

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Спосіб для аналізу якості дитячого харчування за його складом на пакуванні нейромережевими засобами

Галузь знань _____ 12 – Інформаційні технології _____
Шифр і назва галузі знань
Спеціальність _____ 122 – Комп'ютерні науки _____
Шифр і назва спеціальності
Освітня програма _____ Комп'ютерні науки _____
Назва освітньої програми

Виконав: _____ студент 4 курсу, група КН-19-1 _____ М.С. Штойко _____
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: _____ к.т.н., доцент кафедри КН _____ І.В. Драч _____
Науковий ступінь, посада Підпис Ініціали, прізвище

Нормоконтроль: _____ к.т.н., доцент кафедри КН _____ Р.О. Багрій _____
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:

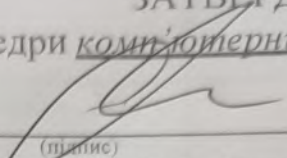
Зав. кафедри КН, д.т.н., професор

5 06 2023 р.

_____ О.В. Бармак _____
Підпис Ініціали, прізвище

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет інформаційних технологій
Кафедра комп'ютерних наук
Освітній ступінь бакалавр
Галузь знань 12 – Інформаційні технології
Спеціальність 122 – Комп'ютерні науки
Освітня програма освітньо-професійна програма підготовки бакалавра

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук


(підпис)
д.т.н., професор О.В. Бармак

06 . 03 «2023 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Спосіб для аналізу якості дитячого харчування за його складом на пакуванні нейромережевими засобами»

2. Завдання видано студенту Штойку Миколі Сергійовичу
(прізвище, ім'я, по батькові)

3. Керівник роботи доцент кафедри КН Драч Ілона Володимирівна
(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від «01» 03 2023 р. № 5

5. Дата видачі завдання студенту: «03» 03 2023 р.

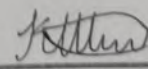
6. Зміст пояснювальної записки (перелік задач) та вихідні дані: *провести аналіз предметної області, здійснити огляд методів автоматизованого зчитування певних символів з використанням нейронної мережі; розробити спосіб для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі; визначити послідовність застосування способу для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі; розробити інформаційну систему з практичної реалізації розробленого способу з використанням сучасних засобів проектування та розробки програмного забезпечення.*

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником	грудень 2022	Виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	січень 2023	Виконано
3	Робота над розділом 1 – Характеристика предметної області та постановка задачі	січень 2023	Виконано
4	Робота над розділом 2 – Спосіб для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі	березень 2023	Виконано
5	Робота над розділом 3 – Програмна реалізація способу для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі	квітень 2023	Виконано
6	Оформлення пояснювальної записки згідно вимог	травень 2023	Виконано
7	Попередній захист кваліфікаційної роботи бакалавра	травень 2023	Виконано
8	Захист кваліфікаційної роботи бакалавра на засіданні Екзаменаційної комісії	червень 2023	Виконано

Виконавець:

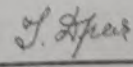
студент 4 курсу, група КН-19-1
Курс, група виконавця


Підпис

М.С. Штойко
Ініціали, прізвище

Керівник:

к.т.н., доцент кафедри КН
Науковий ступінь, посада


Підпис

І.В. Драч
Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: Спосіб для аналізу якості дитячого харчування за його складом на пакуванні нейромережевими засобами

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-19-1 Штойко Микола Сергійович

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КН Драч Ілона Володимирівна

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
64	21	1	27	2

Метою кваліфікаційної роботи бакалавра полягає в розробці способу для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі.

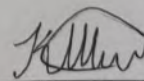
Для досягнення поставленої мети визначені наступні задачі дослідження: опис послідовності застосування способу для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі; розробити спосіб для аналізу якості дитячого харчування за його складом на пакуванні; спроектувати та реалізувати інформаційну систему запропонованого способу.

Результатом виконання кваліфікаційної роботи бакалавра є спосіб для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі.

Ключові слова: аналіз тексту, автоматизоване зчитування, оцінка якості зчитування.

Виконавець:

студент 4 курсу, група КН-19-1
Курс, група виконавця


Підпис

М.С. Штойко
Ініціали, прізвище

Зміст

Перелік скорочень	3
Вступ	4
Розділ 1 Характеристика предметної області та постановка задачі.....	5
1.1 Аналіз предметної області.....	5
1.2 Аналіз відомих метрик та методів оцінки якості автоматизованого зчитування текстової інформації	8
1.3 Аналіз існуючих рішень для подібних систем	13
1.4 Мета, задачі та вимоги до реалізації інформаційної системи.....	17
Розділ 2 Спосіб..... для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі.....	19
2.1 Використання нейронної мережі для аналізу якості дитячого харчування за даними на пакуванні щодо складу продукту	19
2.2 Функціональна структура способу	24
2.3 Проектування структури інформаційної системи.....	31
2.3.1 База даних.....	38
2.4 Висновки до розділу 2	41
Розділ 3 Програмна реалізація способу для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі.....	43
3.1 Структура модулів системи, їх взаємозв'язок	43
3.2 Особливості реалізації способу для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі.....	48
3.3 Опис функціональних можливостей інформаційної системи	52
3.4 Висновки до розділу 3	60
Висновки	62
Перелік посилань	63
ДОДАТКИ.....	65

Перелік скорочень

Скорочення, термін, позначення	Пояснення
БД	База даних
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
ROC-AUC	Receiver Operating Characteristic - Area Under Curve
AUC	Area Under the Curve
BLEU	Bilingual Evaluation Understudy
LOOCV	Leave-one-out cross-validation
CNN	Convolutional Neural Network
DNN	Deep Neural Network
OCR	Optical Character Recognition

Вступ

Кваліфікаційна робота бакалавра присвячена розробці способу для аналізу якості дитячого харчування за його складом на пакуванні нейромережевими засобами.

Актуальність. Аналіз складу дитячих продуктів харчування за допомогою нейромережових методів, що містяться на їх пакуваннях, є дуже актуальною темою дослідження, оскільки правильне харчування дитини є важливим чинником для її здоров'я та розвитку. Водночас, багато продуктів харчування, які призначені для дітей, містять шкідливі речовини, або ж мають неправильний баланс харчових компонентів, що може негативно впливати на здоров'я дитини.

Мета і завдання кваліфікаційної роботи. Мета кваліфікаційної роботи полягає в розробці способу для аналізу якості дитячого харчування за його складом на пакуванні нейромережевими засобами.

Для досягнення поставленої мети визначені наступні завдання дослідження:

- дати опис визначення якості автоматизованого зчитування текстової інформації з використанням нейронної мережі;
- розробити структуру способу для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі;
- реалізувати інформаційну систему для аналізу якості дитячого харчування за його складом на пакуванні нейромережевими засобами;
- провести експериментальне тестування розробленого способу.

Об'єкт дослідження – процес оцінки якості автоматизованого зчитування текстової інформації.

Предмет дослідження – моделі, методи, алгоритми та засоби для створення способу аналізу якості дитячого харчування за його складом на пакуванні нейромережевими засобами.

Розділ 1 Характеристика предметної області та постановка задачі

1.1 Аналіз предметної області

Дитяче харчування - це продукти харчування, які призначені для дітей різного віку та розвитку, з метою забезпечення нормального фізичного та психічного розвитку. Такі продукти мають специфічні вимоги щодо їх складу, калорійності, вітамінно-мінерального складу та інших факторів, які впливають на здоров'я дитини.

Нейромережеві засоби є потужним інструментом для аналізу даних та визначення закономірностей у складі продуктів харчування. Для аналізу якості дитячого харчування можна використовувати нейромережеві засоби, які здатні розпізнавати різні складові продукту, їх кількість та співвідношення.

Застосування нейромережевих засобів дозволяє:

- 1) автоматично аналізувати склад продукту;
- 2) виявляти відхилення від нормативів;
- 3) передбачати якість продукту на основі складу та інших характеристик;
- 4) прогнозувати ефективність продукту в залежності від різних факторів.

Одним з основних переваг використання нейромережевих засобів є їх здатність до самонавчання, що дозволяє їм підвищувати точність аналізу з кожним новим набором даних.

Застосування нейромережевих засобів для аналізу якості дитячого харчування може бути корисним для батьків та фахівців з питань дитячого харчування. Вони зможуть використовувати результати аналізу для вибору найбільш якісних продуктів для дітей та порівняння різних брендів та виробників.

Аналіз якості дитячого харчування за його складом можна провести, розглядаючи наступні чотири параметри:

Перший параметр. Вік дитини: Вікові особливості дітей та їхні потреби в поживних речовинах є дуже важливим при забезпеченні якісного дитячого

харчування. Діти різного віку мають різні потреби в поживних речовинах, оскільки їхні організми знаходяться на різних етапах розвитку[10].

Наприклад, немовлята до 6 місяців мають винятково високі потреби в білках, оскільки вони забезпечують швидкий ріст та розвиток дитини. Крім того, в них мало ензимів для розщеплення складних білків, тому вони потребують легко засвоюваних форм білка, який міститься у грудному молоці.

Діти старшого віку, наприклад, мають інші потреби в поживних речовинах. За цей період діти зазвичай зростають менш інтенсивно, тому їхня потреба в білках зменшується. Однак, вони мають більші потреби в енергії, вітамінах та мінералах, таких як залізо та кальцій, для забезпечення правильного фізичного та розумового розвитку.

Також важливо враховувати, що вікові особливості дітей впливають не тільки на їхні потреби в окремих поживних речовинах, але і на те, які продукти і які страви діти можуть з'їдати. Наприклад, немовлята не можуть їсти тверді продукти, а діти дошкільного віку можуть бути вибагливими до текстури та смаку їжі.

Другий параметр. Енергетична цінність харчування: другий параметр стосується забезпечення дітей достатньою кількістю енергії для забезпечення правильного фізичного та розумового розвитку. Діти швидко ростуть і розвиваються, тому їм необхідно отримувати достатню кількість калорій для забезпечення правильного розвитку. Недостатня кількість калорій може призвести до затримки росту та розвитку, порушень фізичного та психічного здоров'я.

Щоб забезпечити достатню кількість калорій, діти повинні отримувати правильний раціон харчування з продуктами, що містять достатню кількість енергії. Наприклад, продукти з високою енергетичною цінністю, такі як м'ясо, риба, яйця, молочні продукти, горіхи та насіння, можуть бути корисними джерелами енергії для дітей..

Третій параметр. Склад харчування: третій параметр стосується необхідності забезпечення дитини достатньою кількістю необхідних поживних речовин. Для правильного функціонування організму дитини потрібно отримувати достатню кількість білків, жирів та вуглеводів, а також необхідну кількість

вітамінів та мінералів[16].

Білки є важливим будівельним матеріалом для тканин організму, таких як м'язи, шкіра, волосся та нігті. Діти повинні отримувати достатню кількість білка для росту та розвитку. Джерелами білків є м'ясо, риба, яйця, молочні продукти, бобові, горіхи та насіння.

Жири також є важливою складовою частиною раціону дитини. Жири необхідні для правильного функціонування мозку та нервової системи. Джерелами здорових жирів є олія, масло, риба та горіхи.

Вуглеводи є головним джерелом енергії для організму. Діти повинні отримувати достатню кількість вуглеводів для забезпечення енергетичних потреб організму. Джерелами вуглеводів є хліб, крупи, фрукти, овочі та цукор.

Крім того, дитина повинна отримувати достатню кількість вітамінів та мінералів для забезпечення здорового фізичного та розумового розвитку. Наприклад, вітамін С необхідний для підтримки імунної системи та здоров'я шкіри, вітамін D необхідний для здорового розвитку кісток, а залізо необхідне для забезпечення достатнього рівня гемоглобіну в крові.

Четвертий параметр. Якість продуктів: якість продуктів є одним з ключових аспектів забезпечення здорового харчування дітей. Діти мають більшу схильність до шкідливого впливу хімічних речовин, що містяться у харчових продуктах. Тому, важливо обирати якісні продукти, які не містять шкідливих домішок, пестицидів та інших небезпечних речовин[10]. Щоб забезпечити якість продуктів, можна звернути увагу на такі фактори:

Виробник: виробник повинен мати добру репутацію та відповідні сертифікати якості продукції. Рекомендується обирати продукти від відомих виробників, які дотримуються стандартів виробництва та контролю якості.

Дата виготовлення та термін придатності: при купівлі продуктів, важливо звернути увагу на дату їх виготовлення та термін придатності. Для збереження максимальної свіжості та якості продукту, рекомендується обирати продукти з максимально можливим терміном придатності.

Вид продукту: важливо обирати продукти з мінімальною кількістю домішок

та оброблених інгредієнтів. Для забезпечення максимальної якості продукту, рекомендується обирати натуральні продукти без штучних добавок та консервантів[11].

Отже дитяче харчування є важливою складовою правильного фізичного та психічного розвитку дитини. Якість дитячого харчування залежить від таких параметрів, як вік дитини, енергетична цінність харчування, склад харчування та якість продуктів. Забезпечення достатньої кількості поживних речовин у раціоні харчування дітей важливо для правильного функціонування організму та його розвитку. Якість продуктів є ключовим аспектом забезпечення здорового харчування дітей, тому важливо обирати якісні продукти з мінімальним вмістом шкідливих речовин.

1.2 Аналіз відомих метрик та методів оцінки якості автоматизованого зчитування текстової інформації

Основні методи та метрики для аналізу якості автоматизованого зчитування текстової інформації включають:

Accuracy (Точність) - це метрика оцінки точності прогнозування, яка вимірює відповідність між прогнозом та фактичним значенням. Вона визначає, наскільки правильно модель передбачає класи для тестового набору даних та обчислюється як відношення кількості правильних передбачень до загальної кількості передбачень, що зробила модель. Ця метрика є простою в обчисленні та інтерпретації, але може бути не досить інформативною, коли класи даних не збалансовані.

Для прикладу, якщо ми маємо 100 прикладів, і наша модель правильно класифікувала 80 з них, то точність моделі становитиме 80%. Однак, Accuracy може бути помилково високим, якщо в наборі даних домінують один клас, тоді модель може просто передбачати цей клас для всіх прикладів, отримуючи високий показник Accuracy, але при цьому не буде в змозі коректно передбачати інші класи [1].

Також, Accuracy не є відповідною метрикою для незбалансованих даних, коли один клас переважає над іншими, оскільки в цьому випадку Accuracy може бути високим, але модель може погано класифікувати менш представлений клас. Тому, при використанні Accuracy для оцінки моделі, важливо звернути увагу на розподіл класів в наборі даних.

Precision (Точність) - вимірює точність системи у визначенні позитивних результатів, тобто відповіді, які система вважає правильними. Precision обчислюється як відношення кількості правильних позитивних результатів до загальної кількості передбачених позитивних результатів.

Для кращого розуміння поняття точності (precision), можна подати приклад. Допустимо, у нас є система, яка класифікує електронні листи на спам і не спам. Якщо система помилково визначає невідповідний електронний лист як спам, то це може створити деякі проблеми для користувача. У такому випадку, точність системи визначається як кількість правильно визначених спам-листів, поділена на загальну кількість визначених спам-листів (правильні та помилкові). Чим більше точність системи, тим менше шансів на помилкову класифікацію спам-листів.

Recall (Повнота) - вимірює, наскільки добре система знаходить всі позитивні результати. Recall обчислюється як відношення кількості правильних позитивних результатів до загальної кількості позитивних результатів у даних [2].

Прикладом може бути задача виявлення спам-листів в електронній пошті. Recall відображає, яку частину спам-листів система виявила з усіх наявних в базі даних спам-листів. Важливою особливістю Recall є те, що він не враховує кількість неправильних результатів. Наприклад, якщо система повертає дуже багато позитивних результатів, в тому числі і неправильних, її Recall може бути високим, але при цьому Precision буде низьким. Це означає, що система виявила більшість позитивних екземплярів, але серед них багато неправильних.

F1 score - це метрика, яка забезпечує збалансовану оцінку якості системи класифікації, оскільки враховує як точність, так і повноту системи. Вона є гармонічним середнім між Precision та Recall і об'єднує інформацію про точність та повноту визначення результатів у системі класифікації. Розраховується F1 score як відношення подвійного добутку Precision та Recall до їх суми, що дає змогу враховувати як точність, так і повноту при оцінці якості системи класифікації. F1 score є корисним для оцінки моделей, де розподіл класів нерівномірний, або коли важлива як точність, так і повнота.

ROC-AUC (Receiver Operating Characteristic - Area Under Curve) - це метрика, яка використовується для оцінки якості бінарної класифікації. Вона вимірює відношення між True Positive Rate (TPR) та False Positive Rate (FPR) при різних

порогах класифікації. ROC-AUC може бути корисним для оцінки якості системи, коли важливі як Precision, так і Recall. ROC-AUC обчислюється як площа під кривою ROC, яка показує залежність між TPR та FPR при різних значеннях порогів класифікації. Чим більше значення ROC-AUC, тим краще якість класифікації, а значення 0,5 вказує на випадковий вибір класу.

TPR вимірює, яка частка правильно класифікованих позитивних прикладів знаходиться серед усіх позитивних прикладів, тобто відповідає на питання "як часто система вірно класифікує позитивні приклади?".

FPR вимірює, яка частка неправильно класифікованих негативних прикладів знаходиться серед усіх негативних прикладів, тобто відповідає на питання "як часто система помилково класифікує негативні приклади як позитивні?".

Коли ROC крива підноситься близько до верхнього лівого кута графіку, то це свідчить про дуже високу якість класифікації, де система має високий TPR та низький FPR. Навпаки, коли крива опускається до діагоналі, то це свідчить про систему, яка не має відмінних відносин з областю класифікації, тобто випадкові передбачення.

AUC (Area Under the Curve) це площа під ROC кривою, яка вимірює загальну якість системи класифікації. Значення AUC може бути від 0 до 1, де 1 відповідає ідеальній класифікації, а 0 відповідає випадковим передбаченням. Використовуючи ROC-AUC, можна порівнювати різні системи класифікації та вибирати найкращу з них. BLEU (Bilingual Evaluation Understudy) - метод, який використовується для оцінки якості машинного перекладу тексту. Він порівнює перекладений текст з декількома референсними текстами та використовує метрику precision з n-gram.

Метод крос-валідації (Cross-validation) - цей метод дозволяє перевірити ефективність моделі, розділивши навчальний набір даних на кілька частин і використовуючи їх для тренування та тестування моделі. Після цього можна обчислити середній показник точності та інші метрики ефективності.

Метод крос-валідації є одним з найбільш поширених методів оцінки ефективності моделей машинного навчання. Його використовують для того, щоб зменшити ризик перенавчання моделі та оцінити її загальну ефективність [3].

Основна ідея полягає в тому, щоб розділити навчальний набір даних на кілька частин, зазвичай 5 або 10, які називаються "фолдами". Потім модель тренують на

одному фолді, а тестують на іншому. Процес повторюється для кожного фолда. Наприкінці, результати тестування об'єднуються, і обчислюється середній показник точності та інші метрики ефективності[27].

Крос-валідація дозволяє дослідити, наскільки добре модель підходить для нових даних, тобто для даних, яких немає в навчальному наборі. Це допомагає зрозуміти, чи перенавчилася модель на навчальному наборі даних, чи вона може ефективно працювати з новими даними.

Крім того, крос-валідація дозволяє визначити оптимальні значення гіперпараметрів моделі. Гіперпараметри - це параметри моделі, які не вивчаються в процесі навчання, а встановлюються вручну перед тренуванням. Використовуючи крос-валідацію, можна знайти оптимальні значення гіперпараметрів, що дозволяє підвищити ефективність моделі.

