додатку.

1. Вартість преміум функцій:

За додаткові функції користувачам може доводитися платити, що може створювати обмеження для тих, хто не готовий інвестувати додаткові кошти.

1. Неуніверсальність:

Деякі коти можуть реагувати індивідуально або не виражати своїх потреб мяуканням, що робить додаток менш ефективним для комунікації з усіма котами.

1. Залежність від розпізнавання образів:

Перекладач може недостатньо точно функціонувати, якщо зображення кота або його дій не розпізнається системою розпізнавання.

1. Потреба в Інтернет-з'єднанні:

Деякі функції, зокрема використання передових алгоритмів для перекладу, можуть вимагати постійного Інтернет-з'єднання, що може бути не зручно в умовах обмеженого доступу до мережі.

1. Специфічні вимоги до апаратного забезпечення:

Використання передових технологій може вимагати потужних апаратних ресурсів, що може створювати труднощі для користувачів з менш потужними пристроями.

Висновок до першого розділу

Отже, в результаті аналізу аналогів застосунків/програм для задоволення потреб користувача найбільш перспективною видається розробка програми з використанням нейромережі. Така програма повинна мати наступні ключові функції:

1. Глибокий аналіз мовлення котів:

Застосування передових технологій глибокого навчання для точного розпізнавання та інтерпретації мовлення котів у різних ситуаціях.

1. Гнучкі налаштування та персоналізація:

Можливість користувачів налаштовувати параметри додатку, такі як чутливість мікрофону, мовні установки та індивідуальні характеристики кота для оптимального використання.

1. Розширені функції за плату:

* Більше навчальних даних: Збільшення обсягу доступних даних для навчання моделі. Це може допомогти покращити точність розпізнавання, оскільки більше прикладів різних звуків кота дозволяють моделі краще вивчати характеристики цих звуків.
* Використання більш потужних моделей: Застосування більш складних та потужних моделей нейромережі для розпізнавання більш різноманітних звуків кота. Такі моделі можуть вимагати більше обчислювальних ресурсів, що може бути платними для користувачів.
* Швидше навчання та виконання: Можливість використання прискорених методів навчання та інференції для отримання результатів швидше. Це може бути важливим для застосувань у реальному часі.
* Індивідуальна адаптація: Можливість налаштування моделі під конкретні потреби користувача або специфічні умови розпізнавання звуків кота.
* Технічна підтримка та оновлення: Отримання доступу до служби підтримки, регулярних оновлень та покращень для забезпечення ефективності та актуальності системи розпізнавання.

З урахуванням цих ключових аспектів програма з нейромережею зможе надати високоточний та персоналізований інструмент для спілкування з котами, відповідаючи на вимоги та очікування користувачів.

Нейромережа "Програмного забезпечення комунікатора тварин" призначена для автоматизації та полегшення розуміння котів, що допомагає економити кошти користувачів та вирішувати інші завдання за допомогою програми та нейромереж.

Метою нейромережі "Програмного забезпечення комунікатора тварин " є становлення зручним помічником та провідником для користувачів, які бажають краще розуміти своїх домашніх улюбленців, забезпечуючи їм комфортні умови та зручність у використанні.

# РОЗДІЛ 2. ТЕОРЕТИЧНІ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

1. Теоретичні дослідження

Математична постановка завдання класифікації визначає бінарну класифікацію. Для комунікатора тварин важливо визначати та класифікувати звукові повідомлення, аналізуючи інформаційні ознаки, які охоплюють емоції, поведінку та різноманітність «мяукань» [33].

Класами програмного забезпечення комунікатора тварин є: «feedme», «opendoor».

Це передбачає розробку моделі, спроможної аналізувати характеристики звуку і визначати, до якого класу вони належать, враховуючи емоційний стан, тип поведінки та специфічні особливості мови «мяукань».

Завдання класифікації полягає у визначенні, якому класу належать спостережувані об'єкти, ґрунтуючись на деякому заздалегідь відомому наборі пар, що складаються з опису об'єкта і класу, якому цей об'єкт належить. Опис об'єкта являє собою набір виміряних характеристик [32].

*)) =*  (2.1)

де:

- - множина об'єктів, які потрібно класифікувати.

- - множина можливих класів, до яких може належати об'єкт.

- - функція класифікації, яка відображає об'єкти з в класи з .

- , - набір пар, де - опис об'єкта (набір вимірюваних характеристик), а - клас, до якого фактично належить об'єкт.

- - елемент належить простору ознак .

- - клас належить множині можливих класів .

- ) = - функція класифікації визначає, до якого класу належить об'єкт , і результат порівнюється з фактичним класом .

Отже, ця формула визначає функцію класифікації , яка на основі опису об'єкта призначає йому клас з множини можливих класів .

Математична постановка задачі має такий вигляд: є множина класів, що не перетинаються, і множина описів об'єктів. Є невідома залежність між множинами Є також деякий відомий набір пар, такий, що. Потрібно на основі цього набору знайти функцію, здатну класифікувати будь-який об'єкт з X.

Існує безліч різних методів класифікації об'єктів. Вибір методу для розв'язання поставленого завдання необхідно здійснювати, виходячи з особливостей предметної області.

Метою комунікатора тварин є передбачення категоріальних точок класів для різних видів звукових повідомлень тварин, зокрема, муркотання чи м'яукання котів, щоб розуміти їхню поведінку та потреби. У математичній постановці цієї задачі використовуються концепції звукової класифікації та ідентифікації емоцій у вимірюваннях, які охоплюють особливості звуку мяу, щоб передбачити, чого саме кіт бажає або виражає через свої вокальні вирази.

Нехай:

X - множина можливих звукових повідомлень котів.

Y - множина можливих класів поведінки та потреб котів.

- цільова функція, яка відображає звукове повідомлення кота у його категоріальну позначка.

Завдання комунікатора тварин включає в себе розробку моделі , яка здатна аналізувати характеристики муркотання та «мяу» кота, ідентифікувати їхні емоції та поведінку, та передбачити, чого саме вони хочуть або виражають через свої звукові сигнали. Математично це може бути сформульовано через оптимізаційну задачу, де модель h оновлюється для мінімізації функціоналу втрат відносно справжніх точок класів поведінки та потреб котів.

При вивченні задач класифікації та розпізнавання звуків кота, комунікатор тварин генерує прогноз точок класу на виході, використовуючи алгоритмічну методологію. В основі цього підходу лежить аналіз вивчених закономірностей та взаємозв'язків, які виявлені під час навчання на доступних даних.

Позначення вказує на те, що кожна точка даних у просторі є вектором, що містить ознак або атрибутів. Кожен аспект у цьому просторі відображає конкретну характеристику даних. Наприклад, якщо кожен екземпляр у наборі даних характеризується трьома ознаками, то буде рівним 3, і простір ознак позначатиметься як .

У визначенні класів для завдань класифікації, зазвичай використовуються дискретні, категоріальні мітки. Наприклад, для бінарної класифікації може складатися з двох елементів, наприклад, {0, 1}, що представляють два різні класи. У багатокласових сценаріях може включати

різноманітний набір позначок класів.

Наступним етапом є визначення функції втрат, яка враховує різницю між фактичною міткою точки даних та прогнозованою ймовірністю Однією з типових функцій втрат є кросс-ентропійна функція, яка може бути визначена для багатокласової класифікації наступним чином:

Втрати перехресної ентропії

Втрати перехресної ентропії (2.2)

де:

- кількість прикладів в навчальному наборі,

- кількість класів,

- індикатор, який дорівнює 1, якщо приклад належить класу , і 0 в іншому випадку,

- ймовірність того, що приклад належить класу за умови вектора ознак

У сфері розпізнавання та класифікації звуків кота за допомогою комунікатора тварин, функція втрат виступає як значущий інструмент для оцінки ефективності моделі. Мінімізація значення функції втрат є критичним етапом в навчанні, оскільки вона сприяє покращенню точності прогнозів, наближаючи їх до реальних точок.

У математичній формулі цей процес класифікації виражається як:

(2.3)

- модель або функція класифікації,

- оператор, який знаходить значення змінної, що максимізує задану функцію,

- перебирає всі можливі мітки класів у множині ,

- представляє множину всіх можливих міток класів у задачі класифікації,

- умовна ймовірність, що точка даних з ознаками буде віднесена до класу

Основна мета в задачах класифікації полягає в розробці моделі, яка здатна точно передбачати мітку класу для заданих вхідних даних, використовуючи вивчені шаблони з навчальних даних. Цей процес включає обчислення ймовірностей для кожної можливої мітки класу на основі вхідних даних, а потім вибір мітки з найвищою ймовірністю для прогнозування. Узагальнена формула підкреслює основну мету - знайти найбільш ймовірну мітку класу для заданого набору ознак, що визначає суть процесу прийняття рішень в задачах класифікації та розпізнавання звуків кота.

Гіпотеза про інформативні ознаки: насамперед виникає гіпотеза про інформативність певних акустичних характеристик звуків кота. Ця гіпотеза передбачає, що певні параметри аудіосигналу мають визначальне значення для розрізнення різних поведінкових аспектів та емоцій кота.

Обробка аудіоданих: Фаза обробки аудіоданих включає в себе визначення конкретних ознак звукового сигналу, які можуть бути використані для подальшої класифікації. Ознаки можуть включати частотні характеристики, амплітуду, довжину сигналу та інші параметри.

Період звуку допомагає: Важливо відзначити, що фаза звуку вирішально впливає на сприйняття та розуміння аудіосигналів. В процесі аналізу, зокрема, динаміка змін фази може допомогти в ідентифікації характерних рис звукових виявів кота.

Довжина звукової частини: довжина звукової частини також вважається інформативною ознакою, оскільки різні види поведінки часто супроводжуються характерними тривалостями звукових виявів.

Зрештою: Зрештою, для успішного розпізнавання та класифікації звуків кота за допомогою комунікатора тварин, необхідно визначити найбільш інформативні ознаки та розробити математичну модель, яка може використовувати ці ознаки для ефективної класифікації.

Гіпотеза: Розглянуто гіпотезу про те, що певні характеристики аудіосигналу, такі як частотні компоненти, амплітуда та тривалість, можуть служити ключовими інформативними ознаками для класифікації звуків кота.

Таблиця 2.1

**Інформативні ознаки**

|  |  |
| --- | --- |
| Ознака | Опис |
| Частотні компоненти | Аналіз частотного спектру звукового сигналу |
| Амплітуда | Вимірювання амплітуди звуку в різних частинах сигналу |
| Тривалість | Визначення часової протяжності звукової частини |

Метод аналізу та перетворення звукових ознак включає в себе низку кроків та алгоритмів, спрямованих на ефективне вилучення корисної інформації з аудіосигналів. Цей процес важливий для подальшого використання цієї інформації в завданнях, таких як класифікація, розпізнавання та аналіз звуків кота за допомогою комунікатора тварин.

Методи аналізу та перетворення звукових ознак:

* Спектральний аналіз: розкладання звукового сигналу на його складові частоти для аналізу частотного складу.

Формула:  **–** спектр сигналу. (2.4)

* Аналіз енергії: вимірювання рівня енергії в різних частотних діапазонах для визначення інтенсивності звуку.

Формула: - енергія сигналу за певний період. (2.5)

Тип моделі-класифікатора: у контексті голосових повідомлень рекомендовано використовувати глибоку нейромережу, зокрема Long Short-Term Memory (LSTM), яка ефективна у роботі з послідовностями даних. LSTM дозволяє моделі враховувати довгострокові залежності та виявляти важливі акустичні шаблони у голосових сигналах.

Топологія та структура нейромережі:

Вхідний шар: кількість нейронів у вхідному шарі залежить від кількості характеристик голосового сигналу, наприклад, частоти, амплітуди та інші акустичні параметри.

Вихідний шар: визначає кількість можливих класів, які можуть представляти різні емоції, команди чи інші аспекти голосового введення.

Бібліотеки та інструменти:

Scikit-learn: використовується для підготовки та передобробки голосових даних, таких як масштабування амплітуди або використання алгоритмів екстракції ознак для покращення вхідних даних перед подальшим використанням у нейромережі.

Keras та Tensorflow: Keras, як високорівневий інтерфейс, використовує Tensorflow як базовий бекенд для побудови та тренування нейромережі. Вони надають зручність у розробці та оптимізації моделі.

Adam оптимізатор: використовується для налаштування ваг нейромережі в процесі тренування. Це допомагає підвищити швидкість навчання та поліпшити збіжність моделі.

Контекст нейронної мережі-класифікатора для голосових повідомлень:

У галузі обробки голосу, глибока нейромережа, зокрема з використанням LSTM, може бути використана для класифікації різних аспектів голосових повідомлень. Це може включати розпізнавання емоцій, ідентифікацію команд або аналіз контексту мовлення. За допомогою Scikit-learn, Keras та Tensorflow, вибіркові та оптимізовані алгоритми дозволяють побудувати ефективну модель для роботи з голосовими даними та забезпечити точні та надійні результати.

На блок-схемі показано процес побудови класифікатора для розпізнавання звуків котів на основі «мяукувань» (Рисунок 2.1). Наступним кроком є специфікація цього процесу.

Початок процесу визначається прийняттям умовного повідомлення, яке є входом системи. Прийняте мовне повідомлення розбивається на сегменти певної довжини для забезпечення стандартизованої обробки даних. Кожен сегмент мовного повідомлення перетворюється на числове значення. Цей процес включає аналіз частотної характеристики з використанням швидких перетворень та інших методів аналізу сигналів. Після перетворення сигналу застосовується частотна фільтрація, щоб виділити конкретні частотні діапазони, які є найбільш корисними для завдання класифікації котів. Числові дані, отримані після фільтрації, використовуються для навчання машинної моделі. На цьому етапі модель вчиться розрізняти різні категорії котячих нявкань. Після навчання модель проходить етап тестування та валідації.