Метод валідації на інших даних, також відомий як Holdout validation, є одним з найпростіших методів оцінки ефективності моделі машинного навчання. Він включає розділення навчального набору даних на дві частини: тренувальну та валідаційну. Модель навчається на тренувальних даних, а потім перевіряється на валідаційних даних, які раніше не використовувалися для тренування. Цей метод дозволяє оцінити, наскільки добре модель пристосовується до нових даних та допомагає уникнути перенавчання моделі на наявних даних.

Зазвичай тренувальний набір даних складає 60-80% від загальної кількості даних, а валідаційний набір - 20-40%. Модель навчається на тренувальних даних, після чого перевіряється на валідаційних даних, які раніше не використовувалися для тренування.

Оцінка ефективності моделі проводиться шляхом порівняння передбачених значень на валідаційних даних з їх відповідними справжніми значеннями. Метрики, такі як точність, F1-score, ROC-AUC тощо, можуть бути використані для оцінки ефективності моделі.

Перевагою методу валідації на інших даних є простота та швидкість виконання. Однак, у разі обмеженої кількості даних, розбиття на тренувальний та валідаційний набори може призвести до втрати частини інформації, що може призвести до перенавчання або недонавчання моделі. Також важливо враховувати, що результати оцінки можуть бути залежні від випадкового розподілу даних між

тренувальним та валідаційним наборами.

Метод перехресної перевірки (Leave-one-out cross-validation) - цей метод полягає в тому, що кожен елемент навчального набору даних використовується як тестовий набір даних, під час коли всі інші елементи використовуються для тренування моделі.

У методі LOOCV, для кожного елемента навчального набору даних, модель будується на інших елементах, а потім тестується на конкретному елементі, що був залишений. Загальна кількість ітерацій, що потрібні для завершення процесу крос-валідації, дорівнює кількості елементів в навчальному наборі даних.

Метод LOOCV має кілька переваг. Оскільки кожен елемент використовується як тестовий набір даних, метод LOOCV дає точні оцінки ефективності моделі. Крім того, метод LOOCV допомагає уникнути проблеми дисбалансу класів, оскільки він гарантує, що кожен клас буде представлений в тестовому наборі даних.

Однак метод LOOCV також має деякі недоліки. Зокрема, він може бути дуже часо затратним для великих навчальних наборів даних, оскільки для кожної ітерації потрібно побудувати нову модель. Крім того, якщо навчальний набір даних містить шумові або неправильні дані, то ці дані можуть негативно впливати на точність оцінки ефективності моделі[26].

Нижче наведена таблиця 1.1 короткого опису переваг та недоліків методів для аналізу якості дитячого харчування за його складом на пакуванні за допомогою нейромережових засобів та інших метрик.

Таблиця 1.1 - Переваги та недоліки методів та метрик.

Методи та метрики	Переваги	Недоліки
Accuracy	Проста метрика для оцінки точності моделі.	Не показує відносну кількість правильних та неправильних передбачень.
Precision	Показує відношення кількості правильних передбачень до загальної кількості передбачених.	Не враховує кількість помилкових відкидань.

Recall	Показує відношення кількості правильних передбачень до загальної кількості значень.	Не враховує кількість помилкових пропусків.
F1 score	Комбінація precision та recall для оцінки збалансованості моделі.	Не враховує важливість певних типів помилок.
ROC-AUC	Дозволяє порівнювати різні моделі за кривою ROC та площею під нею.	Не показує конкретну точність моделі та може бути менш корисним для несбалансованих даних.
Метод крос-валідації	Дозволяє оцінювати ефективність моделі на різних частинках навчальних даних.	Потребує більшого обчислювального часу та ресурсів.
Holdout validation	Дозволяє оцінювати ефективність моделі на тестових даних.	Потребує достатньої кількості даних для надійної оцінки та може бути менш.
Метод перехресної перевірки	Дозволяє оцінювати ефективність моделі на тестових даних.	Може бути менш точним, оскільки модель не тренувалася на цих даних.

Отже застосування відповідних методів та метрик оцінки дозволяє визначити якість автоматизованого зчитування текстової інформації, тобто виміряти, наскільки точно і повно програмне забезпечення зчитує та інтерпретує текст.

1.3 Аналіз існуючих рішень для подібних систем

Для аналізу якості дитячого харчування за його складом на пакуванні нейромережевими засобами можуть бути використані такі нейромережеві засоби, як глибокі нейронні мережі (deep neural networks), рекурентні нейронні мережі (recurrent neural networks) та згорткові нейронні мережі (convolutional neural networks).

Щодо існуючих рішень, можна відзначити такі:

1. "AI powered Food Scanner" - додаток, що базується на згорткових нейронних мережах. Він дозволяє користувачам сканувати штрих-коди продуктів і отримувати інформацію про їх склад та якість. Додаток має функції оцінки харчової цінності та калорійності продукту, а також здатності визначати наявність алергенів [4].

Для створення "AI powered Food Scanner" була використана згорткова нейронна мережа (Convolutional Neural Network або CNN), яка є однією з найбільш ефективних моделей для обробки зображень.

Для початку було створено набір даних, що містить зображення продуктів та відповідну інформацію про їх склад. Після цього було створено CNN, яка була навчена на цьому наборі даних.

Одна з ключових особливостей CNN є те, що вона може самостійно визначати важливі характеристики продуктів, що можуть бути важливими для аналізу їх складу. Це робить можливим автоматичне визначення складу продуктів на основі зображень.

Для покращення точності результатів було використано техніку transfer learning, що дозволяє навчання CNN на попередньо навченій моделі, яка була натренована на великому наборі зображень. Це покращило точність моделі та зменшило час навчання.

Крім того, для підвищення точності результатів було використано метод аугментації даних, який полягає в збільшенні розміру набору даних за рахунок створення додаткових зображень на основі вже існуючих. Наприклад, можна створити зображення з різними кутами освітлення, розмірами та перспективами, що дозволяє покращити роботу моделі на нових зображеннях.

Для оцінки якості продукту було використано алгоритми обробки даних, що дозволяють визначати харчову цінність, калорійність, наявність алергенів та інші важливі параметри продукту.

У результаті було створено додаток, який дозволяє користувачам швидко та зручно отримувати інформацію про склад продукту та його якість на основі сканування штрих-коду.

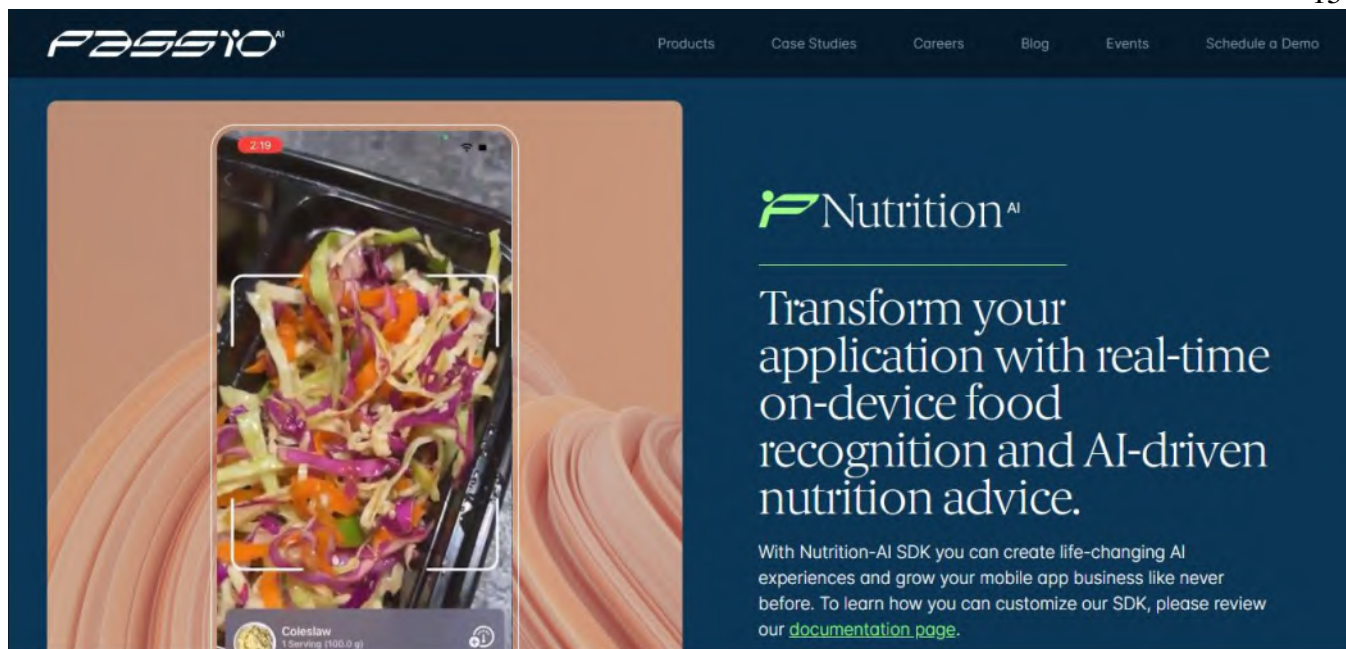


Рисунок 1.1 - Головна сторінка застосунку AI powered Food Scanner

2."FoodAI" - система, що використовує глибокі нейронні мережі для аналізу складу продуктів. Вона складається з декількох модулів, які забезпечують аналіз якісних показників продуктів, таких як вміст вуглеводів, білків, жирів, вітамінів та мінералів. Крім того, система може рекомендувати оптимальний режим харчування на основі аналізу складу продуктів.

Для створення "FoodAI" була використана глибока нейронна мережа (Deep Neural Network або DNN), яка є однією з найбільш потужних моделей для обробки даних [5].

Система "FoodAI" складається з декількох модулів, які відповідають за різні функції. Кожен з цих модулів використовує свою власну DNN для аналізу конкретного аспекту продуктів.

Для аналізу складу продуктів була створена DNN, яка була навчена на великому наборі даних про склад продуктів. Навчання включало в себе процес оптимізації вагів та зсувів нейронів, що дозволило мережі "вивчити" закономірності в даних.

Крім того, для покращення точності результатів була використана техніка Batch Normalization, що дозволяє підвищити стабільність та швидкість навчання мережі.

Кожен з модулів системи "FoodAI" використовує відповідну DNN для аналізу конкретного аспекту продуктів, таких як вміст вуглеводів, білків, жирів, вітамінів та

мінералів. Ці модулі об'єднуються в одну систему, що дозволяє користувачам отримати повну інформацію про склад продуктів та їх якість.

Для рекомендації оптимального режиму харчування на основі аналізу складу продуктів, система "FoodAI" використовує комплексну модель, яка забезпечує аналіз вмісту різних компонентів продуктів та рекомендації з врахуванням потреб користувача. Ця модель також базується на DNN та була навчена на великому наборі даних про харчування та здоров'я людей.

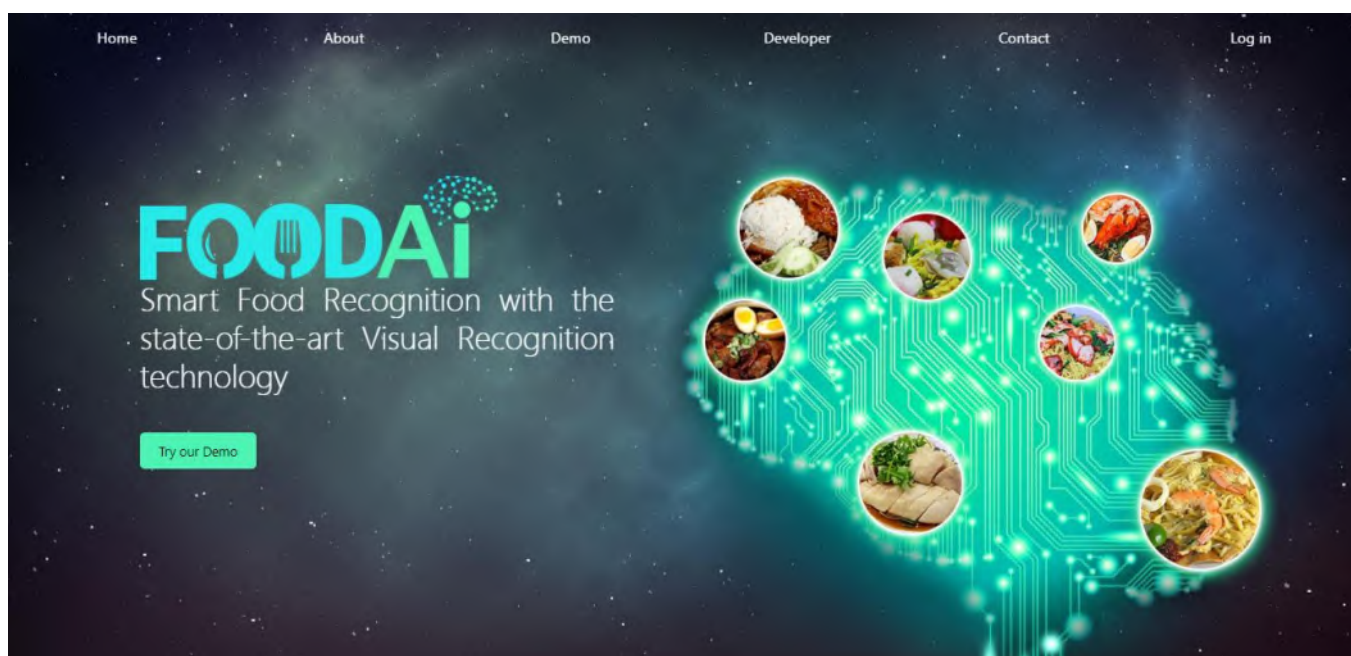


Рисунок 1.2 - Головна сторінка застосунку FoodAI

3. "Foodvisor" - додаток, що використовує рекурентні нейронні мережі для аналізу складу продуктів та оцінки їх харчової цінності. Він має функції відстежування калорійності продуктів та визначення оптимального режиму харчування на основі цих даних.

Для створення додатку "Foodvisor" була використана рекурентна нейронна мережа (RNN), яка є однією з найбільш ефективних моделей для роботи з послідовними даними, такими як текст, звук або часові ряди [6].

Для початку було створено набір даних, що містить інформацію про склад продуктів та їх харчову цінність. Далі була створена RNN, яка була навчена на цьому наборі даних.

Одна з ключових особливостей RNN є здатність до аналізу послідовних даних та збереження інформації про попередні стани внутрішніх шарів. Це робить

можливим автоматичний аналіз складу продуктів та визначення їх харчової цінності на основі послідовності вмісту різних поживних речовин.

Додатково, для покращення точності результатів було використано техніку word embedding, яка дозволяє перетворити слова у вектори чисел. Це допомагає моделі краще розуміти зв'язки між словами та їх значеннями, що в свою чергу покращує точність аналізу складу продуктів та їх харчової цінності.



Рисунок 1.3 - Головна сторінка застосунку Foodvisor

Отож з усіх розглянутих рішень для аналізу якості дитячого харчування за його складом на пакуванні, можна виділити "AI powered Food Scanner", "FoodAI" та "Foodvisor". Ці рішення базуються на використанні глибоких нейронних мереж, які здатні аналізувати інформацію про склад продуктів та давати користувачеві рекомендації щодо дієти або харчування взагалі.

1.4 Мета, задачі та вимоги до реалізації інформаційної системи

Мета кваліфікаційної роботи полягає в розробці способу для аналізу якості дитячого харчування за його складом на пакуванні нейромережевими засобами.

Для досягнення поставленої мети визначені наступні задачі дослідження:

- дати опис визначення якості автоматизованого зчитування текстової інформації з використанням нейронної мережі;

- розробити структуру способу для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі;
- реалізувати інформаційну систему для аналізу якості дитячого харчування за його складом на пакуванні нейромережевими засобами;
- провести експериментальне тестування розробленого способу.

Розділ 2 Спосіб для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі

2.1 Використання нейронної мережі для аналізу якості дитячого харчування за даними на пакуванні щодо складу продукту

Спосіб для аналізу якості дитячого харчування за його складом може бути наступним

Зібрати дані про склад дитячого харчування, включаючи інгредієнти, кількість та відсоток вмісту. Перший пункт передбачає збір даних про склад дитячого харчування. Це означає, що необхідно зібрати інформацію про всі інгредієнти, які містяться в продукті, їх кількість та відсоток вмісту.

Для збору даних можна використати різні джерела інформації, такі як:

1. Інформація на упаковці продукту: на упаковці зазвичай наведені всі інгредієнти, які входять до складу продукту, а також їх кількість та відсоток вмісту. Цю інформацію можна зібрати вручну або використовувати програми для автоматичного зчитування тексту з упаковки.

2. Бази даних про продукти: існують різні бази даних, які містять інформацію про продукти та їх склад. Наприклад, такими базами можуть бути бази даних супермаркетів або спеціалізовані бази даних з інформацією про харчування.

3. Дослідження та наукові публікації: дослідники зазвичай збирають інформацію про склад продуктів у процесі проведення наукових досліджень та публікують цю інформацію у відкритих наукових публікаціях.

У процесі збору даних необхідно враховувати, що склад продукту може змінюватись залежно від країни виробника, бренду, а також залежно від сезону та умов зберігання продукту. Тому важливо зібрати достатньо велику вибірку продуктів, щоб забезпечити точність та репрезентативність аналізу.

Підготувати дані для використання в нейронній мережі, наприклад, перетворити текстову інформацію в числові значення. При використанні нейронної мережі для аналізу якості дитячого харчування, необхідно підготувати дані для використання в мережі. Оскільки дані про склад харчування зазвичай містять текстову інформацію (наприклад, назви інгредієнтів), а нейронні мережі працюють з числовими значеннями, то необхідно перетворити текстову інформацію в числові

значення.

Для цього можна використовувати метод векторизації тексту, який полягає в тому, що кожне слово (або інгредієнт) замінюється на числовий вектор. Для створення такого вектора можна використовувати різні підходи, наприклад:

1. One-Hot Encoding: Кожне слово замінюється на вектор з нулями та одним "1" на позиції, яка відповідає індексу слова в словнику. Для застосування цього методу до текстових даних, таких як слова, кожне слово замінюється вектором, який має розмірність словника, і всі елементи цього вектора, крім одного, будуть рівні нулю[21].

2. Bag of Words: Кожен документ (у нашому випадку - харчування) представляється як множина слів. За допомогою підрахунку кількості згадувань кожного слова у документі, створюється вектор значень[22].

3. Word2Vec: Кожне слово замінюється на вектор, який відображає його семантичний зміст. Word2Vec (слово вектора) - це алгоритм, що використовується для отримання векторних представлень слів на основі їх семантичного змісту. Він використовує нейромережеві моделі для навчання векторних представлень слів на основі контексту, в якому вони зустрічаються у тексті[23].

Після перетворення текстової інформації в числові значення, дані можна підготувати для використання в нейронній мережі, наприклад, розділити дані на тренувальний та тестовий набори, нормалізувати дані для забезпечення більш стабільного навчання мережі та ін.

Створити нейронну мережу з вхідним, прихованим та вихідним шарами. Створення нейронної мережі зазвичай включає в себе визначення архітектури мережі, а саме кількості шарів та кількості нейронів у кожному з них.

Вхідний шар отримує дані про склад дитячого харчування у вигляді числових значень після попередньої підготовки даних. Приховані шари відповідають за обробку даних та виявлення залежностей між інгредієнтами та показниками якості продукту, використовуючи різні методи навчання, такі як згорткові нейронні мережі (Convolutional Neural Networks), рекурентні нейронні мережі (Recurrent Neural Networks) та інші. Вихідний шар містить нейрони, які відповідають за виведення результатів аналізу у вигляді показників якості продукту[25].

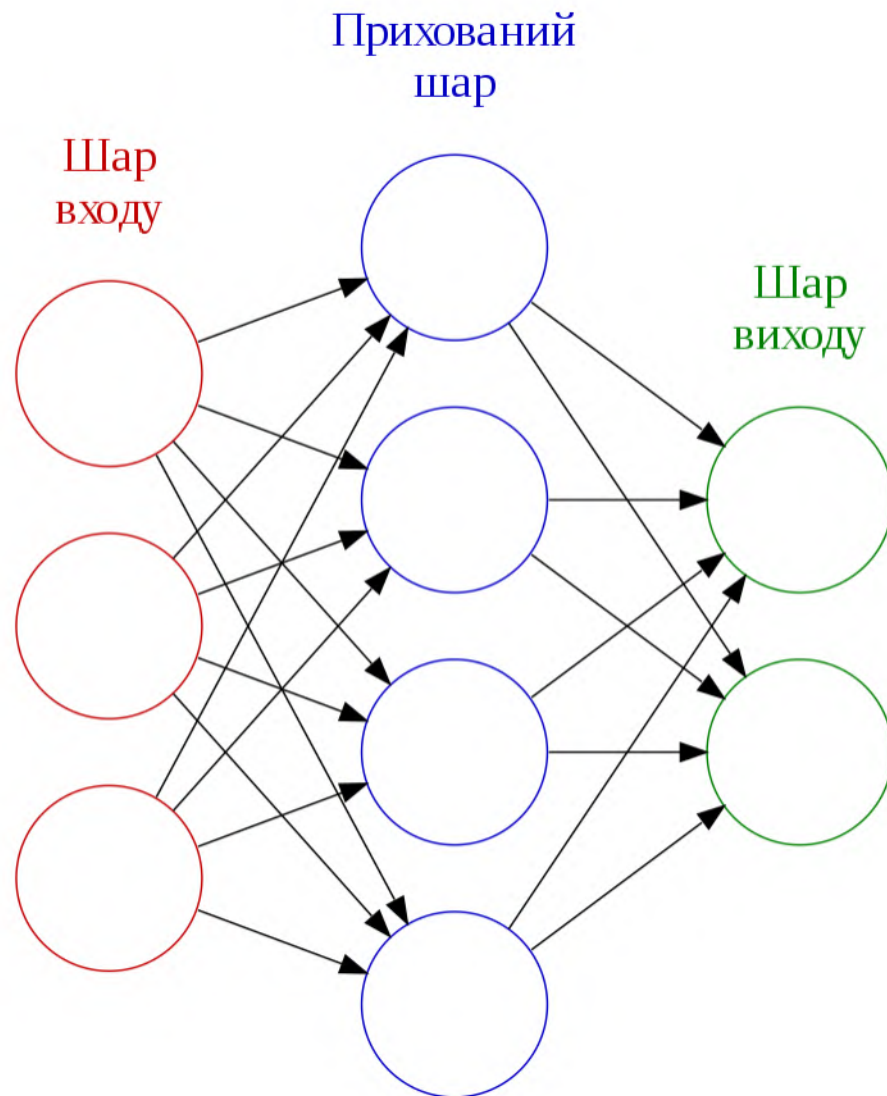


Рисунок 2.1 - Приклад архітектури нейромережі.

Для створення нейронної мережі можна використовувати різні бібліотеки та фреймворки для машинного навчання, такі як Tensorflow, PyTorch, Keras тощо. В процесі розробки моделі можна також використовувати різні методи оптимізації та регуляризації моделі, щоб забезпечити оптимальну ефективність та уникнути перенавчання (overfitting).

Налаштувати параметри нейронної мережі, такі як кількість нейронів у прихованих шарах та швидкість навчання. Параметри нейронної мережі включають різноманітні налаштування, які впливають на її поведінку та ефективність. Налаштування, такі як кількість нейронів у прихованих шарах та швидкість навчання, можуть бути оптимізовані для досягнення кращих результатів.

Кількість нейронів у прихованих шарах може бути визначена

експериментальним шляхом. Загалом, збільшення кількості нейронів у прихованих шарах може збільшити точність нейронної мережі, проте це може також привести до перенавчання (overfitting), коли модель стає дуже точною на тренувальних даних, але менш ефективною на нових даних. Тому необхідно бути обережним при виборі кількості нейронів у прихованих шарах.

Швидкість навчання визначає, наскільки швидко відбувається навчання нейронної мережі. Швидкість навчання може бути налаштована за допомогою методів градієнтного спуску або інших оптимізаційних алгоритмів. Вибір оптимальної швидкості навчання є важливим кроком в процесі навчання нейронної мережі, оскільки занадто висока швидкість може призвести до того, що модель навчатиметься занадто швидко та не зможе дати достатньо точних результатів, а занадто низька швидкість може призвести до того, що навчання триватиме надто довго, що може бути некорисним з практичної точки зору[12].

Інші параметри, які можуть бути налаштовані, включають рівень регуляризації, функції активації, кількість епох навчання та багато іншого

Навчити нейронну мережу на навчальних даних, щоб вона могла виявляти залежності між інгредієнтами та показниками якості продукту.

Після налаштування параметрів нейронної мережі, наступним кроком є навчання мережі на навчальних даних. Під час навчання мережі використовується метод зворотного поширення помилки, який допомагає мережі виявляти залежності між інгредієнтами та показниками якості продукту.

Під час навчання нейронної мережі використовуються навчальні дані, які складаються з інформації про склад дитячого харчування та її показників якості. Навчальні дані подаються на вхід нейронної мережі, і мережа прогнозує показники якості продукту на основі складу продукту.

Під час навчання нейронної мережі на навчальних даних, ваги нейронів в прихованих та вихідних шарах мережі налаштовуються таким чином, щоб зменшити помилки передбачення. Цей процес триває до тих пір, поки мережа не зможе давати достатньо точні прогнози показників якості продукту на основі складу продукту.

Після завершення процесу навчання нейронної мережі, можна перевірити її точність на тестових даних, які не використовувалися під час навчання мережі. Якщо точність на тестових даних є задовільною, то нейронна мережа може бути

застосована для аналізу якості дитячого харчування за його складом на пакуванні.

Перевірити ефективність нейронної мережі на тестових даних, щоб переконатися, що вона здатна правильно визначати якість продукту на основі його складу.

Після того, як нейронна мережа навчилася на навчальних даних, необхідно перевірити її ефективність на тестових даних. Для цього необхідно взяти певну кількість тестових даних з відомими показниками якості продукту і використати навчену нейронну мережу для прогнозування якості продукту на основі їх складу.

Після того, як мережа зробила свій прогноз, необхідно порівняти його з фактичними показниками якості продукту. Це можна зробити за допомогою різноманітних метрик, таких як точність, чутливість, специфічність та F1-оцінка. Ці метрики допомагають оцінити ефективність нейронної мережі на тестових даних.

Якщо результати перевірки ефективності нейронної мережі на тестових даних задовільні, то можна вважати, що мережа здатна правильно визначати якість продукту на основі його складу. Однак, якщо результати не задовільняють, то необхідно повернутися до налаштування параметрів мережі та оптимізації її архітектури для досягнення кращих результатів.

Використовувати навчену нейронну мережу для аналізу якості нових продуктів на основі їх складу, використовуючи інформацію, надруковану на пакуванні.

Після успішного навчання та перевірки ефективності нейронної мережі, можна використовувати її для аналізу якості нових продуктів на основі їх складу, використовуючи інформацію, надруковану на пакуванні. Для цього потрібно зібрати дані про склад нового продукту та підготувати їх для використання в нейронній мережі, перетворивши текстову інформацію в числові значення. Далі, запустити нові дані через навчену нейронну мережу, щоб отримати прогнозовану якість продукту[19].

В результаті аналізу нейронною мережею можна отримати різноманітну інформацію про якість продукту, таку як смакові характеристики, консистенцію, хімічний склад тощо. Це допоможе виробникам збільшити якість продукту та підвищити задоволеність клієнтів. Крім того, цей метод можна використовувати для перевірки відповідності декларованої інформації на пакуванні до фактичної якості

продукту, що дозволяє забезпечити дотримання стандартів та правил щодо харчових продуктів та їх безпеки.

Таким чином, спосіб для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі може допомогти споживачам зрозуміти, які продукти є безпечними та здоровими для дітей на основі їх складу та характеристик.

2.2 Функціональна структура способу

Функціональна структура нейромережі для аналізу якості дитячого харчування за його складом складається з декількох шарів, кожен з яких виконує певну функцію.

Tesseract-OCR - це безкоштовний і відкритий оптичний розпізнавач символів, який використовується для розпізнавання тексту з зображень. Для розпізнавання тексту Tesseract використовує глибокі нейронні мережі, які працюють за допомогою декількох етапів [8].

Перший етап - передобробка зображення. У цьому етапі виконується попереднє оброблення зображення, таке як зменшення шуму, збільшення контрастності і бінаризація зображення, щоб підготувати його для подальшої обробки.

Другий етап - розпізнавання тексту. У цьому етапі виконується розпізнавання тексту з обробленого зображення. Tesseract використовує глибокі нейронні мережі для розпізнавання символів, слів та рядків[17].

Третій етап - постобробка тексту. У цьому етапі виконується подальше оброблення розпізнаного тексту, таке як корекція помилок, форматування тексту, розділення тексту на рядки та слова.

Кожен етап містить багато алгоритмів і технік обробки зображень та розпізнавання тексту. Глибокі нейронні мережі, що використовуються в Tesseract, навчені на великих наборах даних, що дозволяє їм розпізнавати текст з високою точністю. В цілому, Tesseract-OCR здатний ефективно розпізнавати текст з багатьох типів зображень, включаючи зображення з різними розмірами, шрифтами та якістю[18].



Рисунок 2.2 - Внутрішня архітектура OCR Tesseract

Діаграма внутрішньої архітектури OCR Tesseract може включати наступні етапи обробки зображення та розпізнавання тексту[7]:

1. Оригінальне зображення: Початкове зображення, що містить текст, вводиться в систему для подальшої обробки.

2. Адаптивна бінаризація: Зображення піддається адаптивній бінаризації, що дозволяє розділити текст та фон на чорно-білі пікселі. Це допомагає виділити контури символів та полегшує їх подальше розпізнавання.

3. Бінарне зображення: Після бінаризації отримуємо бінарне зображення, де символи представлені чорним кольором, а фон - білим.

4. Аналіз компонентів фаза: На цьому етапі виконується аналіз компонентів зображення, де виявляються окремі області, що містять текстові компоненти. Це можуть бути окремі символи або групи символів, такі як слова або рядки.

5. Виявлення контурів: Компоненти тексту проходять процес виявлення контурів, що допомагає точно визначити межі кожного символу або текстової області.

6. Виявлення слів, рядків, абзаців: На цьому етапі визначаються групи символів, які утворюють окремі слова, рядки або абзаци. Це допомагає встановити логічну структуру тексту.

7. Організація слова: Компоненти тексту, що належать до одного слова, організовуються у відповідному порядку для подальшого розпізнавання.

8. Розпізнавання в два кроки: На цьому етапі застосовуються методи розпізнавання символів для перетворення зображення символу в текстове представлення.

9. Редагований документ: Після розпізнавання символів і перетворення їх у текстове представлення, отримується результуючий редагований документ. Це означає, що текст, який був розпізнаний з зображення, структурований у вигляді окремих слів, рядків та абзаців і може бути використаний для подальшого аналізу, пошуку або збереження у відповідному форматі (наприклад, текстовий файл, база даних тощо)[18].

Ця діаграма внутрішньої архітектури OCR Tesseract відображає послідовність етапів обробки зображення та розпізнавання тексту. Кожен етап виконує певні операції для виділення та розпізнавання текстової інформації з зображення. Завдяки цій архітектурі, Tesseract-OCR може ефективно розпізнавати текст і перетворювати його на редагований документ[9].

Архітектура неймережі Tesseract-OCR складається з декількох шарів, які виконують різні функції під час розпізнавання тексту.



Рисунок 2.3 - Архітектура неймережі Tesseract-OCR

Основні шари, які використовуються в Tesseract-OCR, включають:

Вхідний шар (Input Layer): Це перший шар неймережі, який приймає вхідні дані - оброблене зображення, на якому потрібно розпізнати текст.

Процес розпізнавання тексту починається з підготовки зображення, яке може включати кроки, такі як зменшення шуму, видалення фону, вирівнювання тексту та підготовку контрастності. Після підготовки, зображення передається вхідному шару

нейромережі.

Вхідний шар взаємодіє з пікселями зображення і перетворює їх у формат, зрозумілий для нейромережі. Це може включати перетворення кольорових зображень у градації сірого, бінаризацію (перетворення в чорно-біле зображення) або інші операції обробки залежно від потреб і особливостей Tesseract-OCR.

Вхідний шар передає оброблені дані в нейрони наступного шару нейромережі. У випадку Tesseract-OCR, наступні шари використовуються для подальшої обробки зображення, розпізнавання символів та визначення послідовності тексту.

Важливо відзначити, що конкретна структура та розмір вхідного шару залежить від особливостей завдання розпізнавання тексту і архітектури нейромережі. Вхідний шар може мати різні розміри і формати в залежності від розмірів зображення, кількості каналів кольору або інших параметрів вхідних даних.

Згорткові шари (Convolutional Layers): Ці шари використовуються для виявлення важливих ознак на зображенні, таких як краї, форми та текстурні шаблони. Вони використовують сверточні фільтри, щоб застосовувати фільтрацію та згортку до зображення[17].

Згорткові шари є ключовим елементом архітектури нейромережі для обробки зображень. Вони використовуються для виявлення важливих ознак, таких як краї, форми та текстурні шаблони, на зображенні.

Згорткові шари працюють за допомогою сверточних фільтрів, що є невеликими матрицями ваг, які пройшли через вхідні дані зображення. Кожен фільтр застосовується до невеликої області зображення, і його значення множаться з відповідними пікселями цієї області. Цей процес називається згорткою (convolution).

При згортці фільтр проходить через всю область зображення, що дозволяє виявити локальні ознаки. Наприклад, фільтр може реагувати на вертикальні лінії або кругові форми. Коли фільтр проходить через зображення, він створює карту ознак, в якій кожен піксель представляє значення активації для виявленої ознаки. Кількість фільтрів у згортковому шарі визначається параметрами нейромережі і може бути налаштована відповідно до завдання[18].

Після згортки застосовується операція підвибірки (pooling), яка допомагає зменшити розмір отриманої карти ознак. Зазвичай використовується підвибірка з

максимальним значенням, де вибирається найбільше значення з кожного пікселя певної області. Це дозволяє зберегти найважливіші ознаки і зменшити кількість параметрів мережі.

Шари пулінгу (Pooling Layers): Ці шари використовуються для зменшення розмірності виходу згорткових шарів, зберігаючи при цьому важливі ознаки. Вони звичайно використовують підвибірку (subsampling) або максимальне згортання (max pooling).

Шари пулінгу є важливою складовою архітектури нейромережі для зменшення розмірності виходу згорткових шарів і збереження важливих ознак. Вони допомагають знизити кількість параметрів мережі та обробити великі зображення ефективніше.

Шари пулінгу зазвичай використовують дві основні операції: підвибірку (subsampling) та максимальне згортання (max pooling).

1. Підвибірка (Subsampling): Ця операція зводиться до вибору одного представника з пікселів у певній області. Вона дозволяє зменшити розмірність зображення, зберігаючи приблизно таку ж інформацію. Наприклад, підвибірка може вибирати середнє або медіанне значення пікселів у певному околі. Це допомагає знизити розмірність карт ознак і зменшити кількість параметрів мережі.

2. Максимальне згортання (Max Pooling): Ця операція вибирає максимальне значення з пікселів у певній області. Вона дозволяє виокремити найбільш важливу ознаку, яка була виявлена в згортковому шарі. Наприклад, згортковий шар може виявити границі або форми, і максимальне згортання збереже цю інформацію, використовуючи найбільш активований піксель.

Повнозв'язані шари (Fully Connected Layers): Ці шари приймають вихідні дані з попередніх шарів та виконують класифікацію тексту. Вони з'єднані з кожним нейроном попереднього шару, що дозволяє враховувати залежності між ознаками.

Повнозв'язані шари (Fully Connected Layers), також відомі як повністю з'єднані шари або шари зворотного розповсюдження помилки (backpropagation), є останніми шарами в архітектурі нейромережі. Вони грають важливу роль у класифікації тексту та інших завданнях.

Повнозв'язані шари приймають вихідні дані з попередніх шарів, такими як згорткові шари або шари пулінгу. Кожен нейрон у повнозв'язаному шарі з'єднаний з

кожним нейроном попереднього шару, що дозволяє враховувати залежності між ознаками.

У повнозв'язаному шарі кожен нейрон має свою вагу і зсув, які використовуються для обчислення вагової суми вхідних сигналів. Ця вагована сума проходить через активаційну функцію, таку як сигмоїда або ReLU (Rectified Linear Unit), для визначення активації нейрона. Активація може бути інтерпретована як значення, що вказує на наявність або відсутність певної ознаки.

Повнозв'язані шари зазвичай використовуються для класифікації тексту. Останній повнозв'язаний шар має кількість нейронів, яка відповідає кількості класів або категорій, на які потрібно класифікувати текст. Кожен нейрон у цьому шарі представляє ймовірність належності вхідного тексту до відповідного класу.

Вихідний шар (Output Layer): Це останній шар нейромережі, який видає прогнози або ймовірності класів, які відповідають розпізаному тексту. У випадку Tesseract-OCR це можуть бути символи або слова.

Вихідний шар (Output Layer) є останнім шаром нейромережі і має вирішальне значення в процесі розпізнавання тексту. Його завдання - видавати прогнози або ймовірності класів, які відповідають розпізаному тексту.

У випадку Tesseract-OCR, вихідний шар може містити символи або слова. Зазвичай використовується модель класифікації, де кожен нейрон в вихідному шарі представляє певний символ або слово зі словника. Наприклад, якщо набір можливих символів складається з латинських букв, цифр та деяких спеціальних символів, то кількість нейронів у вихідному шарі буде відповідати кількості цих символів.

Кожен нейрон в вихідному шарі видає значення або ймовірність належності розпізаному тексту до певного символу або слова. Зазвичай для задачі класифікації тексту застосовується функція активації Softmax, яка перетворює вихідні значення нейронів на ймовірності. Сума ймовірностей всіх нейронів у вихідному шарі дорівнює 1[20].

Це опис шарів, які використовуються в нейромережі Tesseract-OCR. В реалізації Tesseract можуть використовуватися більш складні архітектури, такі як рекурентні нейронні мережі (RNN) або довго-короткотермінова пам'ять (LSTM), які допомагають враховувати контекстуальну інформацію.

Додатково, в Tesseract можуть бути застосовані такі техніки:

Препроцесинг зображення: Це включає обробку зображення для покращення якості, таку як розмиття, збільшення контрастності, зменшення шуму та бінаризацію.

Розпізнавання з керуванням контекстом: Tesseract може використовувати контекстуальну інформацію для поліпшення точності розпізнавання, наприклад, враховуючи імовірність певного символу в залежності від його контексту.

Корекція помилок: Після розпізнавання тексту Tesseract може використовувати різні алгоритми для виправлення помилок, такі як виправлення правопису або зіставлення розпізнаних слів зі словником.

Всі ці шари і техніки спільно працюють, щоб досягти високої точності розпізнавання тексту. Tesseract-OCR використовує натреновані моделі на великих наборах даних для розпізнавання символів, слів та тексту. Це дозволяє йому ефективно розпізнавати текст на різних типах зображень з високою точністю. Продовжуючи, в реальній реалізації Tesseract-OCR можуть використовуватися більш складні архітектури, такі як рекурентні нейронні мережі (RNN) або рекурентні згорткові нейронні мережі (RCNN). Ці архітектури дозволяють враховувати контекстуальну залежність між символами та словами, що поліпшує точність розпізнавання [8].