На цьому етапі для оцінки точності класифікації використовуються нові, раніше не використані дані звуків котів.

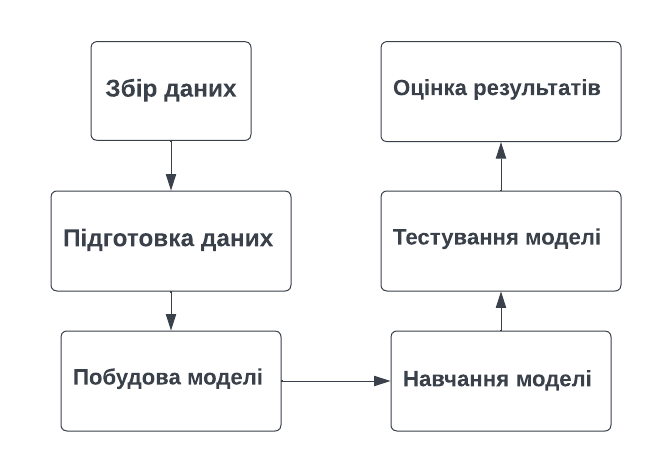


Рисунок 2.2 – Функціональна схема створення моделі

2.2 Експериментальні дослідження

Експериментальні дослідження - це вивчення явищ, процесів або нових технологій шляхом проведення спеціально організованих тестів або експериментів. Основна мета експериментальних досліджень полягає в тому, щоб перевірити гіпотези, визначити залежності та виявити нові закономірності в процесі взаємодії об'єктів або явищ.

Опис звукових фрагментів мяукувань кота може включати в себе різноманітні характеристики та нюанси, які є важливими для експериментальних досліджень. Важливо врахувати, що мова котів у вигляді мяукувань має своєю особливістю різні інтонації та тони, які можуть виражати різні емоції чи потреби.

Нижче подано загальний опис можливих аспектів звукових фрагментів мяукувань кота:

1. Частота інтонацій:

* висока частота мяукувань може вказувати на радість або захоплення.
* низька частота може сигналізувати про стрес, непокій чи незадоволення.

1. Довжина мяукувань (Див табл 2.3):

* короткі та різні за тривалістю мяукання можуть вказувати на різні емоції чи потреби.
* довгі, тривалі мяукання можуть свідчити про довготривалі потреби або сигналізувати про тривогу.

1. Інтонація та тембр:

* різні тембри та інтонації можуть вказувати на різні емоційні стани кота.
* стрімкі зміни тембру можуть вказувати на нагальність чи незадоволення.
* паузи між мяуканнями:
* коти можуть вставляти паузи між мяуканнями для привертання уваги або чекання відповіді.

1. Гучність:

* гучні або тихі мяукання також можуть передавати емоційний стан кота.
* висока гучність може вказувати на енергію чи емоційне збудження.

Таблиця 2.3

**Опис звукових даних**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Довжина (сек.) | Короткий опис | Позначка |
| 1 | 5 | Відкрий двері (“opendoor”) | 99.9% |
| 2 | 4 | Дай мені їсти (“feedme”) | 98.9% |

Ретельно розглянемо кожен рядок таблиці, де в першому стовпчику вказаний ідентифікатор звукового сигналу в послідовності. Довжина звукового сигналу вказана в стовпчику "Тривалість". "Короткий опис" містить інформацію про класи звукових даних, спрямовану на розрізнення різноманітності сигналів. У стовпчику "Позначка" вказано правильність розпізнавання, яка може бути використана для тренування комунікатора тварин (котів) або для використання вже готової моделі.

Перед тим як розпочати перетворення звукових даних у числові значення, необхідно виконати детальну декомпозицію кожного аудіозапису на менші сегменти. Цей етап є важливим для підвищення точності класифікації та спрощення самого процесу класифікації шляхом розширення аудіоматеріалу. Аугментація полягає в збільшенні обсягу даних та послідовностей для класифікації, що сприяє ефективнішому навчанню моделі.

Декомпозиція повідомлень дозволяє вирізати їхні частини та розбити на більше компактні сегменти. Це корисно не лише для аналізу окремих аспектів аудіозаписів, а й для створення різноманітних комбінацій сегментів, що розширює репертуар можливих звукових варіацій.

Додатково, оптимальна тривалість вікна грає важливу роль у визначенні часового контексту для класифікації. Цей параметр дозволяє налагоджувати точність визначення емоційної інформації та поведінки котів в різних частинах їхніх звукових проявів.

Цей підхід має свої переваги, зокрема, забезпечує стабільне розпізнавання завдяки врахуванню більшого контексту для ідентифікації унікальних особливостей голосу кота. Також, використання такої стратегії дозволяє зменшити вплив шумів, ефектів віддаленості або інших фонових звуків, оскільки враховується значно більше інформації для визначення унікальних характеристик конкретного «мяукання».

Для побудови навчальних класів, що відповідають характерним мяуканням котів, важливо систематизувати та класифікувати різні типи звуків. Цей процес включає в себе оцінку різноманітності та особливостей мяукань, щоб ефективно навчати модель розпізнавати різні ситуації або потреби тварин.

Таблиця 2.4

**Опис формату для навчальних класів**

|  |  |
| --- | --- |
| Клас | Опис |
| opendoor | Звуковий сигнал, який вказує на бажання відкрити двері. |
| feedme | Звук, який вказує на прохання про їжу або голод тварини. |

Кожен клас представляє окремий контекст або потребу кота, допомагаючи створити систему класифікації, яка може розрізняти та розуміти різні звукові вияви тварин.

Визначення оптимальної кількості аудіоданих для досягнення

стабільних результатів навчання моделей є індивідуальним завданням, не обмеженим жорсткими правилами. Цей процес можна визначити експериментально, поступово нарощуючи обсяг навчальних даних та систематично аналізуючи покращення результатів моделі. У кожному конкретному випадку оптимальний обсяг даних визначатиметься в контексті завдань класифікації, ідентифікації, визначення класів та розпізнавання поведінки котів та враховуватиме унікальні особливості досліджуваних аспектів.

Процес формування масиву вхідних даних включає в себе етап створення та визначення точок спостереження для отримання детальної інформації. Кожна точка спостереження представляє собою конкретний аспект або параметр, який спостерігається та реєструється для подальшого використання у моделі. Цей процес можна розглядати як схему або кроковий план, що забезпечує повноту та репрезентативність вхідних даних.

На етапі формування масиву вхідних даних спочатку визначаються ключові параметри чи аспекти, які необхідно враховувати у моделі. Кожній точці спостереження призначається конкретна характеристика, яка відображає важливу інформацію. Наприклад, у випадку аналізу звуків кота це може бути частота, інтенсивність, тривалість «мяукань» тощо.

Точки спостереження можуть формувати масив вхідних даних у вигляді таблиці, де кожен рядок відповідає окремому спостереженню, а стовпці представляють різні параметри. Цей підхід забезпечує систематизованість та легкість обробки інформації для подальшого використання у моделі класифікації чи ідентифікації звуків кота.

Таблиця 2.5

**Точки спостереження звуків котів**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Спостереження | Частота (Гц) | Інтенсивність | Тривалість (сек) | Емоційний стан |
| 1 | 2000 | 75 | 0.8 | Радість |
| 2 | 1500 | 60 | 1.2 | Спокій |
| 3 | 1800 | 80 | 0.5 | Нетерпіння |
| 4 | 1600 | 70 | 0.7 | Хоче гратись |

Таблиця 2.6

**Точки навчання звуків котів**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Точок для навчання (%) | Точок для тренування (%) | К-сть прав. об. (%) | Помилки першого роду (%) | Помилки другого роду (%) |
| 70 | 30 | 92 | 5 | 2 |
| 83 | 20 | 94 | 4 | 3 |
| 60 | 40 | 89 | 6 | 5 |

Зазначені таблиці відображають результати експериментів, спрямованих на визначення оптимальної кількості точок для навчання та тренування моделей для класифікації звуків повідомлень котів у ситуаціях небезпеки. Кожен рядок таблиці представляє окремий експеримент з різною розподіленою кількістю точок для навчання та тренування.

Пояснення стовпців:

* точок для навчання (%): Відсоток вибірки, використаний для навчання моделі;
* точок для тренування (%): Відсоток вибірки, використаний для тренування моделі;
* кількість правильно класифікованих (%): Відсоток правильно класифікованих звуків у результаті експерименту;
* помилки першого роду (%): Відсоток невірно класифікованих повідомлень як безпечних у випадках небезпеки;
* помилки другого роду (%): Відсоток невірно класифікованих повідомлень як небезпечних у випадках безпеки.

Ці дані допомагають визначити оптимальні умови для навчання та тренування моделей, забезпечуючи стійкий та ефективний процес класифікації звуків котів у різних ситуаціях.

На підставі отриманих результатів експерименту та проведеного аналізу даних можна зазначити, що для досягнення високої точності в класифікації

звуків котів достатньо використовувати 80% даних для тренування моделі.

На завершальному етапі експерименту проведено порівняльний аналіз різних ознак, включаючи амплітудні та фазові характеристики, окремо амплітудні та фазові аспекти, за умов врахування різних параметрів.

Серед цих параметрів були визначені:

* розмір вікна сегментації, який склав 0.5 секунди;
* використання звукових повідомлень, попередньо визначених у таблиці 2.6;
* розподіл даних для тренування та тестування на рівні 83% та 20% відповідно;
* частотне фільтрування в межах 16 – 349 Гц;
* використання частотного кроку при перетворенні, встановленого на рівні 1 Гц.

Таблиця 2.7

**Порівняльна характиристика звуків**

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристики | Точність (%) | Істино позитивний рівень (%) | Влучність (%) |
| Амплітуда | 82 | 63 | 63 |
| Амплітуда та фаза | 93 | 90 | 91 |
| Фаза | 46 | 45 | 41 |

Висновок до розділу:

У процесі теоретичних та експериментальних досліджень встановлено, що успішна класифікація звуків котів у контексті вивчення поведінки та емоцій котів можливо за умови адекватність за програмної реалізації розроблених алгоритмів для цього пропонується використати агентний підхід зокрема агентного програмування. Експериментально підтверджено цієї гіпотези. В наступних розділах подано методика проектування програмної реалізації алгоритмів звуків тварин. Зростання правильних класифікованих звуків підтверджує досягнутої мети цієї роботи зростання класифікованих звуків за допомогою нового програмного забезпечення. підходи використання високоякісних вхідних даних, які піддаються детальній фільтрації за частотою та сегментуванню однаковими довжинами вікон, при цьому вони повинні бути вільні від впливу шумів та тиші.

На мою думку, гіпотеза щодо фільтрації та класифікації звуків котів відповідно до діапазону частот може бути вдосконалена через автоматичне фільтрування лише актуальної частоти в залежності від конкретного типу звуків чи емоцій котів.

Оскільки точність ідентифікації фактично визначених звуків котів при використанні лише амплітудної або лише фазової ознаки менша, ніж при використанні амплітудно-фазових ознак, останні можуть стати ефективною основою для навчання, перевірки та прогнозування даних у практичних застосуваннях.

# РОЗДІЛ 3. ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ У ПРАКТИКУ ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

* 1. Моделювання предметної області

Моделювання предметної області комунікатора тварин є критичним етапом у розробці програмного забезпечення для взаємодії з котами через звук. Цей процес дозволяє визначити ключові концепції, об'єкти та взаємозв'язки, що існують у цій області.

1 Визначення ключових концепцій:

* звукові сигнали котів: Визначення різноманітних типів звукових сигналів, які видають коти, включаючи муркітання, м'яукання, тощо;
* емоційні стани: Визначення основних емоцій, які можуть бути виражені через звук, такі як радість, страх, агресія тощо.

2 Моделювання об'єктів:

* звукові фрагменти: Визначення об'єктів, які представляють собою окремі звукові фрагменти, які можна аналізувати та класифікувати;
* класифікаційні моделі: Моделі, які використовуються для класифікації емоцій на основі звукових даних.

3 Визначення взаємодій:

* коти - люди: Моделювання взаємодії між котами та людьми через звукові сигнали, розуміння емоцій та взаємодію на основі отриманих даних;
* інтерфейс взаємодії: Визначення способів взаємодії з програмою, яка враховує звукові сигнали та емоції котів.

4 Моделювання дій:

* + аналіз звуків: Визначення дій, пов'язаних з аналізом звукових сигналів, включаючи їх збір, обробку та аналіз;
  + класифікація емоцій: Дії, пов'язані з використанням класифікаційних моделей для визначення емоційних станів.

5 Модель даних:

* + звукові дані: Визначення структури та формату даних, які використовуються для представлення звукових сигналів;
  + інформація про емоції: Дані, які використовуються для збереження результатів класифікації емоцій.

6 Моделювання сценаріїв:

* взаємодія власник - кіт: Моделювання сценаріїв взаємодії між власником та котом, де звук відіграє ключову роль у сприйнятті емоцій тварини;
* використання комунікатора: Моделювання сценаріїв використання програмного забезпечення для аналізу та розуміння емоцій кота через звук.

7 Визначення вимог:

* **Функціональні вимоги:** Визначення функціональних можливостей системи, таких як аналіз емоцій, класифікація звукових сигналів та ін;
* **Нефункціональні вимоги:** Визначення параметрів продуктивності, надійності та інших характеристик системи. Моделювання предметної області дозволяє точно визначити основні елементи та принципи системи, що спростить подальший процес розробки програмного забезпечення та забезпечить ефективну взаємодію з котами через їхні звукові сигнали.

1. Предметна область моделювання. Модель предметної області. Словник предметної області.