Зокрема, рекурентні нейронні мережі (RNN) використовують зворотний зв'язок, щоб передавати інформацію з попередніх кроків обробки до наступних. Це дозволяє моделі враховувати послідовність символів та їх контекстуальні залежності.

Рекурентні згорткові нейронні мережі (RCNN) комбінують переваги згорткових шарів для виявлення локальних ознак та рекурентних шарів для моделювання контекстуальних залежностей. Це дозволяє враховувати як локальні, так і глобальні ознаки тексту при розпізнаванні.

Крім того, Tesseract-OCR може використовувати додаткові методи та алгоритми для поліпшення точності розпізнавання, такі як адаптивне порогоування, застосування мовних моделей для корекції помилок, використання словних баз для виправлення результатів і т. д.[17].

Загалом, точна архітектура нейромережі Tesseract-OCR може залежати від конкретної версії Tesseract та налаштувань, використаних під час навчання та

розпізнавання. Ці архітектури і методи допомагають Tesseract-OCR досягати високої точності при розпізнаванні тексту з зображень.

2.3 Проектування структури інформаційної системи

Визначення функціональних вимог: на основі результатів аналізу потреб користувачів, необхідно визначити функції, які повинна виконувати інформаційна система. Це може включати збір, обробку і аналіз даних про склад продуктів, виведення результатів аналізу у зручній формі для користувача, інтеграцію з базою даних та інші функції.

Для визначення функціональних вимог зазвичай використовують методологію "Use Case" або "Сценаріїв використання". За допомогою цих методологій можна описати різні сценарії використання системи та визначити функції, які повинна виконувати система, щоб забезпечити успішне виконання сценаріїв.

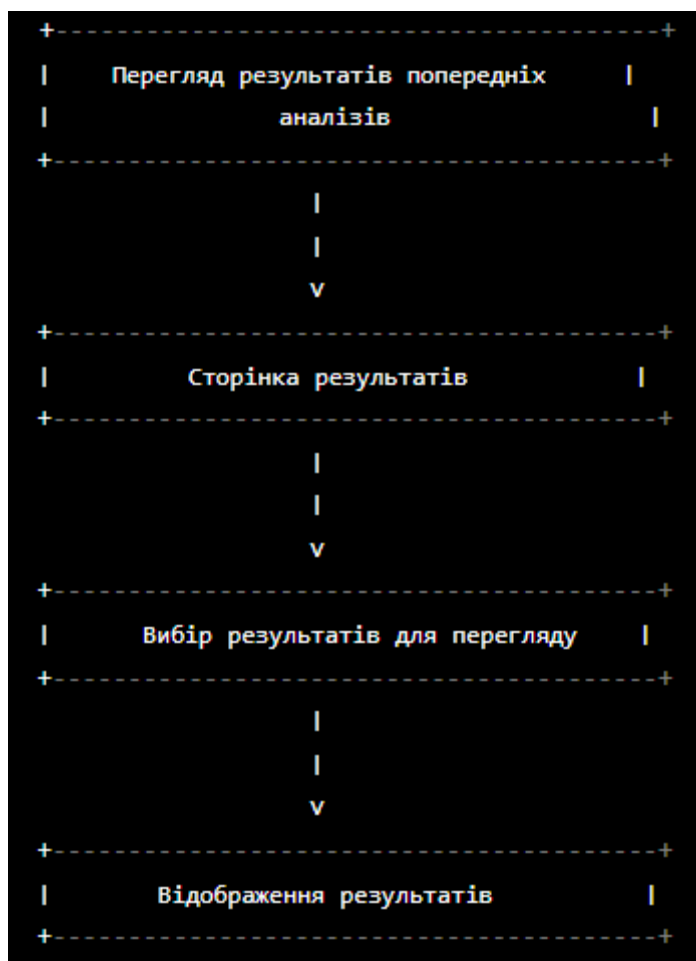


Рисунок 2.4 - Use Case перегляд результатів

Use Case 1: Перегляд результатів попередніх аналізів

Опис: Користувач може переглянути результати попередніх аналізів, включаючи назву продукту, категорію, дату аналізу та результати.

Передумова: Користувач має доступ до системи і знаходиться на сторінці результатів.

Дії користувача:

1. Користувач відкриває сторінку результатів.
2. Користувач вибирає результати, які хоче переглянути, зі списку результатів попередніх аналізів.
3. Система відображає результати вибраних аналізів, включаючи назву продукту, категорію, дату аналізу та результати.

Результат: Користувач може переглянути результати попередніх аналізів.

Для інформаційної системи, що аналізує якість дитячого харчування, визначені наступні функції:

1. Збір даних про склад продуктів з пакування
2. Обробка і аналіз зібраних даних
3. Визначення якості продуктів за їх складом
4. Порівняння якості продуктів з нормативними вимогами
5. Відображення результатів аналізу користувачам у зручній формі
6. Забезпечення можливості збереження результатів аналізу для майбутнього використання

Окрім цього, на етапі визначення функціональних вимог також визначаються вимоги до інтерфейсу користувача, системних вимог та інших вимог, необхідних для успішного виконання функцій системи.

Визначення не функціональних вимог: окрім функціональних вимог, необхідно визначити не функціональні вимоги, такі як надійність, швидкість роботи, безпека, зручність використання та інші, які впливають на ефективність інформаційної системи.

Визначення не функціональних вимог є важливою частиною проектування інформаційних систем, оскільки вони впливають на якість, ефективність та задоволення користувачів. Нефункціональні вимоги описують властивості системи, які не пов'язані безпосередньо з її функціональністю, але мають велике значення для

успішної реалізації та використання системи.

Основні нефункціональні вимоги включають наступні аспекти:

Надійність:

- Система повинна бути стабільною та безперебійною, здатною працювати без збоїв та відновлюватися в разі виникнення помилок.

- Втрата даних повинна бути мінімізована, а система повинна мати механізми резервного копіювання та відновлення інформації.

- Швидкість:

- Система повинна працювати ефективно та забезпечувати задоволення користувачів швидким відгуком на їхні дії.

- Запити до бази даних, обробка даних та завантаження сторінок повинні здійснюватися максимально швидко.

- Безпека:

- Система повинна мати механізми захисту, щоб запобігти несанкціонованому доступу до конфіденційної інформації та злому системи.

- Дані повинні бути шифрованими, а користувачам повинно бути надано відповідні рівні доступу залежно від їхніх прав.

- Зручність використання:

- Система повинна бути простою у використанні та зрозумілою для широкого кола користувачів, незалежно від їхнього рівня технічної підготовки.

- Інтерфейс користувача повинен бути зручно розроблений, з урахуванням зручності та ергономіки. Важливо, щоб користувачі могли легко зорієнтуватися у системі, знаходити потрібні функції та виконувати завдання без зайвих зусиль.

Для досягнення зручності використання можуть бути застосовані наступні підходи та принципи:

1. Інтуїтивний інтерфейс: Інтерфейс користувача повинен бути логічним і легко зрозумілим для користувачів. Елементи керування, такі як кнопки, пункти меню та форми, повинні мати зрозумілі підказки та мітки, що допомагають користувачам розуміти їх призначення.

2. Консистентність: Усі частини системи повинні мати однаковий стиль, оформлення та розташування елементів.

3. **Мінімалізм:** Інтерфейс повинен бути простим і не завантажувати користувача зайвими елементами або інформацією. Важливо зосередитися на головних функціях і відображати лише необхідну інформацію[14].

4. **Попередження та підказки:** Система може надавати користувачам попередження, підказки або пояснення для допомоги у виконанні завдань. Це можуть бути контекстні підказки, валідатори для перевірки правильності введених даних або сповіщення про важливі події.

5. **Простота навігації:** Система повинна мати чітку та просту навігаційну структуру, щоб користувачі могли легко переходити між різними розділами та функціями. Меню, посилання та кнопки повинні бути розташовані логічно та зрозуміло, щоб користувачі могли швидко знайти потрібні їм розділи та функції.

6. **Відповідність контексту:** Інтерфейс повинен враховувати контекст використання і надавати відповідні опції та дії. **Масштабованість:** Система повинна бути готовою до масштабування та розширення в разі збільшення обсягу обробки даних. Це означає, що система повинна бути спроможною ефективно працювати з великим обсягом даних та забезпечувати швидкий доступ до них навіть при зростанні навантаження. Масштабованість може досягатися шляхом використання розподілених систем, кешування, паралельної обробки, оптимізації запитів та інших технік.

Сумісність: Система повинна бути сумісною з різними операційними системами та браузерами, щоб забезпечити доступність для різних користувачів. Це означає, що система повинна працювати на різних платформах, таких як Windows, macOS, Linux тощо, і бути сумісною з різними веб-браузерами, такими як Google Chrome, Mozilla Firefox, Microsoft Edge та інші. Вона повинна забезпечувати коректну роботу та відображення інтерфейсу на різних пристроях та розширеннях екрану.

Управління: Система повинна мати можливість керування правами доступу, налаштуваннями та збереження історії змін. Це означає, що система повинна надавати можливість адміністраторам налаштовувати права доступу користувачів до різних функцій і ресурсів системи. Вона також повинна забезпечувати можливість налаштування параметрів системи, таких як мова, тема оформлення, часові зони тощо. Збереження історії змін дозволяє відстежувати та відновлювати попередні

стани системи та її налаштувань.

Ці аспекти масштабованості, сумісності та управління грають важливу роль у забезпеченні успішної роботи системи та задоволення потреб користувачів.

Розробка інтерфейсу користувача: необхідно розробити зручний та зрозумілий інтерфейс для користувачів системи. Інтерфейс має бути легким у використанні та надавати користувачам можливість зручно взаємодіяти з системою.

Концепція інтерфейсу Способу для аналізу якості дитячого харчування за його складом на пакуванні буде базуватись на принципах простоти та зручності використання.

Інтерфейс користувача реалізований з використанням бібліотеки Tkinter для створення графічного інтерфейсу, OpenCV для обробки зображень, PyTesseract для оптичного розпізнавання символів та SQLite3 для зберігання даних у базі даних.

Після підключення до бази даних та заповнення списків з таблиць "useful_products", "average_products" і "bad_products"[14], інтерфейс має такі можливості:

1. Перегляд зображень та розпізнавання тексту:
 - Користувач може відкрити зображення, використовуючи кнопку "Відкрити файл".
 - Після вибору зображення відбувається обробка зображення, конвертація його в чорно-білий формат та оптичне розпізнавання символів за допомогою PyTesseract.
 - Отриманий текст розбивається на окремі слова.
2. Аналіз складу продуктів та відображення результатів:
 - Знайдені слова порівнюються зі словами зі списків "useful_products", "average_products" та "bad_products".
 - Співпадіння розділяються на три категорії: "Корисні продукти", "Середні продукти" і "Шкідливі продукти".
 - Результати категоризації відображаються відповідно в віджетах Listbox.
3. Відображення додаткової інформації про продукт:
 - При виборі продукту відповідному списку відбувається виклик функції show_product_info.
 - Відображається вікно з ім'ям продукту та його описом.

4. Вихід з програми:

- Кнопка "Вихід" закриває вікно програми і завершує роботу програми.

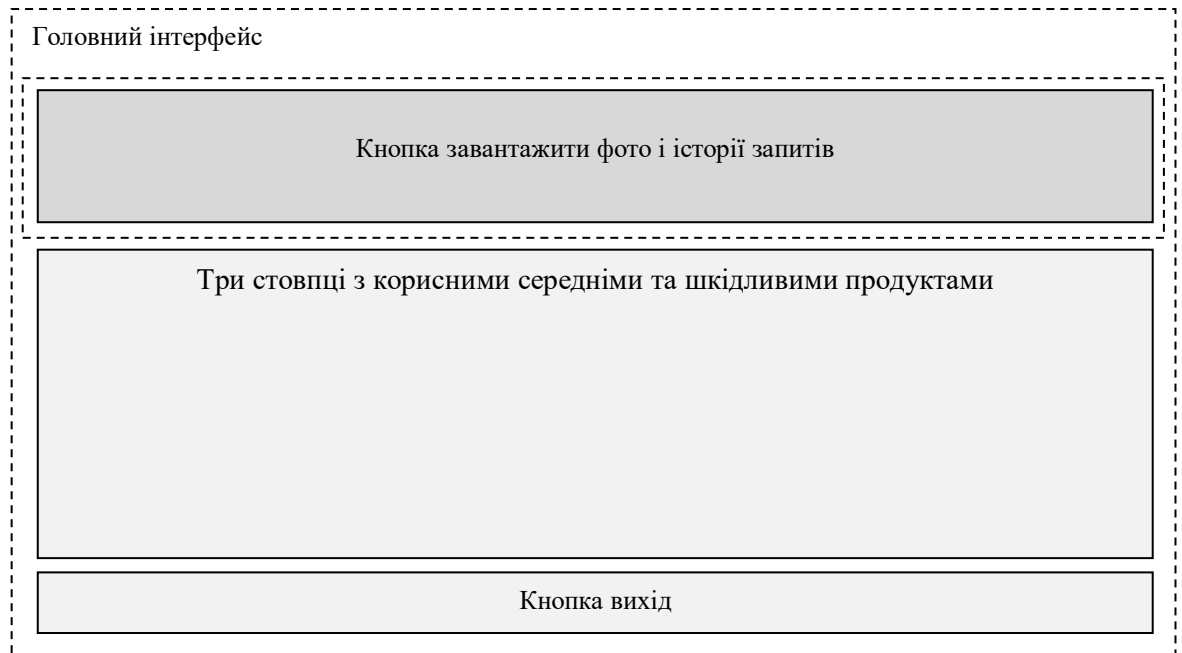


Рисунок 2.5 - Головний інтерфейс

5. Збереження історії запитів:

- При виконанні пошукового запиту зберігається слово та дата запиту в таблиці "search_history" бази даних SQLite.
- Кнопка "Історія запитів" відкриває окреме вікно, де виводиться історія запитів з таблиці "search_history".
- Кнопка "Очистити історію" видаляє всі записи з таблиці "search_history" та очищує віджет Listbox з історією запитів.

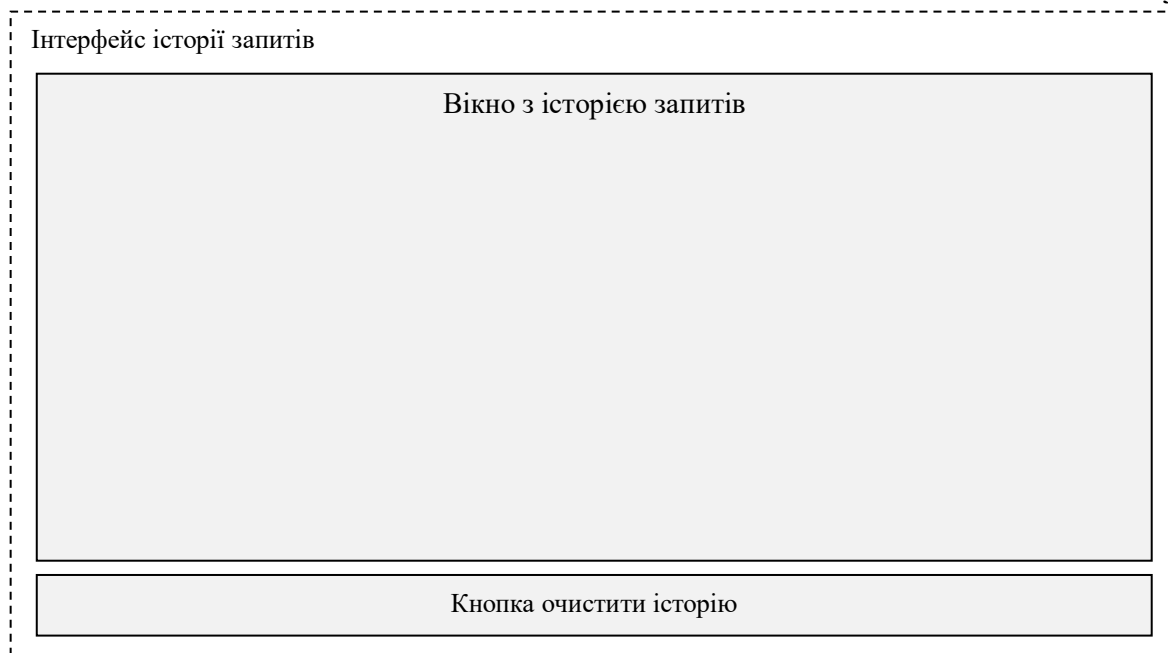


Рисунок 2.6 - Інтерфейс історії запитів

Цей інтерфейс користувача дозволяє користувачеві вибирати зображення, розпізнавати на ньому текст та отримувати інформацію про категорію продуктів, до якої відноситься знайдений текст. Також користувач може переглянути історію запитів та очистити її за потреби[15].

Також можуть бути додаткові функції, наприклад, можливість сортувати результати за датою аналізу або категорією продукту, або можливість експортувати результати в форматі CSV або PDF.

В цілому, проектування інтерфейсу інформаційної системи для аналізу якості дитячого харчування за його складом на пакуванні нейромережевими засобами повинно забезпечувати легкий та зрозумілий доступ користувачів до функцій системи, а також забезпечувати конфіденційність та безпеку персональних даних користувачів.

Розробка тестової системи є важливою складовою процесу розробки будь-якої інформаційної системи, в тому числі системи для визначення якості дитячого харчування. Тестова система дозволяє перевірити правильність роботи програмного забезпечення та алгоритмів, що використовуються в системі.

Для тестування моделі нейронної мережі для аналізу якості дитячого харчування за його складом на пакуванні, ми використаємо тестовий датасет, що складається з зображень продуктів харчування та їх складу.

Метрикою точності, яку ми використовуємо для оцінки нашої моделі, є ассурансу - це відношення кількості правильно класифікованих зображень до

загальної кількості зображень в тестовому датасеті.

Крім того, ми виведемо confusion matrix - це матриця, яка показує, які класи були правильно та неправильно класифіковані моделлю. Для цього ми використаємо бібліотеку Python - scikit-learn, яка містить вбудовані функції для розрахунку метрик класифікації.

2.3.1 База даних

Розробка бази даних Products. Необхідно розробити структуру бази даних, яка буде включати всю необхідну інформацію про дитяче харчування, що збирається з пакування, а також інформацію про результати аналізу.

Система взаємодіє з базою даних для збереження і отримання інформації про продукти, їхні складові та результати аналізу. Вона використовує базу даних для збереження даних про продукти, їхні складові, класифікацію і результати аналізу.

Основні функції взаємодії з базою даних включають:

Запис даних: Система може зберігати інформацію про нові продукти, їхні складові та результати аналізу в базі даних. Це включає збереження даних про назву продукту, його склад, класифікацію і будь-яку іншу важливу інформацію.

Отримання даних: Система може отримувати інформацію про продукти з бази даних для подальшого аналізу. Користувачі можуть отримувати дані про конкретний продукт, його склад, класифікацію або результати попереднього аналізу.

Пошук даних: Система забезпечує можливість пошуку продуктів, за назвою. Це допомагає користувачам швидко знайти необхідну інформацію про продукт.