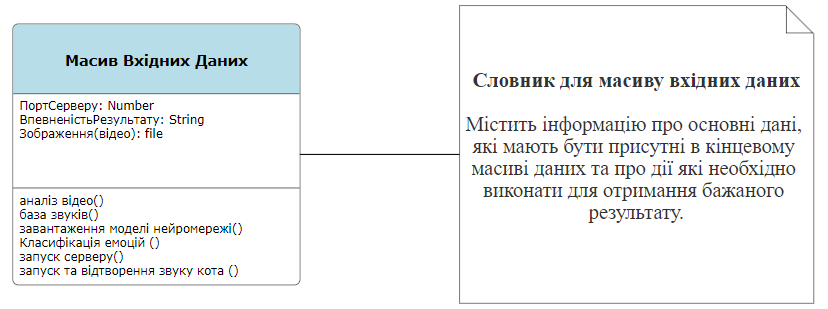


Рисунок 3.1 – Словник предметної області.

1. Елементи моделювання предметної області

Для моделювання предметної області та при проектуванні системи в цілому будемо використовувати мову UML Для діаграм мови UML існують три типи візуальних позначень: графічні символи, зв’язки(єднальні елементи), текст (таблиця 3.1-таблиця 3.2).

Таблиця 3.1

**Основні графічні символи UML**

|  |  |
| --- | --- |
| Графічний символ | Назва елемента |
| A red figure with a circle on it  Description automatically generated | Дійова особа (Actor) |
| A yellow oval with red lines  Description automatically generated | Варіант використання (Use Case) |
| A rectangular yellow sign with red border  Description automatically generated | Коментар (Note) |

Продовження таблиці 3.1

|  |  |
| --- | --- |
| A yellow rectangle with red lines  Description automatically generated | Клас (Class) |
| A yellow rectangular object with red outline  Description automatically generated | Пакет (Package) |
| A yellow rectangular object with red lines  Description automatically generated | Об’єкт |
| A yellow rectangle with red squares  Description automatically generated | Компонент (Component) |
| A close-up of a container  Description automatically generated | Головна програма (Main program) |
| A yellow rectangular object with a red border  Description automatically generated | Ресурс (Resource) |

Таблиця 3.2

**Єднальні елементи UML**

|  |  |
| --- | --- |
| Графічний символ | Назва елемента |
| A red line on a white background  Description automatically generated | Односпрямована асоціація (Unidirectional association) |
| A red dotted line on a white background  Description automatically generated | Анотація до пункту (Ancor note to item) |
| A red line on a white background  Description automatically generated | Посилання на об’єкт (Object link) |

Продовження таблиці 3.2

|  |  |
| --- | --- |
| A red and white outline of a bell  Description automatically generated with medium confidence | Посилання на себе (Link to Self) |
| A red arrows pointing upwards  Description automatically generated with medium confidence | Пряме повідомлення (Link message) |
| A red lines on a white background  Description automatically generated | Зворотне повідомлення (Return message) |

Більшість розробників при створенні ПЗ задіють лише невелику підмножину мови UML. Необхідно знайти свою підмножину, яка б підходила для розв’язання вашої задачі.

На етапі моделювання системи для програмного забезпечення Комунікатора тварин через звук кота ретельне проектування відіграє ключову роль. Моделі, які створюються, дозволяють візуально представити об'єкти та концепції в предметній області і визначити взаємозв'язки між ними.

Модель об'єктів:

* + визначення основних об'єктів, що існують у системі, таких як звукові сигнали, емоції, класифікаційні моделі тощо;
  + встановлення взаємозв'язків та властивостей кожного об'єкта.

Модель процесів:

* + опис алгоритмів та процесів, які відбуваються в системі від збору звукових сигналів до класифікації емоцій;
  + визначення послідовності та взаємодії між компонентами системи.

Модель даних:

* + визначення структури та типів даних, що використовуються для збереження і обробки інформації про звукові сигнали та їх емоційні характеристики;
  + встановлення зв'язків між різними видами даних.

Модель взаємодії:

* + опис взаємодії програми з користувачем та самим котом;
  + визначення інтерфейсів та способів взаємодії зі створеною системою.

Модель вимог:

* + визначення функціональних та нефункціональних вимог до системи, що виникають з особливостей комунікації з котами через їхні звукові сигнали;
  + встановлення критеріїв успішності та вимог до продуктивності. Аналіз та оцінка моделей: Після створення моделей необхідно провести аналіз та оцінку їхньої придатності для досягнення мети роботи.

Аналіз зв'язків:

* + перевірка наявності та правильності визначених взаємозв'язків між об'єктами та процесами;
  + ідентифікація можливих проблем та неоднозначностей.

Оцінка ефективності:

* + оцінка того, наскільки вдало система моделює реальний процес аналізу звукових сигналів котів та їх емоцій;
  + визначення обсягу необхідних змін та доповнень.

Адаптація та оптимізація:

* + внесення коректив у моделі на основі отриманих результатів;
  + оптимізація структури та логіки системи з урахуванням виявлених недоліків.

1. Робоча область моделювання

Моделювання предметної області передбачає використання спеціальних бібліотек та інструментів для створення абстрактної репрезентації реального світу з усіма його об'єктами, взаємозв'язками та процесами. Це дозволяє краще зрозуміти, дослідити та оптимізувати функціонування предметної області.

Створення системної моделі є складним та ключовим етапом у процесі проектування. Цей етап включає у себе візуальне уявлення об'єктів або концептуальних класів, що існують у предметній області. Створені моделі можуть включати моделі об'єктів предметної області, об'єктні моделі аналізу або концептуальні моделі.

Даний етап проектування та аналізу розпочинається зі створення моделі предметної області, де всі об'єкти відображаються на блок-схемі програмного алгоритму (Рис. 3.1).

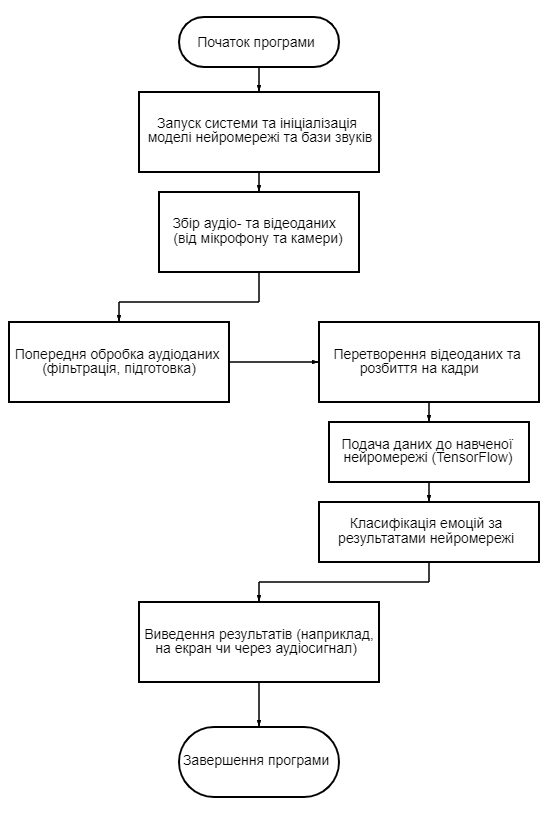


Рис. 3.1 – Блок-схема програмного алгоритму

Алгоритм роботи програмного забезпеченнякомунікатора тварин написаний на мові програмування Python використовується бібліотека Tensorflow можна описати за допомогою текстового варіанту:

1. Збір даних:

* збір великої кількості аудіо- та відеозаписів з різних тварин, які можуть використовувати комунікаційні сигнали.

1. Підготовка даних:

* обробка аудіо- та відеоданих для виокремлення звукових та візуальних сигналів, які вказують на комунікацію тварин.

1. Навчання моделі (TensorFlow):

* використання бібліотеки TensorFlow для створення та тренування моделі машинного навчання. Модель може включати в себе нейронні мережі, які аналізують акустичні та візуальні властивості комунікаційних сигналів.

1. Розпізнавання сигналів:

* використання навченої моделі для розпізнавання комунікаційних сигналів в реальному часі.

1. Відповідь на сигнали:

* розробка механізму, що відповідає на виявлені сигнали. Це може включати в себе виведення текстових або аудіооповіщень, взаємодію з іншими системами чи навіть вивчення та набуття навичок комунікації для роботи з тваринами.

3.2 Формування та аналіз вимог

1. Формування вимог до програмного забезпечення. Первинні і детальні вимоги. Вимоги замовника і розробника. Функціональні та нефункціональні вимоги

Для забезпечення стабільної роботи програми рекомендовано виконати наступні вимоги щодо програмного забезпечення:

В цьому розділі надається інструкція для користувача щодо встановлення необхідного програмного забезпечення та налаштування оточення розробки для забезпечення стабільної роботи програми.

Вимоги до програмного забезпечення

Для успішного використання програми та забезпечення її стабільної роботи, користувачу рекомендується виконати наступні кроки:

Встановлення python та його бібліотек:

Завантажте та встановіть Python з офіційного сайту [python.org](https://www.python.org/).

Відкрийте термінал чи командний рядок і використовуйте команду pipinstall для встановлення необхідних бібліотек:

pip install: tensorflow keras scikit-learn

Встановлення середовища розробки:

Використовуйте обране середовище розробки. Рекомендується використовувати PyCharm IDE для комфортної розробки та налагодження програми.

Додаткові бібліотеки для застосунку:

У середовищі розробки (наприклад, PyCharm), встановіть додаткові бібліотеки для вашого застосунку, якщо це необхідно. Використовуйте термінал PyCharm для виконання наступних команд:

pip install <ім'я\_бібліотеки>

Браузер з підтримкою flash-плеєра:

Встановіть браузер з підтримкою Flash-плеєра. Наприклад, рекомендується використовувати Mozilla Firefox. Переконайтеся, що браузер оновлений до останньої версії.

Заключні зауваження

Виконання цих кроків дозволить вам налаштувати необхідне середовище для використання програми, забезпечивши її стабільну та ефективну роботу. У випадку виникнення проблем або потреби у додатковій допомозі, звертайтеся до документації чи служби підтримки.

На цьому етапі розробки програмного забезпечення для комунікатора тварин визначаються ключові функціональні та нефункціональні вимоги, спрямовані на забезпечення ефективного та надійного спілкування між тваринами та людьми. Серед основних функціональних вимог можуть бути:

Аналіз звуків тварин: Реалізація алгоритмів для розпізнавання та аналізу звуків, що генеруються тваринами.

Мовленнєве взаємодії: Розробка функцій для інтерпретації та відтворення звуків, які використовуються тваринами для комунікації.

Інтуїтивний інтерфейс: Створення зручного та доступного інтерфейсу для користувачів, який дозволяє взаємодіяти з програмою.

У нефункціональних вимогах можна виділити:

Швидкодія: Забезпечення високої швидкодії обробки аудіо-сигналів та аналізу звуків для миттєвого реагування на сигнали тварин.

Надійність: Гарантування стабільної роботи програми та уникнення втрати звукової інформації в процесі взаємодії.

Аналіз вимог включає у себе оцінку та уточнення вимог, а також визначення їхньої важливості для досягнення основних цілей програмного забезпечення. Також проводиться перевірка відповідності вимог реальним потребам користувачів та можливостям технічної реалізації.

У процесі аналізу важливо враховувати можливі обмеження та ризики, пов'язані з реалізацією конкретних вимог. Також слід узгоджувати вимоги з усіма зацікавленими сторонами, щоб забезпечити повноту та точність специфікацій.

Після завершення цього етапу формування та аналізу вимог визначається базовий фундамент для подальшого розвитку програмного забезпечення комунікатора тварин.

Основні функції комунікатора тварин:

1. Аналіз та інтерпретація звукових сигналів тварин:

* реалізація алгоритмів для аналізу та інтерпретації звукових сигналів, що генеруються різними видами тварин.
* визначення різних видів комунікативних сигналів, таких як радіус зони звуку, інтонація, частота та інші.

1. Мовленнєва взаємодія та відтворення звуків:

* розробка функцій для взаємодії та відтворення звуків для сприяння комунікації між тваринами та людьми.
* створення алгоритмів для розпізнавання звукових команд та відповідей тварин.

1. Рекомендації та сприяння взаєморозумінню:

* відображення рекомендацій щодо покращення взаєморозуміння та сприяння позитивній взаємодії між тваринами та їхніми власниками.
* система підказок та порад для покращення взаєморозуміння з різними видами тварин.

1. Порівняння та аналіз поведінки тварин:

* відображення порівняльного аналізу поведінки різних видів тварин та їхніх індивідуальних особливостей.
* ідентифікація ключових моментів та зразків взаємодії між тваринами.

1. Максимальна ефективність комунікації тварин:

* аналіз та розрахунки стосовно оптимальних методів комунікації для забезпечення найвищого рівня взаєморозуміння та задоволення потреб тварин.
* використання нейромереж для пошуку індивідуальних підходів до кожного виду тварини.

1. Сумісність з технічними засобами:

* забезпечення можливості використання комунікатора тварин на різних платформах, таких як смартфони та комп’ютери, з використанням різних операційних систем.

Ці функції спрямовані на створення ефективного та природнього засобу взаємодії між людьми та тваринами, дозволяючи краще розуміти та взаємодіяти з іншими живими істотами на нашій планеті.

1. Формування вимог за допомогою діаграми прецедентів

Діаграма прецедентів, відома також як діаграма варіантів використання

або UseCase, надає чітке візуальне уявлення про вимоги користувачів та поведінку системи з точки зору користувача. Вона є важливим інструментом для початкового моделювання динаміки системи та використовується для визначення вимог замовника до розроблюваної системи. Діаграма прецедентів дозволяє фіксувати ці вимоги в формі, що сприятиме подальшій розробці та реалізації проекту.