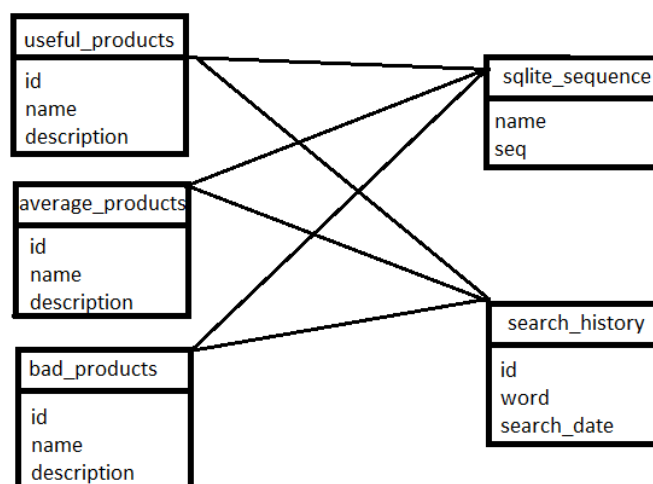


Рисунок 2.7 - База даних Products

Таблиця "useful_products":

- id: унікальний ідентифікатор продукту
- name: назва продукту
- description: опис продукту

useful_products	CREATE TABLE useful_products
id	INT "id" INT
name	VARCHAR(255) "name" VARCHAR(255) NOT NU
description	VARCHAR(255) "description" VARCHAR(255) NC

Рисунок 2.8 - Таблиця useful_products

Таблиця "average_products":

- id: унікальний ідентифікатор продукту
- name: назва продукту
- description: опис продукту.

average_produ...	CREATE TABLE average_pr
id	INT "id" INT
name	VARCHAR(255) "name" VARCHAR(255) NO
description	VARCHAR(255) "description" VARCHAR(255)

Рисунок 2.9 - Таблиця average_products

Таблиця "bad_products":

- id: унікальний ідентифікатор продукту
- name: назва продукту
- description: опис продукту

average_produ...	CREATE TABLE average_pr
id	INT "id" INT
name	VARCHAR(255) "name" VARCHAR(255) NO
description	VARCHAR(255) "description" VARCHAR(255)

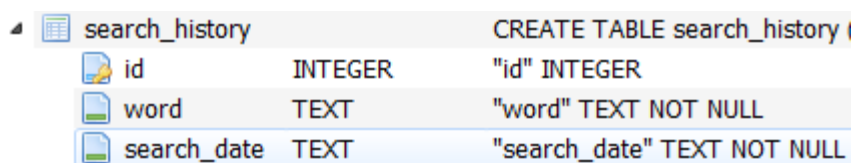
Рисунок 2.10 - Таблиця bad_products

Кожна з цих таблиць містить три поля: "id" для унікальної ідентифікації продукту, "name" для назви продукту та "description" для опису продукту.

Ця структура бази даних надає можливість зберігати інформацію про продукти, які класифікуються як "корисні", "середні" або "погані". Крім цього, ви можете додати до цих таблиць додаткові поля, які відповідатимуть вашим конкретним потребам та вимогам системи.

Таблиця "search_history":

- id: унікальний ідентифікатор запису
- word: слово, за яким здійснювалось пошук
- search_date: дата та час, коли був здійснений пошук

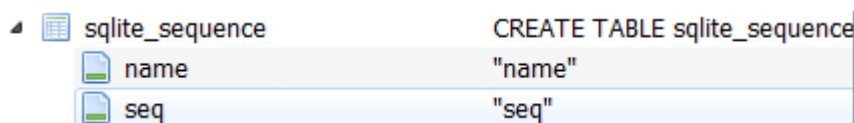


search_history		CREATE TABLE search_history
id	INTEGER	"id" INTEGER
word	TEXT	"word" TEXT NOT NULL
search_date	TEXT	"search_date" TEXT NOT NULL

Рисунок 2.11 - Таблиця search_history

Таблиця "sqlite_sequence":

- name: назва таблиці
- seq: значення останнього ідентифікатора (seq) для відповідної таблиці



sqlite_sequence		CREATE TABLE sqlite_sequence
name		"name"
seq		"seq"

Рисунок 2.12- Таблиця sqlite_sequence

Таблиця "search_history" зберігає історію пошукових запитів, які виконувалися в системі. Кожен запис має унікальний ідентифікатор "id", який слугує для ідентифікації конкретного запиту. Поле "word" містить слово або фразу, за яким здійснювався пошук. Поле "search_date" зберігає дату та час, коли був виконаний пошук.

Таблиця "sqlite_sequence" є вбудованою таблицею у базі даних SQLite, яка використовується для автоматичного збільшення значення ідентифікатора (seq) під час вставки нових записів у таблиці з автоінкрементним полем. Вона містить два поля: "name", яке вказує назву таблиці, і "seq", яке зберігає останнє значення ідентифікатора для цієї таблиці.

Ці дві таблиці доповнюють структуру бази даних, надаючи можливість зберігати історію пошукових запитів та автоматично збільшувати значення ідентифікаторів у таблицях з автоінкрементним полем.

2.4 Висновки до розділу 2

Застосування нейронної мережі для аналізу складу дитячого харчування на пакуванні є перспективним напрямом в галузі харчування. Даний спосіб дозволяє швидко та точно визначати склад продукту, що особливо важливо для дитячого харчування, де безпека та якість є найважливішими аспектами.

У методі для аналізу якості дитячого харчування використовується нейронна мережа, яка навчається розпізнавати залежності між складом продукту та показниками його якості на основі навчальних даних. Цей метод має декілька кроків, таких як збір даних про склад продукту, підготовку даних для використання в нейронній мережі, створення та налаштування самої мережі, навчання та перевірку ефективності на тестових даних, а також використання навченої мережі для аналізу нових продуктів на основі їх складу.

Цей метод може бути корисним для батьків, які хочуть забезпечити своїх дітей якісним харчуванням, а також для виробників дитячих продуктів, які хочуть покращити якість своїх продуктів. Використання нейронної мережі дозволяє автоматизувати процес аналізу якості продукту на основі його складу та забезпечує більш точні результати порівняно з традиційними методами оцінки якості продуктів. Однак, необхідно враховувати, що навчання нейронної мережі потребує значних обчислювальних ресурсів та належної експертизи у галузі машинного навчання для досягнення найкращих результатів.

Функціональна структура нейромережі для аналізу якості дитячого харчування складається з вхідного шару, прихованого шару та вихідного шару. Вхідний шар приймає дані про склад продукту на пакуванні, прихований шар обробляє ці дані, а вихідний шар виводить значення, що відповідає за якість продукту на пакуванні. Нейромережу можна вивчити за допомогою великих наборів даних про якість продуктів на пакуваннях, щоб вона могла застосовуватися для аналізу якості нових продуктів на пакуванні.

Під час проектування інтерфейсу, було враховано потреби користувачів, а також використано сучасні підходи до розробки зручного та інтуїтивно зрозумілого інтерфейсу. Результатом проектування є інформаційна система зі зручним та інтуїтивно зрозумілим інтерфейсом, яка дозволяє виконувати пошук продуктів за їх складом та віковими групами, а також отримувати рекомендації щодо їх споживання.

У результаті, інформаційна система дозволяє батькам та іншим вихователям забезпечити безпечне та якісне харчування для дітей, забезпечує легкий та швидкий доступ до важливої інформації та допомагає підвищити рівень освіченості щодо здорового способу життя та правильного харчування.

Однак, слід зазначити, що аналіз складу продукту на пакуванні з використанням нейронної мережі не замінює повноцінної оцінки якості продукту, здійснюваної відповідними організаціями. Використання даного методу повинно бути доповнене іншими методами оцінки якості продукту та не повинно бути єдиним фактором при виборі дитячого харчування.

Розділ 3 Програмна реалізація способу для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі

3.1 Структура модулів системи, їх взаємозв'язок

Описаний код представляє програму, яка використовує оптичне розпізнавання символів для аналізу зображень з продуктами харчування. Основні кроки програми включають в себе:

1. Імпорт необхідних модулів.
2. Встановлення шляху до Tesseract OCR.
3. Встановлення з'єднання з базою даних "Products" та створення курсора.
4. Виконання запитів до бази даних для отримання списків корисних, середніх та шкідливих продуктів.
5. Оголошення функції `save_search_history(word)`.
6. Оголошення функції `browse_file()`.
7. Оголошення функції `show_product_info(listbox)`.
8. Оголошення функції `show_history()`.
9. Оголошення функції `clear_history(listbox)`.
10. Оголошення функції `main()`.

Далі йде опис функціоналу програми, яка застосовує оптичне розпізнавання символів для аналізу зображень з продуктами харчування. У програмі використовуються модулі для графічного інтерфейсу користувача, роботи з зображеннями та базою даних. Крім того, є можливість зберігати історію запитів та отримувати детальну інформацію про продукти.

Імпортуються необхідні модулі: `tkinter` для графічного інтерфейсу користувача, `cv2` для роботи з зображеннями, `pytesseract` для оптичного розпізнавання символів, `sqlite3` для роботи з базою даних та `filedialog` та `datetime` для роботи з файлами та датами відповідно.

Встановлюється шлях до Tesseract OCR (програмного забезпечення для розпізнавання символів), який використовується пізніше в коді.

Встановлюється з'єднання з базою даних 'B\Products.db' та створюється курсор для виконання запитів до бази даних.

Виконуються запити до бази даних для отримання списків корисних,

середніх та шкідливих продуктів, які зберігаються в словниках з назвою продукту як ключем та описом як значенням.

Функція `save_search_history(word)` відповідає за збереження слова та дати запиту в таблиці `search_history` бази даних. Це забезпечує збереження історії запитів користувачів. Функція виконує INSERT-запит до таблиці `search_history`, додаючи новий запис зі словом та поточною датою.

Основний алгоритм функції `save_search_history(word)` може виглядати так:

1. Встановлення з'єднання з базою даних: Функція встановлює з'єднання з базою даних, яка містить таблицю `search_history`.

2. Створення SQL-запиту: Функція створює SQL-запит типу INSERT, який додає новий запис до таблиці `search_history`. Запит містить значення слова та поточної дати.

3. Виконання SQL-запиту: Функція виконує SQL-запит до бази даних, вставляючи новий запис зі словом та поточною датою в таблицю `search_history`.

4. Закриття з'єднання з базою даних: Після виконання запиту функція закриває з'єднання з базою даних, щоб звільнити ресурси.

Функція `browse_file()` викликає діалогове вікно, яке дозволяє користувачеві вибрати зображення. Обране зображення конвертується в чорно-білий формат, застосовується оптичне розпізнавання символів за допомогою Tesseract OCR, текст розділяється на окремі слова, які потім порівнюються зі списками корисних, середніх та шкідливих продуктів. Збіги зберігаються у віджеті Listbox, а також викликається функція `save_search_history(word)` для збереження історії запитів.

Основний алгоритм функції `browse_file()` може виглядати так:

1. Виклик діалогового вікна вибору файлу: Функція викликає діалогове вікно, яке дозволяє користувачеві вибрати зображення.

2. Конвертація зображення у чорно-білий формат: Обране зображення конвертується у чорно-білий формат, щоб спростити подальший процес розпізнавання символів.

3. Застосування Tesseract OCR для розпізнавання символів: Використовуючи Tesseract OCR (Optical Character Recognition), функція розпізнає символи на зображенні та отримує текстове представлення.

4. Розділення тексту на окремі слова: Отриманий текст розділяється на

окремі слова або токени, щоб здійснити подальший пошук збігів зі списками продуктів.

5. Порівняння слів зі списками продуктів: Кожне слово порівнюється зі списками корисних, середніх та шкідливих продуктів, щоб визначити, до якої категорії воно належить.

Функція `show_product_info(listbox)` відображає детальну інформацію про обраний продукт, коли користувач вибирає слово зі списку. Якщо слово належить до списку корисних продуктів, відображається його опис, аналогічно для середніх та шкідливих продуктів.

Основний алгоритм функції `show_product_info(listbox)` може виглядати так:

1. Отримання обраного слова зі списку: Функція отримує обране слово користувачем з віджета `Listbox`.

2. Перевірка категорії продукту: Функція перевіряє, до якої категорії (корисних, середніх або шкідливих) належить обране слово. Це можна зробити, порівнюючи обране слово зі списками продуктів.

3. Відображення інформації: Залежно від категорії, до якої належить обране слово, функція відображає відповідну детальну інформацію про продукт. Це може бути опис продукту, його корисні властивості або шкідливі наслідки.

Функція `def show_history` створює нове вікно для відображення історії запитів та викликає запит до бази даних для отримання всіх запитів з таблиці `search_history`, виводить їх у віджет `Listbox`. Також функція містить кнопку для очищення історії, яка видаляє всі записи з таблиці `search_history` та очищає віджет `Listbox`.

Основні кроки функції:

1. Створення нового вікна з назвою "Історія запитів".

2. Створення віджету `Listbox` для відображення історії запитів.

3. Виконання запиту до бази даних для отримання всіх запитів з таблиці `search_history`.

4. Виведення історії запитів у віджет `Listbox`.

5. Створення кнопки "Очистити історію".

6. Прив'язка функції `clear_history` до кнопки "Очистити історію".

7. Виведення віджету `Listbox` та кнопки на вікні "Історія запитів".

Функція `clear_history` видаляє всі записи з таблиці `search_history` та очищає

віджет `Listbox`, переданий як аргумент. Для видалення записів використовується запит до бази даних `"DELETE FROM search_history"`, який видаляє всі записи з таблиці `search_history`. Після цього викликається метод `commit()` для збереження змін до бази даних. Для очищення віджету `Listbox` використовується метод `delete()` з аргументами `0` та `tk.END`, який видаляє всі елементи з початку до кінця.

Основний алгоритм функції `clear_history(listbox)` виглядає так:

1. Виконання запиту до бази даних: Функція виконує запит до бази даних за допомогою методу `execute()`. Запит `"DELETE FROM search_history"` видаляє всі записи з таблиці `search_history`.
2. Збереження змін: Після виконання запиту до бази даних, необхідно зберегти зміни за допомогою методу `commit()`. Це забезпечує фізичне видалення записів з таблиці.
3. Очищення віджету `Listbox`: Для очищення віджету `Listbox`, використовується метод `delete()` з аргументами `0` та `tk.END`. Цей метод видаляє всі елементи з початку (`0`) до кінця (`tk.END`) віджету.

Цей код виконує видалення всіх записів з таблиці `search_history` і очищає віджет `Listbox`, що передається як аргумент функції. Після виклику цієї функції, історія запитів буде повністю очищена і віджет `Listbox` не матиме жодних елементів.

Функція `main()` відображає графічний інтерфейс користувача, включаючи віджет `Listbox`, де відображаються збіги при пошуку продуктів, віджет `Label` для відображення детальної інформації про продукт, віджет `Button` для вибору зображення та віджет `Entry` для введення текстового запиту.

1. Віджет `Listbox`: Цей віджет використовується для відображення збігів при пошуку продуктів. Він дозволяє відображати список елементів у вигляді тексту. Користувач може обирати елементи зі списку. Віджет `Listbox` може бути розміщений у вікні за допомогою методу `pack()` або інших методів розміщення.
2. Віджет `Label`: Цей віджет використовується для відображення детальної інформації про продукт. Він може містити текстову інформацію, яка відображається на екрані. Віджет `Label` може бути розміщений у вікні за допомогою методу `pack()` або інших методів розміщення.
3. Віджет `Button`: Цей віджет представляє кнопку, на яку користувач може натиснути для виконання певних дій. Він може мати текст або зображення, що

відображається на кнопці. Коли користувач натискає на кнопку, виконується певна функція або дія. Віджет `Button` також може бути розміщений у вікні за допомогою методу `pack()` або інших методів розміщення.

4. Віджет `Entry`: Цей віджет представляє поле для введення тексту. Користувач може вводити текст у це поле, наприклад, для введення текстового запиту. Значення, введене в це поле, може бути отримане за допомогою методу `get()` для подальшого використання у програмі. Віджет `Entry` може мати певний початковий текст або може бути порожнім при запуску програми. Коли користувач вводить текст і натискає `Enter` або виконує певну дію, значення введеного тексту можна отримати за допомогою методу `get()`. Це значення може бути використане для подальшого аналізу, обробки або передачі іншим функціям для виконання певних операцій.

Також в `main()` встановлюється шрифт та розмір шрифту для графічного інтерфейсу та встановлюється заголовок вікна.

Всі функції та графічний інтерфейс обгорнуті в блок `try-except` для перехоплення можливих помилок та виведення відповідного повідомлення користувачеві.

Цей код розроблено для розпізнавання тексту на зображенні та порівняння його зі списками корисних, середніх та шкідливих продуктів. Він також дозволяє зберігати історію запитів та переглядати її. Код використовує різні модулі, такі як `tkinter`, `cv2`, `pytesseract`, `sqlite3`, `filedialog` та `datetime`, для графічного інтерфейсу, роботи з зображеннями, оптичного розпізнавання символів, роботи з базою даних та файлами. Він містить функції для конвертації зображення в чорно-білий формат, оптичного розпізнавання символів та порівняння отриманих слів зі списками продуктів. Функція `show_product_info` відображає детальну інформацію про обраний продукт. Функція `show_history` дозволяє відображати історію запитів, а функція `clear_history` дозволяє видалити всю історію запитів. Цей код можна використовувати як основу для розробки подібних програм.

3.2 Особливості реалізації способу для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі

Даний код є програмою для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі. Він містить наступні особливості:

Використання бібліотеки `tkinter` для створення графічного інтерфейсу користувача. Бібліотека `tkinter` є одним з основних інструментів для створення графічного інтерфейсу користувача (GUI) в мові програмування Python. Вона є стандартною бібліотекою для Python і може використовуватись для створення різних типів програм, від простих інтерфейсів до більш складних додатків. Основні особливості і переваги `tkinter`:

1. Простота використання: `tkinter` надає простий і зрозумілий інтерфейс, що дозволяє швидко створювати графічні інтерфейси, навіть початківцям в програмуванні. Вона має інтуїтивний синтаксис і просту модель програмування.

2. Переносимість: `tkinter` доступний на багатьох платформах, таких як Windows, macOS та Linux. Це означає, що програми, створені з використанням `tkinter`, будуть працювати на різних операційних системах без необхідності внесення змін у код.

3. Розширюваність: `tkinter` може бути розширена за допомогою додаткових модулів, які надають додаткові можливості і елементи інтерфейсу. Наприклад, модуль `ttk` (Themed Tkinter) надає більш сучасний вигляд для елементів інтерфейсу, таких як кнопки, поля введення, списки і т. д.

4. Можливості налаштування: `tkinter` надає розширені можливості налаштування зовнішнього вигляду елементів інтерфейсу. Ви можете налаштовувати розміри, кольори, шрифти, розташування та інші атрибути елементів для створення бажаного вигляду програми.

5. Інтеграція з Python: `tkinter` ідеально поєднується з мовою програмування Python. Це означає, що ви можете використовувати всі синтаксичні можливості Python для зручного програмування і взаємодії з інтерфейсом. Ви можете легко інтегрувати функції і класи Python у своє програмування GUI з використанням `tkinter`.

Створення графічного інтерфейсу за допомогою tkinter включає створення основного вікна програми, додавання віджетів (елементів інтерфейсу) до вікна та налаштування їх вигляду та поведінки. У tkinter є багато інших елементів, які можна додавати до вікна, таких як мітки, списки, ползунки, радіокнопки, флажки тощо. Ви можете налаштовувати їх властивості, такі як розмір, кольори, шрифти, розташування та інші. Ви також можете додавати функції, які виконуються при натисканні кнопок або при зміні значень елементів.

З'єднання з базою даних за допомогою модуля sqlite3. Модуль sqlite3 є вбудованим модулем Python, який надає можливість з'єднання з базою даних SQLite і взаємодії з нею з допомогою Python. SQLite є легковаго та швидкодіючим двіжком баз даних, який зберігає дані в локальному файлі на комп'ютері.

Для роботи з базою даних потрібно виконати наступні кроки:

1. Підключитися до бази даних. Для цього потрібно викликати метод `connect()` модуля `sqlite3`. Наприклад:

Тут `example.db` - назва файлу бази даних, до якої ми підключаємося. Якщо файл не існує, то він буде створений автоматично.

2. Створити курсор. Курсор дозволяє виконувати запити до бази даних. Для створення курсора використовується метод `cursor()` об'єкта з'єднання.