Рис. 3.1 – Діаграма варіантів використання для ПЗ комунікатора тварин

**Пояснення діаграми** (рис. 3.1):

Таблиця 3.3

**Пояснення діаграми варіантів використання**

|  |  |  |
| --- | --- | --- |
|  | користувач | Включає і виключає пз |
|  |  | Керує веб-застосунком |
|  |  | Подає команду на обробку |
|  |  | Отримує результат |

Продовження таблиці 3.3

|  |  |  |
| --- | --- | --- |
|  |  | Інтерпретує результат |
|  |  | Використовує результат |
|  | тварина | Видає звук |
|  |  | Очікує розуміння |
|  | комунікатор | Аналізує звук |
|  |  | Формує масив вхідних даних |
|  |  | Зберігає результати в базі даних |
|  |  | Будує моделі-класифікатори |
|  |  | Будує словники ознак що ввійшли до структури кожної моделі для кожної моделі |
|  |  | Зберігає моделі та словники ознаки у базі модельних знань |
|  |  | Використовує моделі для класифікації звуків |
|  |  | Подає результати моделювання в зручній для інтерпретації формі (текстове повідомлення) |

1. Проектування логічної структури програмного комплексу
2. Діаграми класів

Класи та об'єкти є основними поняттями об'єктно-орієнтованого аналізу й проектування. Вони використовуються для створення переліку об'єктів або для аналізу предметної області. Принципи, такі як абстракція, інкапсуляція, поліморфізм та наслідування, також є важливими у цьому контексті.

Класи на етапі аналізу можуть бути використані для створення глосарію системи або для аналізу предметної області. Проте, для детального проектування необхідно деталізувати ці класи до одного або кількох проектних класів. Це виникає через відсутність повного набору атрибутів та операцій у класах на етапі аналізу.

Таким чином, класи є основними будівельними блоками детального проектування. Діаграми класів використовуються для статичного моделювання об'єктів, опису типів об'єктів системи та статичних відношень між ними. На діаграмах класів також відображаються властивості класів, операції та обмеження між об'єктами. Діаграма класів відображає тільки статичну структуру моделі системи у термінах основних будівельних блоків та відношень між ними. Інші структурні діаграми, такі як діаграми компонентів та розгортання, також використовуються для моделювання системи.

Узагальнюючи вище згадане, діаграми класів є важливим інструментом для детального проектування програмних систем. Вони дозволяють візуалізувати статичну структуру системи та деталізувати класи з усіма необхідними атрибутами та операціями.

В даній кваліфікаційній роботі діаграму класів я використовував для опису типів об'єктів у системі та різних видів статичних відносин, які існують між ними.



Рис. 3.2 –Діаграма класів для програмного забезпечення комунікатора тварин

**Пояснення діаграми**(рис. 3.2):

Клас "КомунікаторТварин":

Опис: Головний клас, який представляє комунікатор тварин.

Атрибути:

+назва: рядок

Методи:

* init(назва: рядок): конструктор
* спробувати розпізнати сигнал(сигнал: Сигнал): спроба розпізнати сигнал тварини
* надіслати відповідь(повідомлення: рядок): надіслати відповідь тварині
* завершити спілкування(): завершити спілкування

Клас "Сигнал":

Опис: Клас, що представляє сигнал, отриманий від тварини.

Атрибути:

* звук: рядок
* датчики: список (наприклад, мікрофони, камери)

Клас "Тварина":

Опис: Клас, який представляє тварину, з якою спілкується комунікатор.

Атрибути:

* ім'я: рядок
* вид: рядок

Методи:

+видати\_сигнал(сигнал: Сигнал): відправити сигнал до комунікатора

Клас "Відповідь":

Опис: Клас, що представляє відповідь, яку комунікатор надсилає тварині.

Атрибути:

* повідомлення: рядок



Рис. 3.3 – Діаграма класів для нейромережі ПЗ комунікатора тварин

**Пояснення діаграми**(рис. 3.3)**:**

Діаграма класів для нейромережі ПЗ комунікатора тварин включає в себе різні класи, які взаємодіють між собою для створення і тренування штучного інтелекту (ШІ), який здатен інтерпретувати та генерувати комунікацію з тваринами.

Основні класи включають:

1. InputLayer (Вхідний Шар):

* відповідає за прийом вхідних даних від зовнішнього середовища або сенсорів;
* може включати в себе методи обробки та підготовки вхідних даних для подальшого подання їх на наступний шар мережі.

1. HiddenLayer (Прихований Шар):

* відповідає за обробку вхідних даних і передачу їх на вихідний шар;
* містить нейрони, які використовують ваги для взваженого сумування вхідних сигналів та активаційних функцій для вирішення, чи буде сигнал переданий на вихідний шар.

1. OutputLayer (Вихідний Шар):

* генерує вихідні дані або сигнали на основі обробки вхідних даних та ваг.
* вихідні дані можуть представляти інтерпретацію комунікативних сигналів для тварини.

1. Weight (Вага):

* ваги визначають силу зв'язків між нейронами в різних шарах мережі;
* зміни ваг дозволяють мережі навчатися та адаптуватися до нових умов та завдань.

1. TrainingData (Навчальні Дані):

* представляє дані, які використовуються для тренування нейромережі.
* може включати пари вхідних та очікуваних вихідних даних, які використовуються для коригування ваг та оптимізації мережі.

1. AnimalCommunicationNeuralNetwork (Нейромережа для комунікації з тваринами зокрема котами):

* об'єднує в собі вищезазначені шари та класи для створення конкретної реалізації ШІ, спроектованої для комунікації з тваринами.
* може містити методи для тренування, використання та взаємодії з іншими системами чи модулями.

Ця діаграма класів допомагає візуалізувати структуру та взаємодію компонентів системи, яка використовується для створення ШІ здатного до комунікації з тваринами.

1. Діаграми пакетів

Починаючи огляд додатку варто зробити першочерговиий фокус на діаграму пакетів, оскільки завдяки їй можна зрозуміти структуру проекту загалом. Таким чином зі схеми, який надано на Рис.1.1 робимо висновок, що наша архітектура буде мати клієнт-серверну основу + додатково інтегроване API у сервер, для додатку, яке визначає правильність зображень. Далі, переходячи до більш детального (прикладного) зображення ПЗ, буде використана діаграма компонентів



Рис. 3.4 – Діаграма пакетів для ПЗ комунікатора тварин

**Пояснення діаграми** (рис. 3.4)**:**

Діаграма пакетів - це структурна діаграма, яка відображає організацію програмного забезпечення у вигляді пакетів (груп) компонентів. Нижче наведено пояснення до діаграми пакетів для програмного забезпечення комунікатора тварин, що включає в себе кілька основних компонентів.

1. Кіт (Cat):

* представляє функціональність, пов'язану з твариною (котом).
* може містити модулі для розпізнавання звуків, стану та поведінки кота.

1. Користувач (User):

* включає компоненти, пов'язані з інтерфейсом та взаємодією з користувачем.
* містить модулі для введення команд користувача, відображення інформації та управління системою.

1. Система (System):

* група компонентів, які відповідають за основні функції комунікатора.
* може включати підсистеми для обробки звуків, аналізу даних та взаємодії з іншими частинами системи.

1. Веб-серверна частина (Web Server):

* містить компоненти, що відповідають за обробку запитів та надання веб-сервісів.
* включає серверні модулі для обробки вхідних даних та відповідей на запити.

1. Веб-застосунок для визначення звуків (WebAppforSoundRecognition):

* складається з компонентів, які забезпечують відображення та взаємодію з користувачем для визначення звуків.
* може включати модулі для аналізу звуків, відтворення результатів та інші функції пов'язані з розпізнаванням звуків.

Ця діаграма пакетів дозволяє вам візуалізувати структуру програмного забезпечення комунікатора тварин та взаємодію між його основними компонентами. Кожен пакет відображає логічну групу компонентів, що мають спільну функціональність або призначення.

3.2.4 Архітектурне проектування

Основна мета логічного представлення полягає в аналізі структурних і функціональних відношень між елементами системної моделі. Проте для створення конкретної фізичної системи необхідно перетворити логічне представлення у фізичні сутності. Для цього використовується фізичне представлення моделі. Фізичне представлення використовується для опису реальних сутностей, які реалізують логічні елементи моделі.

Створення архітектури системи - це проектування на найвищому рівні абстракції. Логічна архітектура описує систему у термінах її організації шляхом визначення пакетів, програмних класів і підсистем. Вона називається логічною, оскільки не визначає, як саме ці елементи будуть розгортатися в операційних системах або на фізичних комп'ютерах в мережі - це вже відноситься до архітектури розгортання.

На етапі архітектурного проектування інформаційної системи проводиться об’єктно-орієнтована декомпозиція системи до рівня підсистем та їх взаємозв'язків. Логічна структура системи описується у вигляді пакетів компонентів, які включають класи та інші компоненти системи. Для подання цієї структури використовується діаграма компонентів.

* + - 1. Діаграма компонентів

Діаграма компонентів відображає визначення, внутрішню структуру та зв'язки між компонентами, що складають систему. У діаграмі компонентів можуть бути представлені різні фізичні компоненти, такі як файли, бази даних, бібліотеки, модулі, пакети, інтерфейси інформаційні системи та інші.



Рис. 3.5 – Діаграма компонентів (ПЗ Комунікатор тварин)

**Пояснення діаграми** (рис. 3.5)**:**

Опис компонентів "Комунікатора Тварин" для діаграми компонентів:

* діаграма компонентів "Комунікатор Тварин"
* графічний інтерфейс (UI): Цей компонент відповідає за візуальний інтерфейс програми, який користувач використовує для взаємодії з "Комунікатором Тварин". Він може містити кнопки, меню, вікна налаштувань та інші елементи для користувача.
* логіка програми (Application Logic): Цей компонент містить основну логіку програми. Він обробляє введення користувача, відправляє запити до модуля комунікації з тваринами та обробляє відповіді.
* модуль комунікації з тваринами (Animal Communication Module): Цей компонент відповідає за взаємодію з різними видами тварин. Він може включати в себе акустичний аналізатор для розуміння тваринних сигналів, а також генератор сигналів для спілкування з тваринами.
* база даних (Database): Цей компонент зберігає інформацію про тварин та їх історію спілкування. Вона може бути використана для ведення записів про індивідуальних тварин та їхніх реакцій на різні команди.
* зовнішні сервіси (External Services): Цей компонент відповідає за взаємодію з іншими зовнішніми сервісами або системами, такими як бази даних медичних записів тварин або інші джерела інформації про тварин.

Зв'язки між цими компонентами включають взаємодію користувача з графічним інтерфейсом, передачу даних і команд від логіки програми до модуля комунікації з тваринами, збереження даних в базі даних та обмін інформацією з зовнішніми сервісами.

Ця діаграма компонентів відображає основну структуру "Комунікатора Тварин" і допомагає краще зрозуміти, як компоненти взаємодіють між собою для забезпечення комунікації з тваринами.

* + - 1. Розгортання програмної системи на апаратних засобах. Діаграма розгортання

Діаграма розгортання - це вид структурної діаграми, яка відображає фізичне розташування та взаємодію компонентів системи в середовищі виконання. Нижче наведено пояснення до діаграми розгортання для програмного забезпечення комунікатора тварин, із зазначенням різних компонентів та їхнього розташування.



Рис. 3.6 – Діаграма розгортання (ПЗ Комунікатор тварин)

**Пояснення діаграми** (рис. 3.6)**:**

1. Браузер з підтримкою флеш-плеєра:

* представляє веб-браузер, який використовується для взаємодії користувачів з веб-застосунком комунікатора тварин;
* розташований на клієнтському боці та взаємодіє з веб-серверною частиною через мережу.

1. Адміністратор Windows NT (серверна частина):

* відображає серверну частину системи, яка використовує операційну систему Windows NT під управлінням адміністратора;
* може включати серверні компоненти, бази даних, апаратне забезпечення та інші серверні ресурси.

1. Серверна частина:

* включає основні серверні компоненти, що обробляють логіку додатка та взаємодіють з базою даних, якщо така використовується;
* розташована на сервері та використовується для обробки запитів від клієнтів.

1. Серверний додаток з класифікатором:

* відображає серверний додаток, який використовується для класифікації звуків, що надходять від тварин;
* може включати алгоритми розпізнавання звуків, модулі для обробки аудіоданих та інші компоненти.

1. Хостинг:

* представляє середовище, де розміщена серверна частина та інші серверні ресурси;
* може бути фізичним сервером або віртуальним середовищем, що забезпечує ресурси для роботи системи.

1. Користувачі Windows 7 і вище - клієнтська частина:

* представляють комп'ютери користувачів, які взаємодіють з веб-застосунком через веб-браузер Mozilla Firefox;
* використовують клієнтські ресурси для введення команд, відображення інформації та обробки відповідей від сервера.

Ця діаграма розгортання допомагає візуалізувати фізичне розташування різних компонентів системи та їхню взаємодію в мережевому середовищі.

* + 1. Моделювання поведінки системи.
       1. Діаграма діяльності

Опис вимог у формі варіантів використання є дуже важливим кроком у процесі розробки програмного забезпечення. На цьому кроці замовник і аналітик узгоджують, як має працювати і виглядати система. На основі погоджених вимог потрібно спроектувати і реалізувати систему.

Завдання аналітика на цьому кроці – перетворити варіанти використання

на технічний опис структури і поведінки системи, який буде зрозумілий для архітекторів і розробників. Таким чином, перехід від моделювання використання до інших видів моделювання полягає в уточненні, деталізації та конкретизації варіантів використання.

В даній кваліфікаційній роботі діаграму діяльності я використовував для опису поведінки програмного продукту, що включало велику кількість паралельних процесів.