3. Виконати запит до бази даних. Для цього використовуються методи курсора, такі як `execute()`, `fetchone()` і `fetchall()`. Наприклад, вибрати всі записи з таблиці `users` можна так:

4. Закрити з'єднання і курсор. Після завершення роботи з базою даних потрібно закрити з'єднання та курсор.

Це базовий сценарій роботи з базою даних SQLite за допомогою модуля `sqlite3`. Для роботи з базою даних вам також можуть знадобитися такі методи, як `commit()` для збереження змін у базі даних та `rollback()` для відміни змін.

Використання бібліотеки `cv2` для роботи з зображеннями та оптичного розпізнавання символів. Бібліотека `cv2` (OpenCV) є однією з найбільш популярних бібліотек для обробки зображень та комп'ютерного зору. Вона містить численні функції для розпізнавання облич, детекції об'єктів, зміни розміру зображень, обрізки та обробки зображень, а також оптичного розпізнавання символів (OCR). Основна мета `cv2` полягає в тому, щоб допомогти розробникам створювати програми

комп'ютерного зору, що дозволяють зрозуміти та аналізувати зображення.

Бібліотека `cv2` (OpenCV) є потужним інструментом для роботи з зображеннями та комп'ютерного зору. Вона надає розширені можливості для обробки зображень, включаючи обрізку, зміну розміру, фільтрацію, розмиття, видалення шуму та багато іншого. Крім того, вона також включає функції для роботи з відео, включаючи захоплення відео з камери, обробку кадрів, виявлення руху тощо.

Одним з найбільш популярних застосувань `cv2` є оптичне розпізнавання символів (OCR). Завдяки спеціальним алгоритмам та моделям, `cv2` дозволяє виявляти та розпізнавати символи на зображеннях або відео. Для цього можна використовувати модуль `cv2.text`, який надає функції для підготовки зображень до розпізнавання, виявлення тексту та його конвертації у текстовий формат[24].

Крім оптичного розпізнавання символів, `cv2` також дозволяє виконувати інші завдання обробки зображень, такі як детекція об'єктів (наприклад, облич, автомобілів або предметів), визначення контурів, розпізнавання образів, побудова гістограм тощо. Завдяки великому набору функцій та алгоритмів, `cv2` використовується у багатьох галузях, включаючи комп'ютерний зір, робототехніку, медичну діагностику, автоматичне розпізнавання номерних знаків, розробку ігор та багато іншого.

Використання бібліотеки `pytesseract` для оптичного розпізнавання символів зображення. Бібліотека `pytesseract` є Python-обгорткою для Tesseract OCR (Optical Character Recognition), відкритої бібліотеки для оптичного розпізнавання символів. Вона дозволяє легко виконувати оптичне розпізнавання символів на зображеннях і отримувати текстовий вихід з них.

Бібліотека `pytesseract` також підтримує багато інших параметрів, які можна використовувати для налаштування розпізнавання, таких як розмір шрифту, порогове значення, розпізнавання чисел і так далі.

У програмі встановлюється шлях до програми Tesseract OCR, яка відповідає за розпізнавання символів на зображеннях. Цей шлях використовується для подальшого виклику оптичного розпізнавання символів за допомогою модуля `pytesseract`.

Збереження історії запитів у таблиці `search_history` бази даних. Збереження

історії запитів у таблиці `search_history` бази даних може бути корисним функціоналом для багатьох додатків. Це дозволяє зберігати інформацію про запити користувачів, щоб потім можна було аналізувати ці дані і покращувати функціональність додатку.

Крім того, збереження історії запитів дозволяє користувачам повернутися до своїх попередніх запитів та результатів. Вони можуть переглядати свої попередні пошукові запити, відкривати їх і знову переглядати результати аналізу продуктів. Це забезпечує зручний доступ до попередньо зроблених аналізів та дозволяє користувачам вести облік своїх виборів щодо дитячого харчування.

Відображення результатів пошуку у віджетах `Listbox`. `Listbox` є одним із віджетів, який можна використовувати для відображення результатів пошуку в програмі. `Listbox` - це вікно зі списком, яке дозволяє користувачеві вибрати один або більше елементів зі списку. Кожен елемент списку може містити текст або графічне зображення. Цей віджет може бути дуже корисним для відображення результатів пошуку, оскільки дозволяє користувачеві вибрати один або кілька результатів.

Віджет `Listbox` в бібліотеці `tkinter` дозволяє відображати список елементів, з якого користувач може вибрати один або кілька елементів. Він часто використовується для відображення результатів пошуку, оскільки зручно показує доступні варіанти та дозволяє користувачеві зробити вибір.

Для створення віджету `Listbox` потрібно визначити його відповідний батьківський контейнер, такий як головне вікно програми або рамка, а потім додати елементи до списку.

Основні методи, які можна використовувати з віджетом `Listbox`:

- `insert(index, *elements)`: Додає елементи до списку у вказаному порядку. Можна передавати як окремі елементи, так і послідовності елементів.
- `delete(start, end=None)`: Видаляє елементи зі списку від початкового індексу до кінцевого індексу. Якщо не вказано кінцевий індекс, видаляється лише елемент з початковим індексом.
- `get(index)`: Повертає текстове значення елемента з вказаним індексом.
- `curselection()`: Повертає кортеж, що містить індекси вибраних елементів.
- `selection_clear(start, end=None)`: Знімає виділення з елементів списку від

початкового індексу до кінцевого індексу. Якщо не вказано кінцевий індекс, знімається виділення лише з елемента з початковим індексом.

Для кращого візуального представлення списку елементів можна використовувати Scrollbar (прокрутка), який дозволяє переглядати список, якщо він не поміщається на екрані. Можливість перегляду історії запитів у окремому вікні. Можливість перегляду історії запитів у окремому вікні є корисною функцією для багатьох додатків. Це дозволяє користувачеві переглянути список всіх запитів, які він робив раніше, і знайти той, який його цікавить. У додатку, який працює з базою даних, історія запитів може бути особливо корисною, оскільки користувачі можуть шукати певні записи, які вони робили в минулому.

Програма для аналізу якості дитячого харчування за його складом на пакуванні є функціональною та корисною. Вона використовує багато потужних бібліотек та модулів, таких як tkinter, sqlite3, cv2 та pytesseract, для створення графічного інтерфейсу, роботи з базою даних, обробки зображень та оптичного розпізнавання символів. Ці компоненти допомагають забезпечити зручність використання, масштабованість системи, сумісність з різними операційними системами та управління правами доступу та налаштуваннями. Також програма забезпечує збереження історії запитів та відображення результатів пошуку у віджетах Listbox. Загалом, ця програма може бути цінним інструментом для аналізу якості дитячого харчування та полегшення прийняття відповідних виборів[13].

3.3 Опис функціональних можливостей інформаційної системи

Інформаційна система "Спосіб для аналізу якості дитячого харчування за його складом на пакуванні нейромережевими засобами" має наступні функціональні можливості.

Деякі з цих можливостей включають:

1. Аналіз складу продуктів: Система використовує нейромережеві алгоритми для аналізу інформації про склад продуктів, яка надрукована на їхній упаковці.

2. Збереження історії запитів користувачів: Система зберігає історію запитів користувачів, що дозволяє їм переглядати попередні аналізи продуктів та результати класифікації.

3. Результати та поради: Система надає користувачам результати аналізу продуктів, включаючи класифікацію їх якості з погляду дитячого харчування.

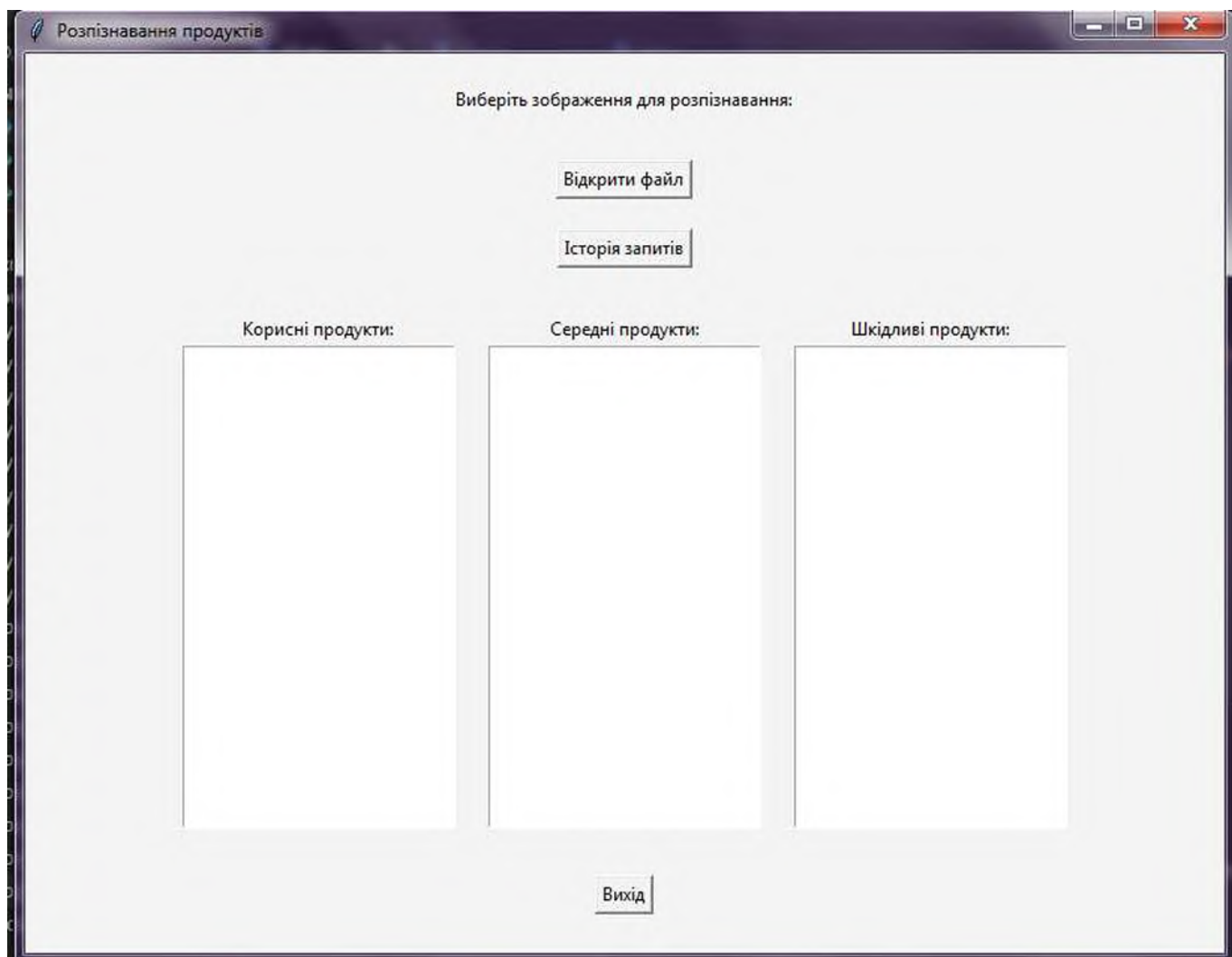


Рисунок 3.1 - Інтерфейс інформаційної системи

Система для аналізу якості дитячого харчування здатна аналізувати інформацію про склад продуктів, яка надрукована на їхній упаковці. Завдяки використанню нейромережових засобів, система може автоматично визначити наявність різних компонентів у дитячому харчуванні. Ці компоненти можуть включати вітаміни, мінерали, білки, жири, вуглеводи та інші складові елементи, які є необхідними для правильного функціонування та розвитку дитячого організму.

Для початку аналізу, користувачу необхідно натиснути на кнопку "Відкрити файл" в системі, що дозволить завантажити фото або зображення складу продукту. Це може бути фотографія упаковки з дитячим харчуванням або склад продукту, вказаний у текстовому форматі. Після завантаження файлу, система застосовує нейромережові алгоритми для аналізу зображення та визначення складових елементів продукту.

Нейромережеві засоби, які використовуються в системі, навчаються розпізнавати та класифікувати різні компоненти, шляхом аналізу великої кількості даних про склад дитячого харчування. Вони розпізнають особливості та шаблони, що відповідають різним складовим елементам продукту. Після аналізу зображення система може вказати, які компоненти присутні в продукті і в якій кількості.

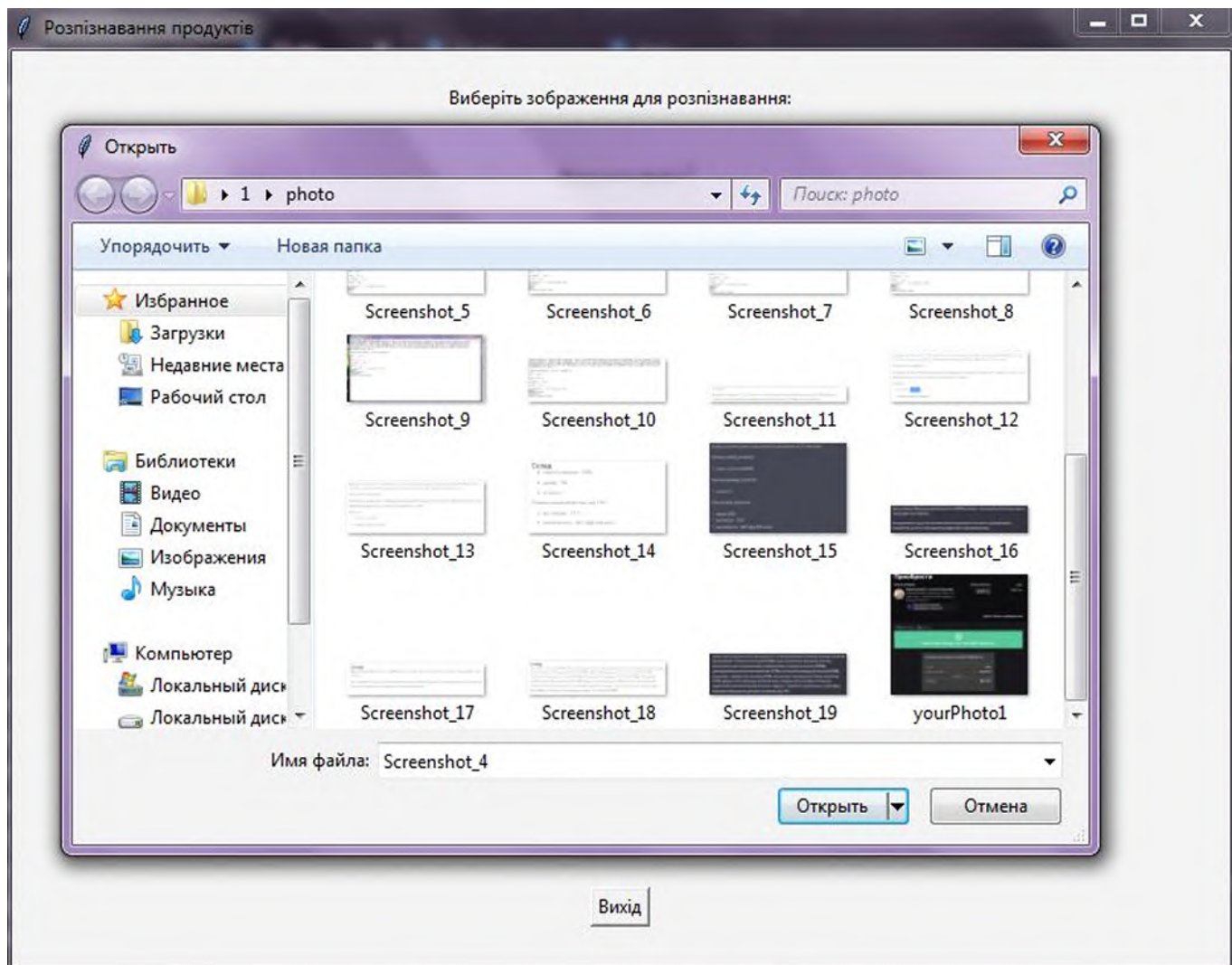


Рисунок 3.2 - Завантаження фото в системи

Система може визначити якість дитячого харчування на основі його складу. Використовуючи нейромережеві алгоритми, система може класифікувати продукти на корисні, середні та погані з погляду їхнього впливу на здоров'я дитини.

Ця класифікація відображає вплив продукту на здоров'я дитини. Корисні продукти містять оптимальні кількості необхідних компонентів та сприяють здоровому росту та розвитку. Середні продукти можуть містити деякі корисні компоненти, але не настільки оптимальні, як у корисних продуктів. Погані продукти

мають недостатню кількість корисних компонентів або можуть містити шкідливі речовини, які можуть негативно вплинути на здоров'я дитини.

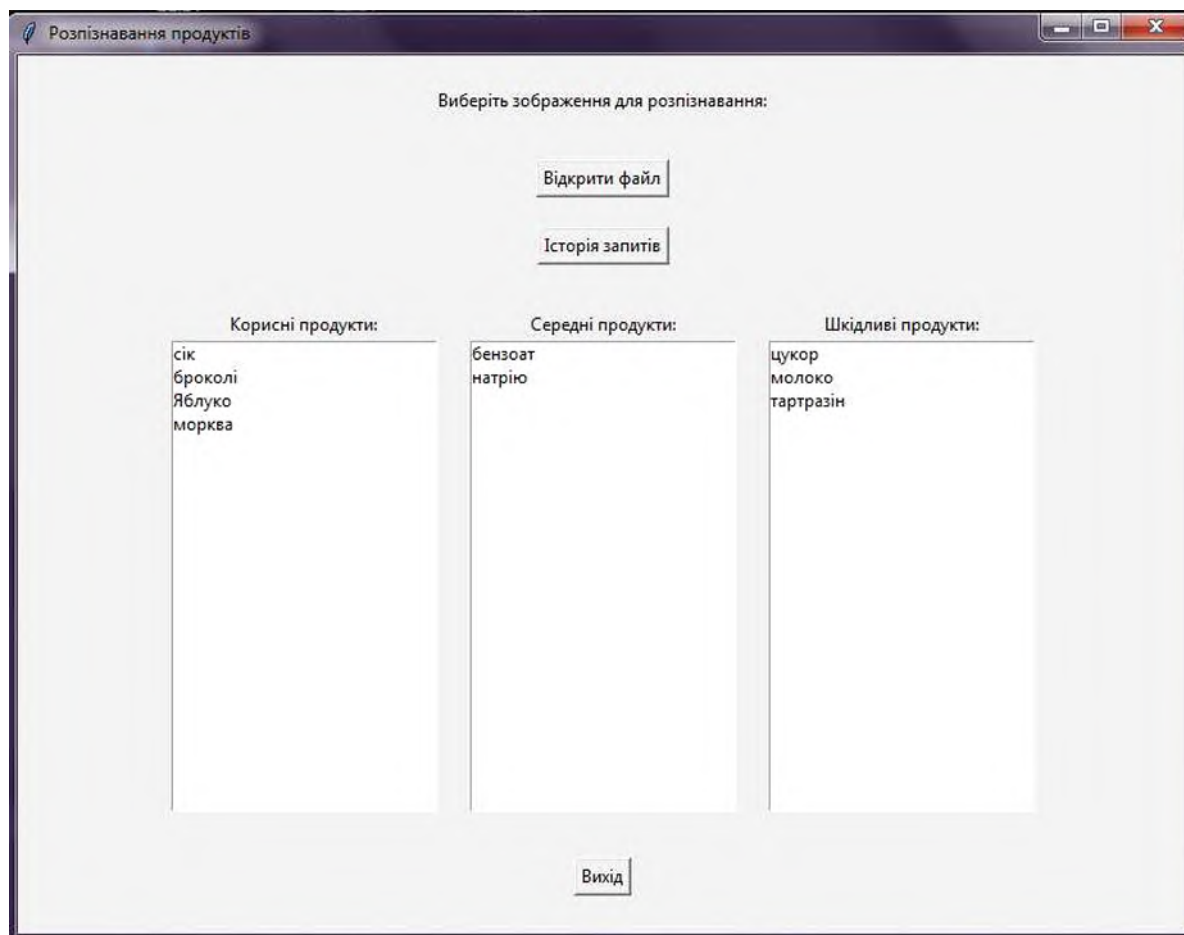


Рисунок 3.3 - Класифікація якості продукту

Результати аналізу і класифікації продуктів представлені користувачу у зручній формі. При натисканні на зацікавивший нас продукт відкривається вікно з інформацією про цей продукт.