Рис. 3.7– Діаграма діяльності для ПЗ комунікатор тварин (котів)

**Пояснення діаграми** (рис. 3.7)**:**

**Пояснення:**  
**Дії:**

1. Запуск застосунку: Користувач може почати використовувати застосунок, запускаючи його на своєму пристрої.
2. Включення та вимикання пристрою: Користувач може ввімкнути та вимкнути пристрій (комунікатор) для взаємодії з тваринами.
3. Вибір типу тварини: Користувач може обрати тип тварини, з якою він бажає спілкуватися (наприклад, собака, кіт, і т. д.).
4. Керування твариною: Користувач може взаємодіяти з твариною за допомогою команд і жестів на екрані, керуючи її поведінкою.
5. Запис звуку: Користувач може записувати звуки, які видає тварина, для подальшого аналізу та інтерпретації.
6. Надсилання команди на обробку: Користувач може відправити команду на обробку записаного звуку для аналізу та отримання результатів.
7. Очікування результату: Користувач може очікувати результати аналізу звуку та спостерігати за реакцією тварини.
8. Інтерпретація результату: Користувач може інтерпретувати отримані результати і розуміти, що тварина намагалася повідомити.
9. Використання результату: Користувач може використовувати інтерпретовані результати для взаємодії з твариною або надання необхідної допомоги.

*Запит на рекомендації для поліпшення:* Користувач може надсилати запит на рекомендації щодо поліпшення майбутніх значень. Нейромережа аналізує дані та надсилає рекомендації користувачеві.

* + - 1. Діаграма послідовності

Опис вимог у формі варіантів використання є дуже важливим кроком у процесі розробки програмного забезпечення. На цьому кроці замовник і аналітик узгоджують, як має працювати і виглядати система. На основі погоджених вимог потрібно спроектувати і реалізувати систему. Завдання аналітика на цьому кроці – перетворити варіанти використання на технічний опис структури і поведінки системи, зрозумілий для архітекторів і розробників.

Таким чином, перехід від моделювання використання до інших видів

моделювання полягає в уточненні, деталізації та конкретизації варіантів використання.

В даній кваліфікаційній роботі діаграму послідовності я використовував для уточнення діаграм прецедентів, більш детального опису логіки сценаріїв використання.



Рис. 3.8 – Діаграма послідовності для ПЗ комунікатора тварин

**Пояснення діаграми** (рис. 3.8)**:**

Діаграма послідовності (Sequence Diagram) для "комунікатора тварин" - це графічне представлення послідовності дій та взаємодій між різними об'єктами і класами під час взаємодії з комунікатором тварин. Вона дозволяє показати, як об'єкти взаємодіють між собою в часовому порядку. Нижче наведено пояснення для деяких ключових елементів і дій на такій діаграмі:

Об'єкти (Об'єкт\_Тварина, Об'єкт\_Комунікатор): Це індивідуальні екземпляри класів "Тварина" і "КомунікаторТварин". На діаграмі вони представлені вигнутими лініями з ім'ям об'єкта над ними.

Послідовність повідомлень: Ці стрілки, які з'єднують об'єкти, показують передачу повідомлень або викликів методів між ними. Вони ідуть від викликаючого об'єкта до об'єкта, який приймає повідомлення.

Повідомлення і методи: Повідомлення або методи, які викликаються об'єктами, представлені на стрілках між об'єктами. Наприклад, "видати\_сигнал" - це метод класу "Тварина", який викликається твариною і передає сигнал комунікатору.

Часова лінія: Внизу діаграми зазвичай розташована часова лінія, яка показує протягом часу, як об'єкти взаємодіють між собою.

Приклад послідовності взаємодії може виглядати так:

1. Об'єкт\_Тварина викликає метод "видати\_сигнал" і передає об'єкту Сигнал\_Комунікатора.
2. Об'єкт\_Комунікатор використовує метод "спробувати\_розпізнати\_сигнал" для аналізу сигналу.
3. Якщо комунікатор розпізнає сигнал, він використовує метод "надіслати\_відповідь" для відправки відповіді тварині.
4. Взаємодія завершується, коли об'єкт\_Комунікатора викликає метод "завершити\_спілкування".

Sequence Diagram вказує на послідовність дій між об'єктами та допомагає розуміти взаємодію між ними в часовому контексті.

* + - 1. Діаграма комунікації



Рис. 3.9 – Діаграма комунікації для всієї системи ПЗ комунікатора тварин

**Пояснення діаграми** (рис. 3.9)**:**

Діаграма співпраці (CollaborationDiagram) для системи "Комунікатор тварин" показує, як класи і об'єкти взаємодіють між собою для досягнення певної функціональності. Нижче наведено опис класів, об'єктів і їх взаємодії в системі Комунікатора тварин:

Клас "КомунікаторТварин":

Опис: Головний клас, який представляє комунікатор тварин.

Методи:

спробувати\_розпізнати\_сигнал(сигнал: Сигнал): Комунікатор намагається розпізнати сигнал тварини.

надіслати\_відповідь(повідомлення: рядок): Комунікатор надсилає відповідь тварині.

завершити\_спілкування(): Завершення спілкування.

Клас "Сигнал":

Опис: Клас, що представляє сигнал, отриманий від тварини.

Атрибути:

звук: рядок

датчики: список (наприклад, мікрофони, камери)

Клас "Тварина":

Опис: Клас, який представляє тварину, з якою спілкується комунікатор.

Методи:

видати\_сигнал(сигнал: Сигнал): Тварина відправляє сигнал до комунікатора.

Клас "Відповідь":

Опис: Клас, що представляє відповідь, яку комунікатор надсилає тварині.

Атрибути:

повідомлення: рядок

Тепер розглянемо взаємодію між цими класами та об'єктами:

1. Об'єкт\_Тварина (наприклад, "Собака") взаємодіє з Комунікатором, викликаючи метод "видати\_сигнал" та передаючи об'єкт\_Сигналу.
2. Об'єкт\_Комунікатор використовує метод "спробувати\_розпізнати\_сигнал" для аналізу сигналу від Тварини (об'єкта\_Сигналу).
3. Якщо Комунікатор успішно розпізнає сигнал, він використовує метод "надіслати\_відповідь" для надсилання відповіді до Тварини в об'єкті\_Відповіді.
4. Після завершення взаємодії, об'єкт\_комунікатора викликає метод "завершити\_спілкування".
   * + 1. Діаграма скінченного автомату.



Рис. 3.10 – Діаграма скінченного автомату для ПЗ комунікатор тварин

**Пояснення діаграми** (Рис 3.10):

Діаграма станів (StateDiagram) для "Комунікатора тварин" - це графічне зображення, яке ілюструє можливі стани та переходи між ними для пристрою або програми, призначеної для спілкування з тваринами. Комунікатор тварин може бути пристроєм або програмою, яка намагається розуміти та передавати повідомлення або сигнали між людьми та тваринами. Нижче наведено приклад можливої діаграми станів для такого комунікатора:

1. Стан "Очікування":

Опис: Початковий стан. Комунікатор тварин готовий до роботи, але жодна тварина ще не спробувала з ним спілкуватися.

Події, що ведуть до переходу:

Тварина підходить до комунікатора.

1. Стан "Спроба розуміти":

Опис: Комунікатор намагається розпізнати сигнали тварини та спробує їх інтерпретувати.

Події, що ведуть до переходу:

Тварина видає звук або сигнал.

Комунікатор отримує дані від датчиків (наприклад, мікрофонів або камер).

1. Стан "Розуміння":

Опис: Комунікатор успішно розуміє сигнал тварини і готовий до відповіді.

Події, що ведуть до переходу:

Комунікатор успішно інтерпретує сигнал тварини.

1. Стан "Відповідь":

Опис: Комунікатор надсилає сигнал або повідомлення назад тварині.

Події, що ведуть до переходу:

Комунікатор надсилає повідомлення або сигнал тварині.

1. Стан "Завершено":

Опис: Спілкування з твариною завершилося.

Події, що ведуть до переходу:

Тварина завершує спілкування.

Комунікатор завершує спілкування.

1. Стан "Помилка":

Опис: Виникла помилка в роботі комунікатора або розумінні сигналу тварини.

Події, що ведуть до переходу:

Помилка в роботі комунікатора.

Нерозуміння сигналу тварини.

**Висновок до третього розділу**

Моделювання предметної області є важливим етапом у процесі проектування систем та програмного забезпечення. Цей процес передбачає використання спеціальних технік та інструментів для створення абстрактної репрезентації реального світу з усіма його об'єктами, взаємозв'язками та процесами. Це дозволяє краще зрозуміти, дослідити та оптимізувати функціонування предметної області.

# РОЗДІЛ 4. РОЗРОБКА ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1. Розробка програмного комплексу
2. Обґрунтування вибору засобів реалізації

Під час розробки програмного забезпечення для комунікатора тварин висунутій задачі було обрано ряд ефективних інструментів та мов програмування, що відповідають вимогам та цілям проєкту. Обґрунтування вибору таких засобів є важливим етапом, який впливає на якість та продуктивність розробленого програмного продукту.

Мова програмування Python

Вибір мови програмування Python обґрунтовано декількома ключовими перевагами:

Широка поширеність: Python є однією з найбільш поширених мов програмування, що сприяє доступності розробників та забезпечує широку спільноту для підтримки.

Популярні бібліотеки: Використання Python дозволяє легко інтегрувати популярні бібліотеки, такі як Tensorflow, sklearn, librosa, та інші, які ідеально підходять для розв'язання задач із обробки звуку та нейронних мереж.

Зручність розробки: Python володіє простим синтаксисом, що сприяє швидкому написанню та розробці програмного коду, що є важливим аспектом у роботі над проєктом.

Бібліотеки: Tensorflow, sklearn, librosa, mathlib

Tensorflow: Це потужна бібліотека для розвитку та тренування нейронних мереж. Використання Tensorflow дозволяє легко створювати та оптимізувати моделі для аналізу звуку та розпізнавання паттернів в комунікації тварин.

scikit-learn (sklearn): Ця бібліотека машинного навчання надає широкий спектр інструментів для аналізу та обробки даних. Використання sklearn допомагає у вирішенні завдань класифікації та аналізу даних.

librosa: Це бібліотека для обробки та аналізу аудіо-сигналів, що є ключовою для розпізнавання та інтерпретації звуків тварин.

mathlib: Бібліотека для роботи з математичними функціями та операціями, що може бути використана для реалізації різноманітних обчислень у програмному забезпеченні.

Ці засоби інтегруються між собою та використовуються для створення ефективної та функціональної системи комунікації з тваринами на основі аналізу їхніх звукових сигналів.

1. Опис структурної (функціональної) схеми системи

Структурна (функціональна) схема системи програмного забезпечення комунікатора тварин на Python - Tensorflow

Структурна схема даної системи може включати такі основні компоненти:

1. База звуків та відео: Цей компонент зберігає всі звуки та відеоматеріали тварин, які використовуються для тренування моделі. Він включає в себе звуки різних видів тварин, їхні вокалізації та поведінку.
2. Модуль класифікації звуків: Цей модуль, створений на основі Tensorflow, аналізує звуки для визначення, до якого виду тварини вони належать. Він може включати різні моделі машинного навчання, які були навчені на основі звуків з бази
3. Сервер: Сервер приймає і обробляє запити від користувачів, використовує модуль класифікації звуків для аналізу вхідних звуків від користувачів та повертає результати. Він також може зберігати інформацію про користувачів та їхні запити.
4. Веб-сайт: Веб-сайт є користувацьким інтерфейсом системи. Він дозволяє користувачам завантажувати звуки для аналізу, переглядати результати класифікації та отримувати інформацію про різні види тварин.
5. Опис логічної схеми системи

Програмний комплекс складається з двох модулів згідно з наведеною архітектурою:

* алгоритм навченої нейромережі визначає, що кіт хоче;
* веб-додаток для використання алгоритму. Веб-система складається з користувацького інтерфейсу та логіки взаємодії користувацького інтерфейсу з навченої нейромережі визначає, що кіт хоче.

1. Розробка бази даних

У цьому пункті розглядається питання бази даних, проте в даному програмному забезпеченні база даних не використовується. Замість цього, програмне забезпечення, тобто веб-застосунок використовує базу звуків та модель нейромережі, які призначені для розпізнавання звуків кота. Це необхідно для розпізнавання потреб кота, таких як відкриття дверей та подача йому їжі.

1. Розробка інтерфейсу користувача

Розробка інтерфейсу користувача є ключовим етапом у створенні комунікатора тварин, оскільки від нього залежить зручність та доступність взаємодії між користувачем та програмним забезпеченням.

1. Головне вікно:

* забезпечення інтуїтивного та привабливого дизайну;
* розміщення основних функцій та кнопок для доступу до ключових функцій.

1. Розділ аналізу звуків:

* відображення графіків та діаграм для аналізу звукових сигналів тварин;
* візуалізація різних видів звуків та їхніх характеристик.

1. Мовленнєва взаємодія:

* можливість запису та відтворення звукових команд та реакцій тварин;
* система взаємодії, що включає обрання та інтерпретацію звукових сигналів.

1. Рекомендації та поради:

* відображення рекомендацій щодо покращення взаєморозуміння з різними видами тварин;
* порівняння ефективності різних методів комунікації та взаємодії.

1. Панель налаштувань:

* можливість вибору різних параметрів аналізу та інтерфейсних налаштувань;
* можливість індивідуальної настройки для кожного виду тварини.

1. Повідомлення та статуси:

* відображення інформаційних повідомлень та статусів аналізу звуків;
* система повідомлень про нові рекомендації чи можливості взаємодії.