Крім того, система надає можливість детальнішого ознайомлення з будь-яким продуктом, що викликає інтерес. Користувач може натиснути на зацікавлений продукт, і відкриється окреме вікно з додатковою інформацією про цей продукт. У цьому вікні можуть бути вказані детальні характеристики, вміст корисних компонентів, рекомендації щодо споживання, можливі побічні ефекти або шкідливі речовини, які можуть бути присутні.

Це дає користувачам можливість отримати більш повну картину про кожен конкретний продукт і прийняти інформоване рішення щодо його використання в дитячому харчуванні. Додаткова інформація допомагає зрозуміти переваги і можливі ризики, пов'язані з продуктом, і забезпечує більшу освіченість користувачів

у питаннях дитячого харчування.

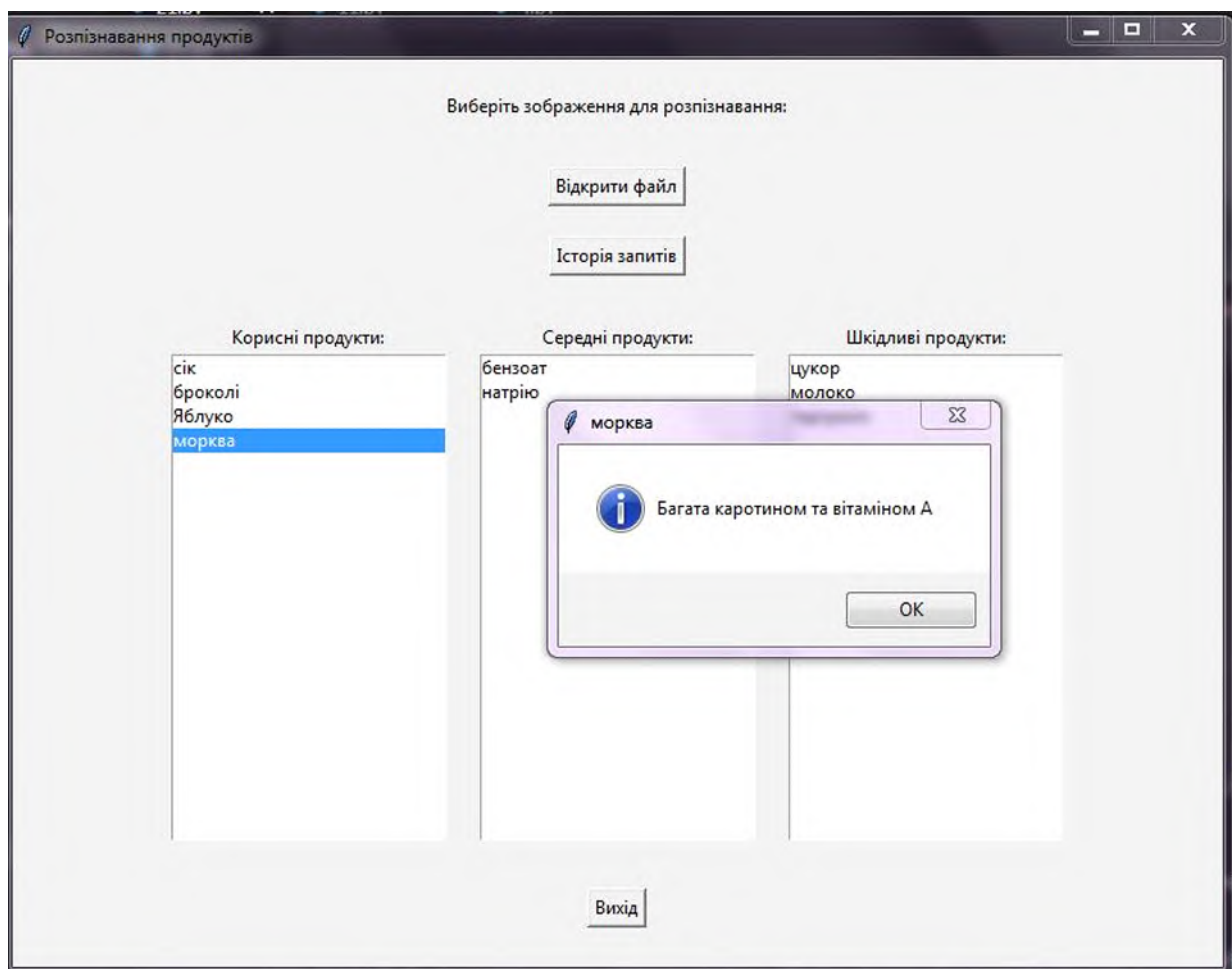


Рисунок 3.4 - Вікно з інформацією про корисний продукт

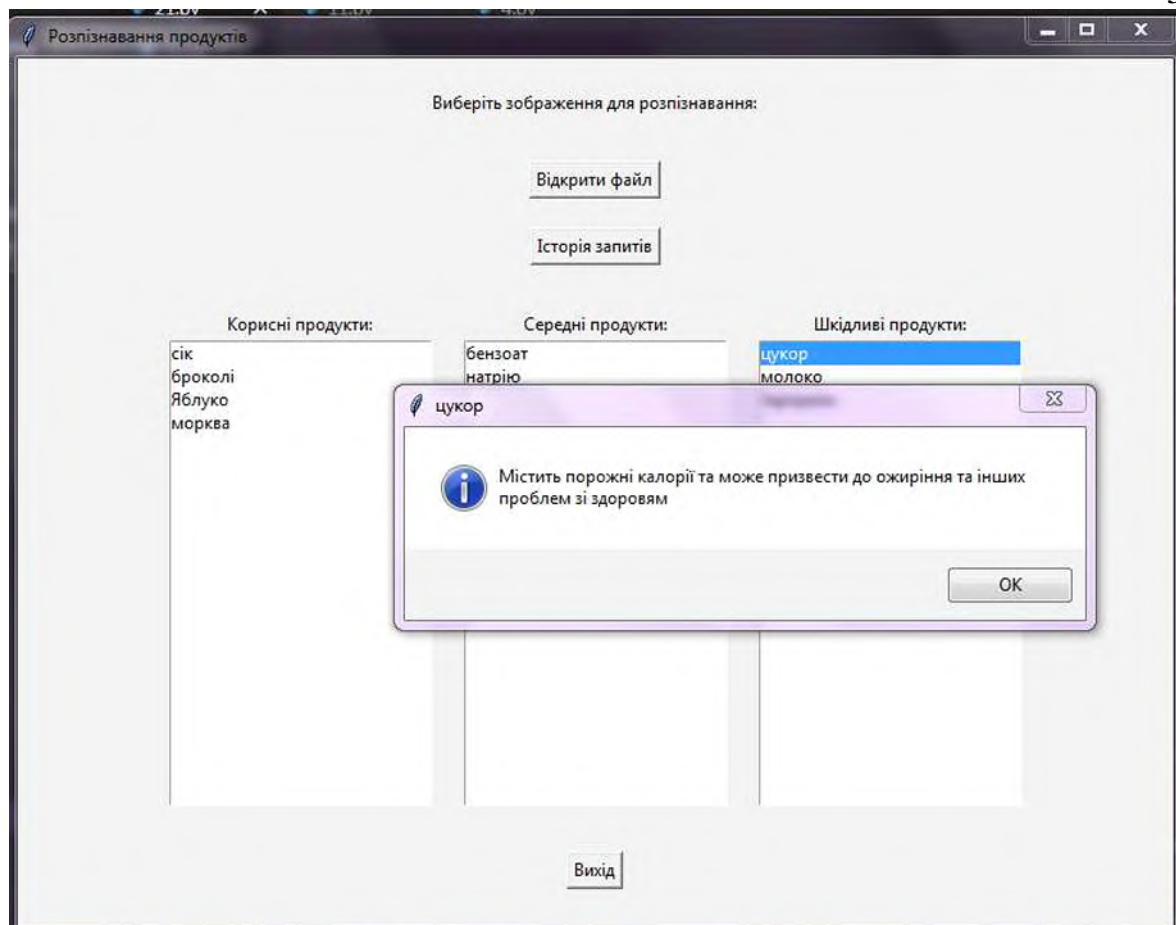


Рисунок 3.5 - Вікно з інформацією про шкідливий продукт

Система зберігає історію запитів користувачів, що дозволяє їм переглядати попередні аналізи продуктів і результати класифікації. Це допомагає користувачам відстежувати їхні вибори щодо дитячого харчування та забезпечує зручний доступ до попередньо зроблених аналізів.

Ця функція дозволяє користувачам зручно переглядати свої попередні запити і отримувати доступ до результатів аналізу продуктів, який був зроблений раніше. Вони можуть переглядати класифікацію продуктів на корисні, середні або погані і зрозуміти, які продукти вони вже аналізували та їхні результати.

Це особливо корисно для тих користувачів, які активно використовують систему для дослідження дитячого харчування та здійснюють багато запитів. Вони можуть легко відстежувати свої вибори та прийняті рішення щодо дитячого харчування, а також забезпечувати зручний доступ до попередньо зроблених аналізів.

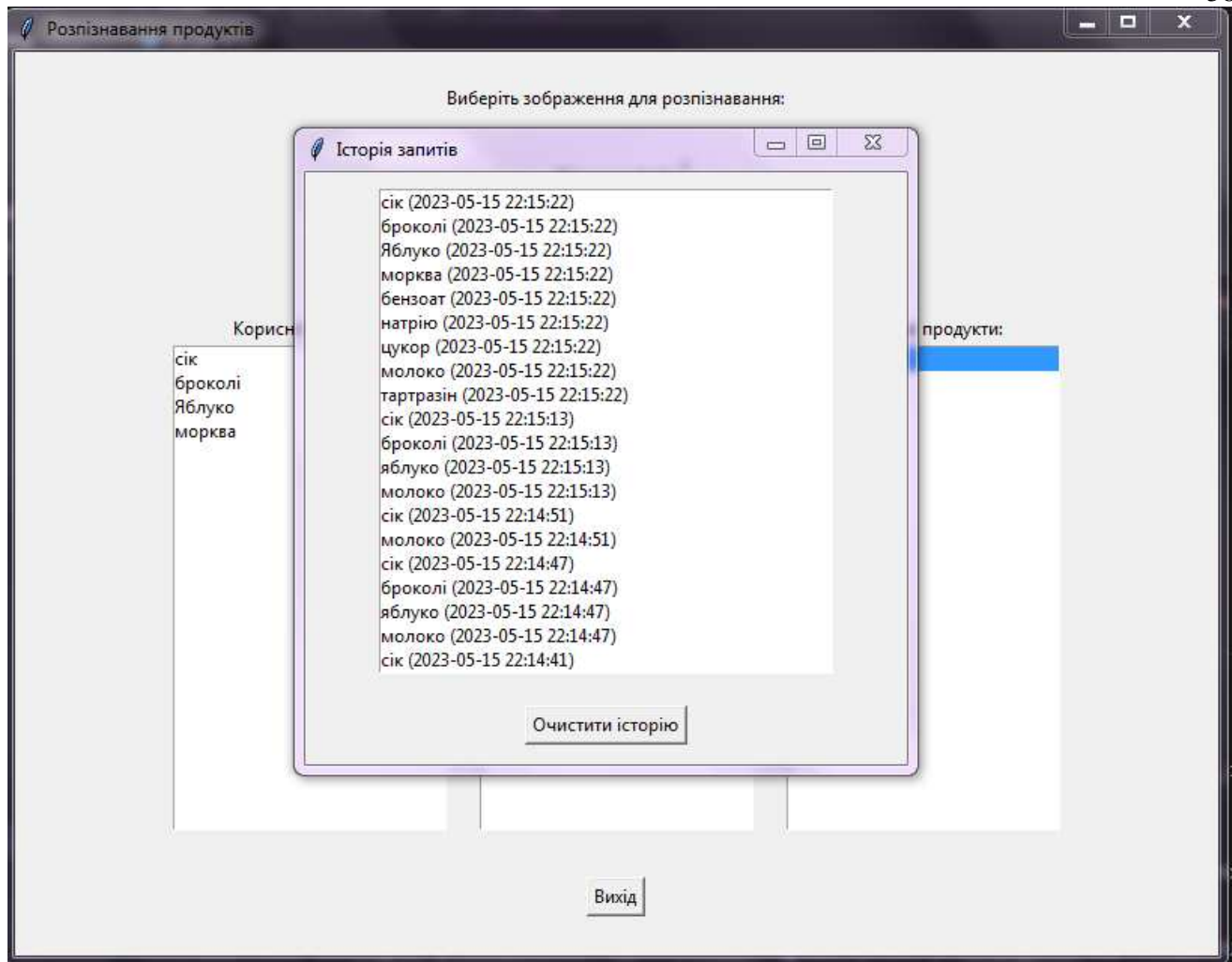


Рисунок 3.6 - Вікно з історією запитів

Крім того, система також надає можливість очищення історії запитів. Це означає, що користувачі мають контроль над своїми даними і можуть видаляти старі запити з історії, які більше не потрібні або не бажані. Ця функція дозволяє підтримувати організованість історії запитів і зберігати лише актуальну інформацію.

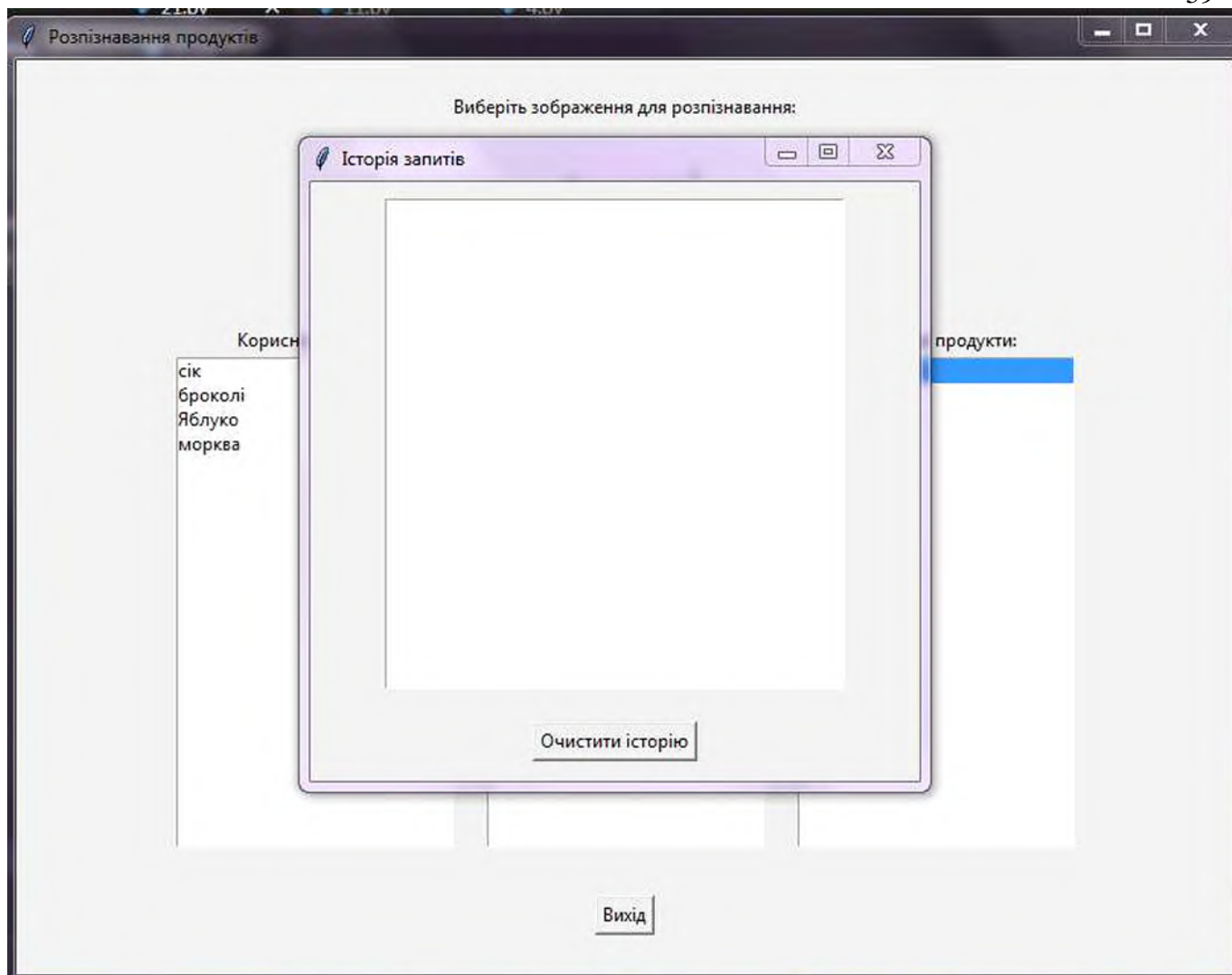


Рисунок 3.7 - Вікно з очищення історії запитів

Інформаційна система "Спосіб для аналізу якості дитячого харчування за його складом на пакуванні нейромережевими засобами" є потужним інструментом для аналізу та оцінки якості дитячого харчування. Вона дозволяє користувачам отримувати достовірну інформацію про склад продуктів, що надрукована на їхній упаковці, за допомогою нейромережових алгоритмів. Система здатна визначати наявність різних компонентів, таких як вітаміни, мінерали, білки, жири, вуглеводи та інші, необхідні для здоров'я дитини.

Одним із ключових аспектів системи є її здатність визначати якість дитячого харчування на основі складу продуктів. За допомогою нейромережових алгоритмів, система може класифікувати продукти на корисні, середні та погані, враховуючи їхній вплив на здоров'я дитини. Це дозволяє батькам та опікунам зробити освічені рішення щодо вибору харчування для своїх дітей.

Результати аналізу та класифікації продуктів система представляє

користувачам у зручній формі. При натисканні на конкретний продукт, відкривається вікно з детальною інформацією про цей продукт, що допомагає зробити обґрунтований вибір.

Додатково, система забезпечує збереження історії запитів користувачів. Це дозволяє переглядати попередні аналізи продуктів та результати класифікації, що сприяє відстеженню власних виборів щодо дитячого харчування.

3.4 Висновки до розділу 3

У розділі 3 була представлена програмна реалізація способу для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі. Були описані основні модулі системи, їх взаємозв'язок та функціональні можливості.

Модуль для взаємодії з базою даних забезпечує збереження та отримання даних про продукти, складові та результати аналізу. Модуль нейронної мережі використовується для аналізу складу продуктів на основі інформації з їх упаковки. Модуль взаємодії з користувачем надає зручний інтерфейс для взаємодії з системою. Модуль обробки зображень здійснює виявлення тексту на упаковці та витягнення інформації про склад продукту.

Система використовує навчену нейронну мережу для класифікації продуктів на корисні, середні або погані з точки зору дитячого харчування. Результати аналізу зберігаються у базі даних, а користувачі можуть взаємодіяти з системою, вводити дані про продукти, переглядати результати аналізу та отримувати інформацію про якість дитячого харчування.

Інформаційна система "Спосіб для аналізу якості дитячого харчування за його складом на пакуванні нейромережевими засобами" представляє собою потужний інструмент для аналізу та оцінки якості дитячого харчування. Завдяки використанню нейромережових алгоритмів, система може швидко та точно аналізувати інформацію на упаковках продуктів та визначати їхній склад з урахуванням різних компонентів, необхідних для здоров'я дитини.