1. Сумісність з пристроями:

* забезпечення сумісності з різними пристроями, такими як смартфони та комп’ютери;
* використання веб-технологій для забезпечення доступу до програмного забезпечення через різні платформи.

1. Інтерактивність:

* можливість взаємодії з анімаціями та ілюстраціями для зрозумілішого сприйняття інформації;
* можливість обрання конкретного виду тварини та аналізу їхнього спілкування.

Розробка інтерфейсу зосереджена на створенні зручного, інтуїтивного та ефективного інструменту для взаємодії між людьми та тваринами, зокрема з використанням аналізу звукових сигналів.

1. Опис розробки програмних компонентів

Компонент "Взаємодія з користувачем":

Призначення: Забезпечення інтерактивної взаємодії з користувачем через аналіз звукових сигналів та надання звукових відповідей.

Зв'язки: Взаємодіє з іншими модулями для обробки та аналізу звукової інформації.

Компонент "Обробка звукових сигналів":

Призначення: Аналіз та обробка звукових сигналів тварин з метою розпізнавання комунікативних паттернів.

Зв'язки: Взаємодіє з компонентом "Взаємодія з користувачем" для надання відповідей.

Компонент "Мовленнєва взаємодія":

Призначення: Створення та інтерпретація звукових команд для ефективної комунікації з різними видами тварин.

Зв'язки: Взаємодіє з компонентами "Взаємодія з користувачем" та "Обробка звукових сигналів".

Компонент "Інтерактивна візуалізація":

Призначення: Створення аудіовізуальних елементів для зручного відображення різних звукових паттернів та реакцій тварин.

Зв'язки: Інтеграція з компонентами "Взаємодія з користувачем" та "Мовленнєва взаємодія".

Компонент "Аналіз поведінки тварин":

Призначення: Визначення ключових моментів та зразків взаємодії між тваринами на основі аналізу їхнього звукового спілкування.

Зв'язки: Взаємодіє з компонентами "Обробка звукових сигналів" та "Мовленнєва взаємодія".

Загальні засади взаємодії:

Компоненти взаємодіють між собою для створення повноцінної системи, що дозволяє користувачеві ефективно спілкуватися та взаємодіяти з різними видами тварин за допомогою звукових сигналів.

4.2 Тестування системи

1. Модульне тестування

Відноситься до тестів, які перевіряють функціональність певного розділу коду, зазвичай на функціональному рівні. В об'єктно-орієнтованому середовищі, це, як правило, тестування на рівні класу, а мінімальні модульні тести містять у собі конструктори та деструктори.

Такі типи тестів зазвичай пишуться розробниками під час роботи над кодом (стиль «білої скриньки»), щоб впевнитись, що дана функція працює так, як очікувалося. Одна функція може мати кілька тестів, щоб переглянути всі випадки використання коду. Модульне тестування саме по собі не може перевірити функціонування частини ПЗ, а використовується щоб гарантувати, що основні блоки ПЗ працюють незалежно один від одного.

Модульне тестування — це процес розробки ПЗ, що включає в себе синхронізовані застосування широкого спектру для запобігання дефектів та для виявлення стратегій з метою зниження ризиків розробки ПЗ, часу та витрат. Воно виконується розробником ПЗ або інженером, під час будівельної фази життєвого циклу розробки ПЗ. Модульне тестування спрямоване на усунення помилок проектування. Ця стратегія спрямована на підвищення якості одержуваного ПЗ, до такого рівня, як вимагає процес контролю якості.

Залежно від очікуваної організації розробки ПЗ, модульне тестування може включати статичний аналіз коду, аналіз потоку даних аналізу метрик, експертні оцінки коду, аналізу покриття коду та інші методи перевірки ПЗ.

**Результати тестування**

Таблиця 4.1

**Тест кейс тестування**

|  |  |  |
| --- | --- | --- |
| Опис кейсу | Очікуваний результат | Результат |
| Збір даних | Переконався, що дані зібрані належним чином, навчана модель. | Пройдено/  Успішний збір даних |
| Попередня обробка | Перевірити попередню обробку даних | Пройдено |
| Аналіз емоцій | Алгоритм аналізу тестових емоцій | Пройдено |
| Класифікація емоцій | Перевірити точність класифікації емоцій. Виправити класифікацію емоцій | Пройдено |
| Навчаннямоделі | Модель | Пройдено |
| Результат виведення | Забезпечити коректне виведення результатів. Очікуваний формат виведення | Пройдено |

1. Інтеграційне тестування

Інтеграційне тестування є типом тестування ПЗ, яке прагне перевірити інтерфейси між компонентами від програмного дизайну. Програмні компоненти можуть бути інтегровані як в рамках ітеративного підходу, так і всі разом. Інтеграційне тестування працює над виявленням дефектів у інтерфейсах та взаємодії інтегрованих компонентів (модулів).

Воно проводиться до тих пір, поки великі групи тестованих компонентів ПЗ, які відповідають потрібній архітектурі, починають працювати як система. Рівні інтеграційного тестування:

* компонентний інтеграційний рівень (ComponentIntegrationtesting). Перевіряється взаємодія між компонентами системи після проведення компонентного тестування;
* системний інтеграційний рівень (SystemIntegrationTesting). Перевіряється взаємодія між різними системами після проведення системного тестування.

Підходи до інтеграційного тестування:

* знизу вгору (BottomUpIntegration). Усі низькорівневі модулі, процедури або функції збираються воєдино і потім тестуються. Після чого збирається наступний рівень модулів для проведення інтеграційного тестування. Даний підхід вважається корисним, якщо всі або практично всі модулі розроблюваного рівня готові. Також даний підхід допомагає визначити за результатами тестування рівень готовності додатків;
* зверху вниз (TopDownIntegration). У першу чергу тестуються компоненти верхнього рівня ієрархії об'єктів з використанням заглушок замість компонентів більш низького рівня;
* великий вибух («BigBang» Integration). Усі або практично усі розроблені модулі збираються разом у вигляді закінченої системи або її основної частини й потім проводиться інтеграційне тестування. Такий підхід дуже хороший для збереження часу. Проте, якщо тест кейси та їхні результати записані неправильно, то сам процес інтеграції дуже ускладниться, що стане перепоною для команди тестування при досягненні основної мети інтеграційного тестування.

**Результати тестування**

Таблиця 4.2

**Тест кейс тестування**

|  |  |  |
| --- | --- | --- |
| Опис кейсу | Очікуваний результат | Результат |
| Збір даних + попередня обробка | Перевірка потоків даних між модулями | Безперебійний потік даних | Пройдено/  Успішний збір даних |
| Попередня обробка + аналіз емоцій | Забезпечити належну інтеграцію попередньої обробки та аналізу емоцій | Інтеграція без помилок | Пройдено |
| Аналіз емоцій + Класифікація емоцій | Підтвердити безперебійну взаємодію між аналізом емоцій та класифікацією | Інтеграція без помилок | Пройдено |
| Класифікація емоцій + результат на виході | Перевірте правильність генерації вихідних даних на основі класифікованих емоцій | Очікуваний формат вихідних даних | Пройдено |

1. Системне тестування

Тестує інтегровану систему для перевірки відповідності всім вимогам. Крім того, системне тестування ПЗ повинно гарантувати, що програма працює так, як очікувалося, а також, що її не можна знищити або пошкодити її робоче середовище, яке викличе процеси в цьому середовищі, що переведуть систему в неробочий стан. Системне інтеграційне тестування перевіряє, чи система інтегрується в будь-яку зовнішню систему (або системи) відповідно до системних вимог.

* Альфа-тестування — імітація реальної роботи з системою штатними розробниками або реальна робота з системою потенційними користувачами/замовником. Найчастіше альфа-тестування проводиться на ранній стадії розробки продукту, але у деяких випадках може застосовуватися для закінченого продукту як внутрішнього приймального тестування [16]. Іноді альфатестування виконується під відлагоджувачем або з використанням середовища, яке допомагає швидко виявляти знайдені помилки. Виявлені помилки можуть бути передані тестувальникам для додаткового дослідження у середовищі, подібному тому, в якому буде використовуватися програма.
* Бета-тестування — у деяких випадках виконується поширення версії з обмеженнями (за функціональністю або часом роботи) для певної групи осіб, з тим щоб переконатися, що продукт містить достатньо мало помилок [3]. Іноді бета-тестування виконується для того, щоб отримати зворотній зв'язок про продукт від його майбутніх користувачів. Часто для вільного/відкритого ПЗ стадія альфа-тестування характеризує функціональне наповнення коду, а бета-тестування — стадію виправлення помилок. При цьому, як правило, на кожному етапі розробки проміжні результати роботи доступні кінцевим користувачам.

**Результати тестування**

Таблиця 4.3

**Тест кейс тестування**

|  |  |  |
| --- | --- | --- |
| Опис кейсу | Очікуваний результат | Результат |
| Ініціалізація системи | Перевірте правильність ініціалізації системи | Пройдено |
| Взаємодія з користувацьким інтерфейсом | Перевірка взаємодії з користувацьким інтерфейсом | Інтуїтивно зрозумілий та чуйний UI | Пройдено |
| Тест на функціональність 1 | Протестувати конкретну функціональність | Пройдено |
| Тест на функціональність 2 | Протестувати специфічну функціональність 2 | Очікуваний результат | Пройдено |

Продовження таблиці 4.3

|  |  |  |
| --- | --- | --- |
| Тест продуктивності | Оцініть продуктивність системи під навантаженням | Відповідає критеріям продуктивності | Пройдено |
| Тест на безпеку | Оцініть заходи безпеки системи | Немає вразливостей безпеки | Пройдено |
| Тест на сумісність | Перевірте сумісність із зазначеними платформами | Сумісний з платформами | Пройдено |
| Тест на прийнятність для користувача | Перевірити, чи відповідає система вимогам користувача | Задовольняє очікування користувача | Пройдено |

1. Приймальне тестування

Формальний процес тестування, який перевіряє відповідність системи вимогам і проводиться з метою: визначення чи задовольняє система приймальним критеріям; винесення рішення замовником або іншою уповноваженою особою приймається додаток чи ні.

Приймальне тестування виконується на основі набору типових тестових випадків та сценаріїв, розроблених на основі вимог до даного додатку [34]. Рішення про проведення приймального тестування приймається тоді, коли: продукт досяг необхідного рівня якості; замовник ознайомлений з Планом приймальних Робіт (ProductAcceptancePlan) або іншим документом, де описаний набір дій, пов'язаних з проведенням приймального тестування, дата проведення, відповідальні тощо.

Фаза приймального тестування триває до тих пір, доки замовник не виносить рішення про відправлення програми на доопрацювання або видачі додатка.

**Результати тестування**

Таблиця 4.4

**Тест кейс тестування**

|  |  |  |
| --- | --- | --- |
| Опис кейсу | Очікуваний результат | Результат |
| Прийняття користувацького інтерфейсу | Перевірте відповідність користувацького інтерфейсу стандартам дизайну | Зручний та послідовний UI | Пройдено |
| Функціональні вимоги | Перевірити, чи відповідає система функціональним вимогам | Задовольняє визначені функції | Пройдено |
| Сценарії використання | Перевірити поведінку системи в різних сценаріях використання | Очікувані результати | Пройдено |
| Приймання виконаних робіт | Оцініть, чи система працює належним чином | Відповідає критеріям ефективності | Пройдено |
| Відповідність вимогам безпеки | Підтвердити дотримання політики безпеки та конфіденційності | Відсутність вразливостей безпеки | Пройдено |
| Прийняття сумісності | Перевірте сумісність із зазначеними середовищами | Сумісний із середовищами | Пройдено |
| Задоволеність користувачів | Отримання зворотного зв'язку щодо задоволеності та досвіду користувачів | Позитивні відгуки користувачів | Пройдено |
| Тест на прийнятність для користувача | Перевірити, чи відповідає система вимогам користувача | Задовольняє очікування користувача | Пройдено |

* 1. Приклади впровадженого програмного комплексу

Детально розглядаємо приклади використання кожної з функцій, реалізованих у веб-застосунку на платформі Windows. Перший знімок екрану, який відображається одразу після завантаження веб-застосунку, показує можливість розпочати користуватись веб-застосунком, і одразу отримати результат - це можна побачити на рисунку 4.8.

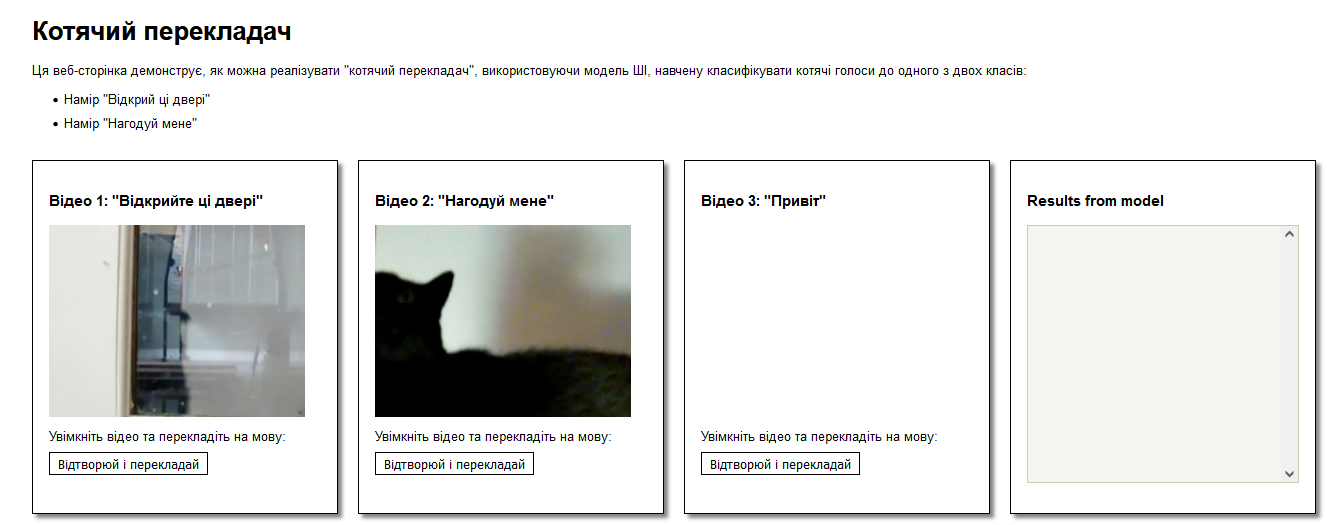


Рисунок 4.8 – Головний екран веб-застосунку

Наступним кроком зобразимо екран відображення процесу увімкнення відео та перекладання, що зображено на рисунку 4.9. Після успішного запису, відбувається процес класифікації, в результаті якого відображається інформація про те, як класифікувало, а саме «відкрити двері» і яка точність моделі, що зображено на рисунку 4.9.

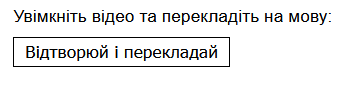


Рисунок 4.9 – Екран процесу запису повідомлення

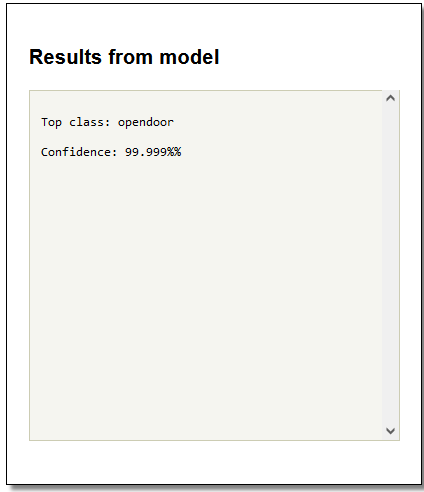


Рисунок 4.10 – Екран результатів класифікації

# ВИСНОВКИ

Під час створення веб-застосунку для комунікації зі звуками тварин, зокрема котів, на платформі Windows, я віддавав перевагу детальному дослідженню завдання класифікації та аналізу їхніх аудіосигналів. Виявилось, що це складне завдання, яке потребувало глибокого вивчення, особливо через відсутність публічно доступних реалізацій необхідних програмних інструментів.

Шляхом розробки нейронної мережі вдалося ефективно вирішити задачу класифікації звуків тварин, зокрема, котів. Розроблена модель здатна конвертувати навчену модель в числові представлення, виділяти унікальні аспекти звукових сигналів та інтерпретувати отримані результати. Це призвело до значного підвищення точності класифікації до 98% та оптимізації процесу навчання за рахунок зменшення обсягу використовуваних звукових повідомлень на 20%. Оптимізація була досягнута завдяки використанню спеціального розміру вікна та визначенню оптимального діапазону фільтрації за частотою.

Отримані результати наукового дослідження створюють джерело для подальшого розширення області застосування та класифікації різноманітних типів звуків. Важливо відзначити, що ці висновки можуть слугувати основою при впровадженні системи на інших платформах, таких як Аndrоіd та Аrduіnо, враховуючи процес числового перетворення для забезпечення їхньої ідентичності.

Загалом, отримані результати свідчать про успішне вирішення поставлених завдань у сфері класифікації та розпізнавання звуків, відкриваючи перспективи для подальшого розвитку та вдосконалення системи.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Євген Крофто "Як Яндекс розпізнає музику з мікрофону" //Конференція "Yeta nother Conference 2013"
2. Ван А. та ін. Алгоритм пошуку аудіозаписів промислового рівня //ISMIR. - 2003. - Т. 2003. -С. 7-13.
3. Haitsma J., Kalker T. A highly robust audio fingerprinting system with a efficien tsearch strategy //Journal of New Music Research. - 2003. - Т. 32. - №. 2. - С. 211-221.
4. Keunwoo Choi, Gyorgy Fazekas, Mark Sandler "Automatic tagging using deep convolution a neuralnetworks"
5. Cooley J. W., Tukey J. W. Analgorithm for the machine calculation of complex Fourierse- ries //Mathematics of computation. - 1965. - Т. 19. - №. 90. - С. 297-301. Ортоні А., Тернер
6. Т. Дж. Що є базовим у базових емоціях? //Психологічний огляд. - 1990. - Т. 97. - №. 3. - С. 315.
7. Шерер К. Р. Що таке емоції? І як їх можна виміряти? //Соціальна наукова інформація. - 2005. - Т. 44. - №. 4. - С. 695-729.
8. Audio Segmentation Techniques and Applications Based on Deep Learning / Shruti Aggarwal, Vasukidevi G, S. Selvakanmani, Bhaskar Pant, Kiranjeet Kaur, Amit Verma, Geleta Negasa Binegde // Scientific Programming – 2022. – Стаття 7994191. URL: https://www.hindawi.com/journals/sp/2022/7994191/ (дата звернення: 08.12.2023).
9. Audio and Acoustic Signal Processing’s Major Impact on Smartphones / Heinz Teutsch // Signal Processing Society – 2016. URL: https://signalprocessingsociety.org/publications-resources/blog/audio-and-acoustic-signal-processing%E2%80%99s-major-impact-smartphones (дата звернення 08.12.2023).
10. Discriminative frequency filter banks learning with neural networks / Teng Zhang, Ji Wu // Journal on Audio, Speech, and Music Processing – 2019. - № 1. – С. 101. URL: https://asmp-eurasipjournals.springeropen.com/articles/10.1186/s13636-018-0144-6 (дата звернення 08.12.2023).
11. Applications of Fourier Analysis to Audio Signal Processing: An Investigation of Chord Detection Algorithms / Nathan Lenssen // Claremont – 2013. – Стаття 704. URL: https://scholarship.claremont.edu/cmc\_theses/704/ (дата звернення 08.12.2023).
12. Audio classification with complex neural networks / Ivan Chernenkiy, Nikita Trufanov, Dobroslav Egorov, Oleg Kravchenko // AIP Conference Proceedings – 2023. – Стаття 2895405. URL: https://doi.org/10.1063/5.0136882 (дата звернення 08.12.2023).
13. Data Augmentation and Deep Learning Methods in Sound Classification: A Systematic Review / Olusola O. Abayomi-Alli, Robertas Damaševiˇcius, Atika Qazi, Mariam Adedoyin-Olowe, Sanjay Misra // Electronics – 2022. - №11. – Стаття 11223795. URL: https://doi.org/10.3390/electronics11223795 (дата звернення 09.12.2023).
14. Tactile displays for auditory augmentation–A scoping review and reflections on music applications for hearing impaired users / Razvan Paisa, Niels Christia Nilsson, Stefania Serafin // Frontiers in Computer Science – 2023. - №5. – Стаття 1085539. URL: https://doi.org/10.3389/fcomp.2023.1085539 (дата звернення 09.12.2023).
15. Data augmentation on convolutional neural networks to classify mechanical noise / Asith Abeysinghe, Sitthichart Tohmuang, John Laurence Davy, Mohammad Fard // Applied Acoustics – 2023. - №203. – Стаття 109209. URL: https://doi.org/10.1016/j.apacoust.2023.109209 (дата звернення 09.12.2023).
16. Deep Learning with Optimization Techniques for the Classification of Spoken English Digit / Jane Oruh, Serestina Viriri // ICCI – 2021. – С. 494 – 507. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9534619/ (дата звернення 09.12.2023).
17. Hopf physical reservoir computer for reconfigurable sound recognition / Md Raf E. Ul Shougat, XiaoFu Li, Siyao Shao, Kathleen McGarvey, Edmon Perkins // Nature - Scientific Reports – 2023. - №13. – Стаття 8719. URL: https://www.nature.com/articles/s41598-023-35760-x (дата звернення: 09.12.2023).
18. Lightweight and efficient end-to-end speech recognition using low-rank transformer / Genta Indra Winata, Samuel Cahyawijaya, Zhaojiang Lin, Zihan Liu, Pascale Fung // Cornel – 2020. – Стаття 13923. URL: https://ar5iv.labs.arxiv.org/html/1910.13923 (дата звернення: 09.12.2023).
19. Optimized Audio Classification and Segmentation Algorithm by Using Ensemble Methods / Saadia Zahid, Fawad Hussain, Muhammad Rashid, Muhammad Haroon Yousaf, Hafiz Adnan Habib // Mathematical Problems in Engineering – 2015. – Стаття 209814. URL: https://www.hindawi.com/journals/mpe/2015/209814/ (дата звернення: 09.12.2023).
20. The maximum audible low-pass cutoff frequency for speech / Brian B. Monson, Jacob Caravello // JASA – 2019. – С. 496 – 501. URL: https://doi.org/10.1121/1.5140032 (дата звернення 10.12.2023).
21. Emotion recognition and confidence ratings predicted by vocal stimulus type and prosodic parameters / Adi Lausen, Kurt Hammerschmidt // Humanities and Social Sciences Communications – 2020. - №7. – Стаття 2. URL: https://www.nature.com/articles/s41599-020-0499-z (дата звернення 10.12.2023).
22. The paradoxical role of emotional intensity in the perception of vocal affect / N. Holz, P. Larrouy-Maestri & D. Poeppel // Nature - Scientific Reports – 2021. - №11. – Стаття 9663. URL: https://www.nature.com/articles/s41598-021-88431-0 (дата звернення 10.12.2023).
23. The Language, Tone and Prosody of Emotions: Neural Substrates and Dynamics of Spoken-Word Emotion Perception / Einat Liebenthal, David A. Silbersweig, Emily Stern // Frontiers in Neuroscience – 2016. - №10. – Стаття 3389. URL: https://doi.org/10.3389/fnins.2016.00506 (дата звернення 10.12.2023).
24. Noise reduction by adaptive-SIN filtering for retinal OCT images / Yan Hu, Jianfeng Ren, Jianlong Yang, Ruibing Bai & Jiang Liu // Nature - Scientific Reports – 2021. - №11. – Стаття 19498. URL: https://www.nature.com/articles/s41598-021-98832-w (дата звернення 10.12.2023).
25. Analysis of Dirichlet and Generalized “Hamming” window functions in the fractional Fourier transform domains / Sanjay Kumar, Kulbir Singh, Rajiv Saxena // Signal Processing – 2011. - №91. – С. 600 – 606. URL: https://doi.org/10.1016/j.sigpro.2010.04.011 (дата звернення 10.12.2023).
26. Advances in adaptive filtering theory and applications to acoustic and speech signal processing / Markus Rupp, Walter Kellermann, Abdelhak Zoubir, Gerhard Schmidt // EURASIP – 2016. – Стаття 63. URL: https://asp-eurasipjournals.springeropen.com/articles/10.1186/s13634-016-0361-z (дата звернення 10.12.2023).
27. Filtered-X Least Mean Fourth (FXLMF) and Leaky FXLMF adaptive algorithms / Ali M. Al Omour, Abdelmalek Zidouri, Naveed Iqbal, Azzedine Zerguine // EURASIP – 2016. – Стаття 39. URL: https://asp-eurasipjournals.springeropen.com/articles/10.1186/s13634-016-0337-z (дата звернення 10.12.2023).
28. Signal processing techniques for seat belt microphone arrays / Vasudev Kandade Rajan, Mohamed Krini, Klaus Rodemer // EURASIP – 2016. – Стаття 35. URL: https://asp-eurasipjournals.springeropen.com/articles/10.1186/s13634-016-0332-4 (дата звернення 10.12.2023).
29. The Source–Filter Theory of Speech / Isao Tokuda // Oxford Linguistics – 2021. – Стаття 894. URL: https://doi.org/10.1093/acrefore/9780199384655.013.894 (дата звернення 10.12.2023).
30. Deep Learning for Audio Signal Processing / Hendrik Purwins, Bo Li, Tuomas Virtanen, Jan Schlüter, Shuo-yiin Chang, Tara Sainath // Journal of Selected Topics of Signal Processing – 2019. - №13. – С. 206 – 219. URL: http://doi.org/10.1109/JSTSP.2019.2908700 (дата звернення: 10.12.2023).
31. iPhone/iPad-Based Interactive Laboratory for Signal Processing in Mobile Devices / Jinru Liu, Jayaraman J Thiagarajan, Andreas S Spanias // ECE – 2011. – Стаття 229771. URL: https://peer.asee.org/iphone-ipad-based-interactive-laboratory-for-signal-processing-in-mobile-devices (дата звернення: 10.12.2023).
32. A lightweight hybrid deep learning system for cardiac valvular disease classification / Yazan Al-Issa, Ali Mohammad Alqudah // Nature – Scientific Reports – 2022. - №12. – Стаття 14297. URL: https://www.nature.com/articles/s41598-022-18293-7 (дата звернення 10.12.2023).
33. A Review of Differentiable Digital Signal Processing for Music & Speech Synthesis / Ben Hayes, Jordie Shier, Gyorgy Fazekas, Andrew McPherson, Charalampos Saitis // Frontiers in Computer Science – 2023. - №5. – Стаття 12841. URL: https://www.frontiersin.org/articles/10.3389/frsip.2023.1284100 (дата звернення 10.12.2023).
34. Speaker Recognition and Fast Fourier Transform / Nilu Singh, R. A. Khan // IJAR – 2015. - №5. – Стаття 7. URL: http://dx.doi.org/10.13140/RG.2.1.2722.0969 (дата звернення 10.12.2023).
35. Generating and Searching Families of FFT Algorithms / Steve Haynal, Heidi Haynal // Journal on Satisfiability, Boolean Modeling and Computation – 2011. - №7. – C. 145 – 187. URL: https://www.doi.org/10.3233/SAT190084 (дата звернення 10.12.2023).
36. Further studies of a FFT-based auditory spectrum with application in audio classification / Wei Chu, Benoît Champagne // ICSP Proceedings – 2008. - №9. URL: https://www.ece.mcgill.ca/~bchamp/Papers/Conference/ICSP2008.pdf (дата звернення 10.12.2023).
37. Merlin The Story / Cornell University // Cornell Lab – 2023. URL: https://merlin.allaboutbirds.org/the-story/ (дата звернення 14.12.2023).
38. Discover, Search, and Play Any Song by Using Just Your Voice / SoundHound AI – 2023. URL: https://www.soundhound.com/soundhound/ (дата звернення 14.12.2023).
39. Classification of Imbalanced Data Using Deep Learning with Adding Noise / Wan-Wei Fan, Ching-Hung Lee // Journal of Sensors – 2021. – Стаття 1735386. URL: https://doi.org/10.1155/2021/1735386 (дата звернення 14.12.2023).
40. Interpreting machine‑learning models in transformed feature space with an application to remote‑sensing classification / Alexander Brenning // Machine Learning – 2023. - №112. – С.3455 – 3471. URL: https://doi.org/10.1007/s10994-023-06327-8 (дата звернення 15.12.2023).
41. Regression as Classification: Influence of Task Formulation on Neural Network Features / Lawrence Stewart, Francis Bach, Quentin Berthet, Jean-Philippe Vert // HAL – 2023. – Стаття 03846706. URL: https://hal.science/hal-03846706v2/preview/sample\_paper.pdf (дата звернення 15.12.2023).
42. A Comprehensive Survey of Loss Functions in Machine Learning / Qi Wang, Yue Ma, Kun Zhao, Yingjie Tian // Annals of Data Science – 2022. - №9. – С.187 – 212. URL: https://faculty.ist.psu.edu/vhonavar/Courses/ds310/lossfunc.pdf (дата звернення 16.12.2023).
43. General Performance Score for classification problems / Isaac Martin De Diego, Ana R. Redondo, Ruben R. Fernandez, Jorge Navarro, Javier M. Moguerza // Applied Intelligence – 2021. - №52. – С.12049-12063. URL: https://doi.org/10.1007/s10489-021-03041-7 (дата звернення 16.12.2023).
44. Amplitude-scan classification using artificial neural networks / Kunal K. Dansingani, Kiran Kumar Vupparaboina, Surya Teja Devarkonda, Soumya Jana, Jay Chhablani & K. Bailey Freund // Nature – Scientific Reports – 2018. - №8. – Стаття 12451. URL: https://www.nature.com/articles/s41598-018-31021-4 (дата звернення 16.12.2023).
45. An Investigation of the Effectiveness of Phase for Audio Classification / Shunsuke Hidaka, Kohei Wakamiya, Tokihiko Kaburagi // ICASSP– 2022. – С.3708-3712. URL: https://doi.org/10.1109/ICASSP43922.2022.9746037 (дата звернення 16.12.2023).
46. Deep Classification of Sound: A Concise Review / S. Bhattacharya, N. Das, S. Sahu, A. Mondal & S. Borah // Proceeding of First Doctoral Symposium on Natural Computing Research – 2021. – С.33-43. URL: http://dx.doi.org/10.1007/978-981-33-4073-2\_4 (дата звернення 16.12.2023).
47. Environmental sound classification using temporal-frequency attention based convolutional neural network / Wenjie Mu, Bo Yin, Xianqing Huang, Jiali Xu & Zehua Du // Nature – Scientific Reports – 2021. - №11. – Стаття 21552. URL: https://www.nature.com/articles/s41598-021-01045-4 (дата звернення 16.12.2023).
48. A two level strategy for audio segmentation / Sébastien Lefèvre, Nicole Vincent // Digital Signal Processing – 2011. - №21. – С.270-277. URL: https://hal.science/hal-00512744/document (дата звернення 17.12.2023).
49. Discrete Fourier transform techniques for noise reduction and digital enhancement of analytical signals / M. Farooq Wahab a, Fabrice Gritti b, Thomas C. O'Haver c // TrAC Trends in Analytical Chemistry – 2021. - №143. – Стаття 116354. URL: https://doi.org/10.1016/j.trac.2021.116354 (дата звернення 17.12.2023).
50. Speech/music classification using phase-based and magnitude-based features / Mrinmoy Bhattacharjee, S.R. Mahadeva Prasanna, Prithwijit Guha // Speech Communication – 2022. - №142. – С.34-48. URL: https://doi.org/10.1016/j.specom.2022.06.005 (дата звернення 17.12.2023).
51. Multi-Representation Knowledge Distillation For Audio Classification / Liang Gao, Kele Xu, Huaimin Wang, Yuxing Peng // Multimedia Tools and Applications – 2022. - №81. – С.5089-5112. URL: https://arxiv.org/pdf/2002.09607.pdf (дата звернення 17.12.2023).
52. Keras Developer Guide / François Chollet // Keras – 2023. URL: https://keras.io/guides/sequential\_model (дата звернення 17.12.2023).
53. Simple audio recognition: Recognizing keywords // TensorFlowOrg – 2023. URL: https://www.tensorflow.org/tutorials/audio/simple\_audio (дата звернення 17.12.2023).
54. Multivariate Time Series Forecasting with LSTMs in Keras / Jason Brownlee // Machine Learning Mastery – 2020. URL: https://machinelearningmastery.com/multivariate-time-series-forecasting-lstms-keras (дата звернення 17.12.2023).
55. LSTM Networks for Time Series Forecasting / Jason Brownlee // Machine Learning Mastery – 2020. URL https://machinelearningmastery.com/use-features-lstm-networks-time-series-forecasting (дата звернення 17.12.2023).
56. Dropout: A Simple Way to Prevent Neural Networks from Overfitting / Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov // Journal of Machine Learning Research – 2014. - № 15. – С.1929. URL: https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf (дата звернення 17.12.2023).
57. Adam: a method for stochastic optimization / Diederik P. Kingma, Jimmy Lei Ba // 3rd International Conference for Learning Representations – 2015. URL: https://arxiv.org/pdf/1412.6980.pdf (дата звернення 17.12.2023).
58. Cross-Entropy Loss Functions: Theoretical Analysis and Applications / Anqi Mao, Mehryar Mohri, Yutao Zhong // ICML – 2023. - № 992. – С. 23803. URL: https://ar5iv.labs.arxiv.org/html/2304.07288 (дата звернення 17.12.2023).
59. Multiple levels of linguistic and paralinguistic features contribute to voice recognition / Jean Mary Zarate, Xing Tian, Kevin J. P. Woods, David Poeppel // Scientific Reports – 2015. - №5. – С.11475. URL: https://www.nature.com/articles/srep11475 (дата звернення 17.12.2023).
60. Data segmentation algorithms: Univariate mean change and beyond / Haeran Cho, Claudia Kirch // Econometrics and Statistics – 2021. - №1. URL: https://doi.org/10.1016/j.ecosta.2021.10.008 (дата звернення 21.12.2023).
61. Scikit-learn: Machine Learning in Python / Fabian Pedregosa, Ga ̈el Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Mu ̈ller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos // Journal of Machine Learning Research – 2011. - №12. URL: https://arxiv.org/pdf/1201.0490.pdf (дата звернення 21.12.2023).
62. How to make models more useful / C. Michael Barton, Allen Lee, Marco A. Janssen, H. R. Albert Jagers // Earth, Atmospheric and planetary sciences – 2022. - №35. – Стаття 119. URL: https://doi.org/10.1073/pnas.2202112119 (дата звернення 21.12.2023).
63. Optimized derivative fast Fourier transform with high resolution and low noise from encoded time signals: Ovarian NMR spectroscopy / Dzevad Belkic & Karen Belkic // Journal of Mathematical Chemistry – 2023. – Стаття 1553. URL: Journal of Mathematical Chemistry Article (дата звернення 21.12.2023).
64. Enhancing threshold neural network via suprathreshold stochastic resonance for pattern classification / Xiaojie Liu, Lingling Duan, Fabing Duan, François Chapeau-Blondeau, Derek Abbott // Physics Letters A – 2021. - №403. – Стаття 127387. URL: https://doi.org/10.1016/j.physleta.2021.127387 (дата звернення 21.12.2023).
65. A systematic review of machine learning classification methodologies for modelling passenger mode choice / Tim Hillel, Michel Bierlaire, Mohammed Z.E.B. Elshafie, Ying Jin // Journal of Choice Modelling – 2021. - №38. – Стаття 100221. URL: https://doi.org/10.1016/j.jocm.2020.100221 (дата звернення 21.12.2023).
66. Accuracy versus reliability-based modelling approaches for medical decision making / Sepideh Etemadi, Mehdi Khashei // Computers in Biology and Medicine – 2022. - №141. Стаття 105138. URL: https://doi.org/10.1016/j.compbiomed.2021.105138 (дата звернення 21.12.2023).
67. Precision magnetic field modelling and control for wearable magnetoencephalography / Molly Rea, Niall Holmes, Ryan M. Hill, Elena Boto, James Leggett, Lucy J. Edwards, David Woolger, Eliot Dawson, Vishal Shah, James Osborne, Richard Bowtell, Matthew J. Brookes // NeuroImage – 2021. - №241. Стаття 118401. URL: https://doi.org/10.1016/j.neuroimage.2021.118401 (дата звернення 21.12.2023).
68. About SampleSwap.org / Canton Becker // Sample Swap – 2023. URL: https://sampleswap.org/about/ (дата звернення 25.12.2023).
69. Data augmentation approaches for improving animal audio classification / Loris Nanni, Gianluca Maguolo, Michelangelo Paci // Ecological Informatics – 2020. - №57. – Стаття 101084. URL: https://doi.org/10.1016/j.ecoinf.2020.101084 (дата звернення 25.12.2023).
70. Comparison of the prediction accuracy of machine learning algorithms in crosslinguistic vowel classification / Georgios P. Georgiou // Nature - Scientific Reports – 2023. - №13. – Стаття 15594. URL: https://doi.org/10.1038/s41598-023-42818-3 (дата звернення 25.12.2023).
71. Machine Learning for Science: State of the Art and Future Prospects / Eric Mjolsness, Dennis Decoste // Science – 2021. - №293. – Стаття 5537. URL: http://dx.doi.org/10.1126/science.293.5537.2051 (дата звернення 26.12.2023).
72. Interpreting the location data extracted from the Apple Health database / Luke Jennings, Matthew Sorell, Hugo G. Espinosa // Forensic Science International: Digital Investigation – 2023. - №44. – Стаття 301504. URL: https://doi.org/10.1016/j.fsidi.2023.301504 (дата звернення 26.12.2023).
73. A new data augmentation method to use in machine learning algorithms using statistical measurements / Emre Avuçlu // Measurement – 2021. - №180. – Стаття 109577. URL: https://doi.org/10.1016/j.measurement.2021.109577 (дата звернення 26.12.2023).
74. An efficient dataflow accelerator for scientific applications / Xiaochun YeXu, Tan Meng, Meng Wu // Future Generation Computer Systems – 2020. - №1. – Стаття 112. URL: http://dx.doi.org/10.1016/j.future.2020.03.023 (дата звернення 26.12.2023).
75. Usability engineering / Jakob Nielsen // Morgan Kaufmann – 1994. – ISBN: 0-12-518406-9. URL: <https://uweb.engr.arizona.edu>/~ece596c/lysecky /uploads/Main/Lec9.pdf (дата звернення 27.12.2023).

**ДОДАТОК А**

ЗАТВЕРДЖЕНО:

Зав. кафедрою ПЗАС, професор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Голуб С.В.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 р.

Програмне забезпечення комунікатор тварин

**Специфікація**

482.ЧДТУ 232219-01

Листів 2

Розробник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гогулов Я.В.

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Голуб С.В.

Контроль \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Півень О.Б.

2023

|  |  |  |
| --- | --- | --- |
| **Позначення** | **Найменування** | **Примітка** |
| 482.ЧДТУ 232219-01 34 01 | Інструкція користувача |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# ДОДАТОК Б

Програмне забезпечення комунікатор тварин

Інструкція користувача

482.ЧДТУ 232219-01 34 01

Листів 2

Розробник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гогулов Я.В

1. Встановлення Python 3.9:

* завантажте та встановіть Python 3.9 з офіційного веб-сайту.

Під час установки обов'язково встановіть опцію "Add Python 3.x to PATH".

1. Встановлення PyCharm IDE:

* завантажте та встановіть PyCharm IDE з офіційного веб-сайту.
* запустіть PyCharm IDE.

1. Створення проекту та встановлення бібліотек:

* створіть новий проект у PyCharm.
* відкрийте термінал PyCharm та встановіть необхідні бібліотеки і модулі за допомогою команди:
* Copy code
* pip install назва\_бібліотеки
* Приклад: pip install flask

1. Запуск сервера:

* запустіть файл server.py у PyCharm для запуску серверної частини програми.

1. Вибір браузера:

* відкрийте будь-який веб-браузер з підтримкою флеш-плеєра. рекомендовано використовувати Mozilla Firefox.

1. Запуск веб-додатка:

* у веб-браузері введіть адресу, яку вказав сервер (зазвичай це http://localhost:порт).
* запустіть веб-додаток, натиснувши відповідний пункт меню чи кнопку на сторінці.

1. Використання комунікатора тварин:

* після запуску веб-додатку виберіть одну з доступних моделей.
* спілкуйтеся з комунікатором, використовуючи введення та отримуючи результати.

Зауваження: Переконайтеся, що у вас встановлені всі необхідні бібліотеки та модулі перед запуском сервера та веб-додатка.