Функціональні можливості системи дозволяють користувачам з легкістю отримувати інформацію про склад дитячого харчування та його якість. Результати

аналізу та класифікації продуктів представлені у зручній формі, що дозволяє користувачам зробити обґрунтовані вибори щодо харчування своєї дитини. Крім того, система забезпечує зберігання історії запитів користувачів, що дозволяє їм відстежувати попередні аналізи та зручно переглядати результати класифікації продуктів.

Висновки

У результаті проведення дослідження був розроблений спосіб для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі. Програмна реалізація способу дозволяє ефективно аналізувати інформацію на упаковках дитячого харчування та визначати їх склад з високою точністю. Застосування нейронної мережі та додаткових компонентів, таких як бібліотеки для роботи з зображеннями та обробки тексту, дозволяє отримати більш повну та точну інформацію про склад дитячого харчування.

Програмна реалізація способу має потенціал для застосування в різних сферах, зокрема у харчовій промисловості, дитячому харчуванні, сфері контролю якості та безпеки харчових продуктів. Завдяки автоматизованому аналізу складу харчування на упаковках, споживачі отримують більш точну інформацію про продукт, що сприяє свідомому вибору та забезпечує безпеку харчування для дітей.

Проте, важливо зазначити, що програмна реалізація способу має свої обмеження. Наприклад, точність аналізу може залежати від доступності та якості даних на упаковці продукту. Також, необхідно постійно оновлювати нейронну мережу та алгоритми обробки даних для підтримки актуальності та точності аналізу.

Загалом, розроблений спосіб та програмна реалізація представляють собою значний прогрес у напрямку автоматизованого аналізу якості дитячого харчування. Впровадження такого інструменту може сприяти поліпшенню безпеки та якості харчування для дітей та споживачів загалом.

Перелік посилань

1. Accuracy, precision и recall [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/companies/ods/articles/328372/>
2. Класифікаційні метрики [Електронний ресурс]. – Режим доступу: http://www.andriystav.cc.ua/Downloads/MITER/Lecture_04.pdf
3. Loginom [Електронний ресурс]. – Режим доступу: <https://wiki.loginom.ru/articles/cross-validation.html>
4. AI powered Food Scanner [Електронний ресурс]. – Режим доступу: <https://www.passio.ai/nutrition-ai>
5. FoodAI [Електронний ресурс]. – Режим доступу: <https://foodai.org>
6. Foodvisor [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=io.foodvisor.foodvisor&hl=ru&gl=US>
7. ResearchGate Logo [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/figure/fig6Internal-Architecture-of-OCR-Tesseract-8-Version-4-has-a-dataset-knowledge-of_fig3_341779791
8. TensorFlow [Електронний ресурс]. – Режим доступу: <https://www.tensorflow.org/guide/basics?hl=ru>
9. Traineddata Files for Version 4.00 + [Електронний ресурс]. – Режим доступу: <https://tesseract-ocr.github.io/tessdoc/Data-Files#data-files-for-version-400-november-29-2016>
10. Харчові добавки [Електронний ресурс]. – Режим доступу: https://texty.org.ua/articles/49129/Harchovi_dobavky_korysni_i_smertelno_nebezpechni_Khimprom-49129/
11. Nutrix [Електронний ресурс]. – Режим доступу: <https://nutrix.ua/mama-i-malysh/detskoe-pitanie/>
12. Codeguida [Електронний ресурс]. – Режим доступу: <https://codeguida.com/post/739>
13. Sqlite3 [Електронний ресурс]. – Режим доступу: <https://docs.python.org/uk/3.9/library/sqlite3.html>
14. Створення вікон та налаштування їх властивостей [Електронний

ресурс]. – Режим доступу: https://kyb-edu.in.ua/?page_id=482

15. Графічний Інтерфейс користувача у мові Python. [Електронний ресурс]. – Режим доступу: <https://vseosvita.ua/library/graficnij-interfejs-koristuvaca-u-movi-python-177645.html>

16. АНАЛІЗ РИНКУ ДИТЯЧОГО ХАРЧУВАННЯ УКРАЇНИ [Електронний ресурс]. – Режим доступу: https://mmi.fem.sumdu.edu.ua/sites/default/files/mmi2010_1_82_89.pdf

17. Dou [Електронний ресурс]. – Режим доступу: <https://dou.ua/lenta/articles/first-steps-in-nlp-tensorflow/>

18. Colab [Електронний ресурс]. – Режим доступу: <https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/guide/basics.ipynb?hl=ru#scrollTo=9EpiOmXFDSS>

19. Img2txt [Електронний ресурс]. – Режим доступу: <https://img2txt.com/uk/file/7af77eade6353e30227f8fd1b50e1598f881ae81018b824cdadc2b40e8db4f56/ocr/38112245>

20. Medium [Електронний ресурс]. – Режим доступу: <https://medium.com/nuances-of-programming/обнаружение-и-извлечение-текста-из-изображения-с-помощью-python-ae178219e8b9>

21. One-Hot Encoding [Електронний ресурс]. – Режим доступу: <https://www.helenkapatsa.ru/bystroie-kodirovaniie/>

22. Bag of words [Електронний ресурс]. – Режим доступу: <https://wiki.loginom.ru/articles/bag-of-words.html>

23. Word2Vec [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/articles/446530/>

24. OpenCV [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/articles/519454/>

25. Termin [Електронний ресурс]. – Режим доступу: <https://termin.in.ua/neyromerzha/>

26. Codecamp [Електронний ресурс]. – Режим доступу: <https://www.codecamp.ru/blog/cross-validation-k-fold/>

27. Loginom [Електронний ресурс]. – Режим доступу: <https://wiki.loginom.ru/articles/cross-validation.html>

ДОДАТКИ

Додаток А

Програмний

КОД

```

import tkinter as tk
import cv2
import pytesseract
import sqlite3

from tkinter import filedialog
from datetime import datetime

# з'єднання з базою даних
pytesseract.pytesseract.tesseract_cmd = 'C:\\Program Files\\Tesseract-
        useful_products]

conn = sqlite3.connect('B\\Products.db')
c = conn.cursor()
        average_products]

# заповнення списків з таблиці бази даних
        bad_products]
c.execute("SELECT name, description FROM useful_products")
useful_products = {row[0].lower(): row[1] for row in c.fetchall()}
c.execute("SELECT name, description FROM average_products")
average_products = {row[0].lower(): row[1] for row in c.fetchall()}
c.execute("SELECT name, description FROM bad_products")
bad_products = {row[0].lower(): row[1] for row in c.fetchall()}

# створення таблиці для зберігання історії запитів
c.execute("""CREATE TABLE IF NOT EXISTS search_history
        (id INTEGER PRIMARY KEY AUTOINCREMENT,
        word TEXT NOT NULL,
        search_date TEXT NOT NULL)""")
conn.commit()

def save_search_history(word):
    # зберегти слово та дату запиту в таблиці search_history
    search_date = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    c.execute("INSERT INTO search_history (word, search_date) VALUES (?,
        )")
    conn.commit()

def browse_file():
    filename = filedialog.askopenfilename()

    # конвертувати зображення у чорно-білий формат
        image = cv2.imread(filename)
        gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        # застосувати оптичне розпізнавання символів
        text = pytesseract.image_to_string(gray_image, lang='ukr')
        # розділити текст на окремі слова
        words = text.split()
        # перевірити, які зі слів співпадають з корисними продуктами
        matched_words1 = [word for word in words if word.lower() in
            useful_products]
        matched_words2 = [word for word in words if word.lower() in
            average_products]
        matched_words3 = [word for word in words if word.lower() in
            bad_products]

        # очищення віджетів Listbox
        listbox1.delete(0, tk.END)
        listbox2.delete(0, tk.END)
        listbox3.delete(0, tk.END)

        # додавання додання співпадаючих слів до віджетів Listbox та
        for word in matched_words1:
            listbox1.insert(tk.END, word)
            save_search_history(word)
        for word in matched_words2:
            listbox2.insert(tk.END, word)
            save_search_history(word)
        for word in matched_words3:
            listbox3.insert(tk.END, word)
            save_search_history(word)

def show_product_info(listbox):
    selection = listbox.curselection()
    if selection:
        # отримати вибране слово
        word = listbox.get(selection[0])
        # перевірити в якому списку знаходиться слово

```

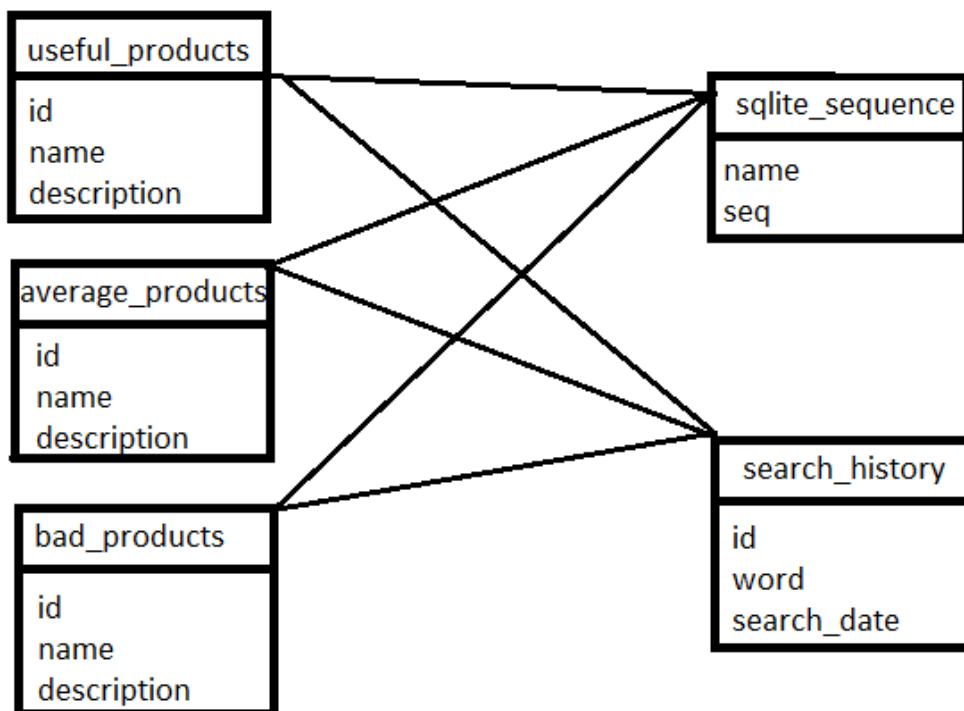
```

if word.lower() in useful_products:
    description = useful_products[word.lower()]
elif word.lower() in average_products:
    description = average_products[word.lower()]
elif word.lower() in bad_products:
    description = bad_products[word.lower()]
# вивести ім'я і опис продукту у вікні messagebox
tk.messagebox.showinfo(title=word, message=description)
def exit_program():
    window.destroy()
def show_history():
    # створення вікна
    history_window = tk.Toplevel(window)
    history_window.geometry("400x400")
    history_window.title("Історія запитів")
    # створення віджету Listbox для відображення запитів
    history_listbox = tk.Listbox(history_window, width=50, height=20)
    history_listbox.pack(pady=10)
    # запит до бази даних для отримання історії запитів
    c.execute("SELECT word, search_date FROM search_history ORDER BY")
    rows = c.fetchall()
    # виведення історії запитів у віджет Listbox
    for row in rows:
        history_listbox.insert(tk.END, f"{row[0]} ({row[1]}")
    # створення кнопки для очищення історії
    clear_button = tk.Button(history_window, text="Очистити історію",
command=lambda: clear_history(history_listbox))
clear_button.pack(pady=10)
def clear_history(listbox):
    show_product_info(listbox1)
    # видалення всіх записів з таблиці search_history
    c.execute("DELETE FROM search_history")
    conn.commit()
    show_product_info(listbox2)
    # очищення віджету Listbox
    listbox.delete(0, tk.END)
# Створення вікна
window = tk.Tk()
window.geometry("800x600")
window.title("Розпізнавання продуктів")
# створення віджетів Label та Listbox
label = tk.Label(window, text="Виберіть зображення для розпізнавання:")
label.pack(pady=20)
button = tk.Button(window, text="Відкрити файл", command=browse_file)
button.pack(pady=10)
history_button = tk.Button(window, text="Історія запитів",
history_button.pack(pady=10)
# Створення фрейму для віджетів Listbox
frame = tk.Frame(window)
frame.pack(pady=20)
label1 = tk.Label(frame, text="Корисні продукти:")
label1.grid(row=0, column=0, padx=10)
label2 = tk.Label(frame, text="Середні продукти:")
label2.grid(row=0, column=1, padx=10)
label3 = tk.Label(frame, text="Шкідливі продукти:")
label3.grid(row=0, column=2, padx=10)
listbox1 = tk.Listbox(frame, width=30, height=20)
listbox1.grid(row=1, column=0, padx=10)
listbox2 = tk.Listbox(frame, width=30, height=20)
listbox2.grid(row=1, column=1, padx=10)
listbox3 = tk.Listbox(frame, width=30, height=20)
listbox3.grid(row=1, column=2, padx=10)
button = tk.Button(window, text="Вихід", command=exit_program)
button.pack(pady=10)
# додавання функції show_product_info до події вибору елемента в
listbox1.bind('<<ListboxSelect>>', lambda event:
listbox2.bind('<<ListboxSelect>>', lambda event:
listbox3.bind('<<ListboxSelect>>', lambda event:
# Запуск GUI
window.mainloop()

```

Додаток Б

База даних



Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 7%

ID: 114501 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА Додано в БД: 2023-06-01 Автора: М.С. Шгойко Керівники: І.В. Драч Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	85585	1264	912 (1%)	14 (1%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми



Ім'я користувача:
Кафедра КН

ID перевірки:
1015376511

Дата перевірки:
01.06.2023 20:01:00 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
01.06.2023 20:07:16 EEST

ID користувача:
100005671

Назва документа: КН-19-1 Штойко

Кількість сторінок: 61 Кількість слів: 13103 Кількість символів: 99692 Розмір файлу: 1.31 MB ID файлу: 1015042198

3.03% Схожість

Найбільша схожість: 1.04% з джерелом з Бібліотеки (ID файлу: 1014994504)

2.47% Джерела з Інтернету

356

Сторінка 63

1.66% Джерела з Бібліотеки

96

Сторінка 65

0.5% Цитат

Цитати

3

Сторінка 66

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Квантовий метод опорних векторів: розробка варіаційних алгоритмів для реалізації класифікації даних у технічних, природничих і соціально-економічних системах

Автор: студент групи КН-19-1 Штойко Микола Сергійович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.т.н., доцент кафедри КН І.В. Драч

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	<i>відповідає</i>
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі Штойка М.С., не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти поширених конструкцій мови та загальновідомі терміни та скорочення.

Обсяг запозичень, визначений системами виявлення збігів/ ідентичності/схожості, складає:

- за системою Anti-Plagiarism: 1%;

- за системою Unichек: 3,03%.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 1% і 3,03% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру дослідження і свідчить на користь кваліфікаційної роботи. Таким чином, запозичення є допустимими та відносяться до описаних вище і адресуються до першоджерел, що, з урахуванням наведених обґрунтувань, свідчить на користь кваліфікаційної роботи.

Керівник роботи

І. Драч

Ілона ДРАЧ

Гарант ОП

Александр Мазурець

Олександр МАЗУРЕЦЬ

Завідувач кафедри КН

Олександр Бармак

Олександр БАРМАК



**ВІДГУК НАУКОВОГО КЕРІВНИКА
на кваліфікаційну роботу бакалавра**

студента гр. КН-19-1 Штойка Миколи Сергійовича
за темою: Спосіб для аналізу якості дитячого харчування за його складом на пакуванні нейромережесевими засобами

1. Актуальність теми

Якість дитячого харчування є важливим аспектом здоров'я та розвитку дітей – майбутнього нації. Відповідальність за якісне харчування дитини несуть, як правило, батьки. Застосування нейромережесевих засобів для аналізу складу дитячого харчування на пакуванні є інноваційним підходом до вирішення цієї проблеми.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

Робота містить інформаційну модель реального процесу – аналізу якості харчових продуктів, яка реалізована із застосуванням нейромережесевих засобів, що відповідає предметній області Стандарту спеціальності 122 Комп'ютерні науки.

3. Професійні та особистісні якості бакалавра

Студент Штойко Микола Сергійович виявив високі професійні та особистісні якості під час виконання роботи. Він проявив зацікавленість та самостійність у дослідженні та виконанні поставлених завдань.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Студент виявив значний рівень самостійності під час виконання кваліфікаційної роботи. Він самостійно вивчив необхідну теоретичну базу, провів дослідження, розробив модель та спосіб для аналізу складу дитячого харчування на пакуванні.

5. Ступінь оволодіння методами дослідження

Студент зібрав та обробив дані, провів необхідні експерименти, застосував нейромережесеві засоби для отримання результатів. Під час виконання кваліфікаційної роботи проявив компетентність в роботі з даними та використанні відповідних аналітичних методів.

6. Повнота та якість розкриття теми роботи

Студент детально розкрив тему роботи і розробив необхідні аспекти аналізу якості дитячого харчування на пакуванні за його складом. Викладений матеріал є повним, зрозумілим та добре структурованим. Застосування нейромережесевих засобів для аналізу дозволило отримати достовірні результати та зробити висновки.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Робота має логічну структуру та послідовне викладення матеріалу. Аргументи та висновки, які були зроблені, підтримані науковими даними та обґрунтовані. Викладений матеріал є літературно грамотним та зрозумілим для читача.

8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Кваліфікаційна робота має практичну цінність, оскільки розроблено спосіб для аналізу якості дитячого харчування на основі його складу на пакуванні з використанням нейромережесвих засобів. Цей спосіб може бути застосований для індивідуального використання у вигляді додатку, на торгівельних підприємствах для контролю якості харчових продуктів, в лабораторіях якості та безпеки харчової продукції для оцінки якості дитячих харчових продуктів.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи достатній рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «добре».

Керівник _____

Г. Дрець

к.т.н., доц., доцент. каф. КН Ілона ДРАЧ



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-19-1 Штойка Миколи Сергійовича
за темою: Спосіб для аналізу якості дитячого харчування за його складом на пакуванні нейромережжевими засобами

1. Актуальність обраної теми

Актуальність обраної теми належно обґрунтована в роботі. Проблема якості дитячого харчування є важливою сьогодні, і використання нейромережжєвих засобів для аналізу складу харчових продуктів на їх пакуванні є цікавим підходом до розв'язання цієї проблеми.

2. Повнота розкриття мети та завдань роботи

Метою кваліфікаційної роботи автор зазначив розробку способу для аналізу якості дитячого харчування за його складом на пакуванні з використанням нейронної мережі. Автор чітко визначив кроки досягнення цієї мети. Мета та завдання роботи ретельно та повно розкриті.

3. Зміст кожного розділу роботи

Зміст кожного розділу роботи є послідовним і логічно побудованим. Автор систематично розглядає важливі аспекти своєї роботи, включаючи огляд літератури, опис методології, деталі аналізу та результати, а також обговорення отриманих даних.

4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблена інформаційна система для аналізу якості дитячого харчування за його складом на пакуванні засобами нейромережі отримала високу оцінку. Автор приводить докази ефективності та точності системи, а також описує її можливі практичні застосування.

5. Якість оформлення кваліфікаційної роботи бакалавра

Робота відповідає вимогам науково-технічного стилю написання і містить всі необхідні розділи (вступ, огляд літератури, методологія, результати, висновки). Текст структурований та чітко поділений на підрозділи, що полегшує сприйняття матеріалу. Використані джерела належно процитовані і включені до списку використаної літератури згідно з встановленими стандартами.

6. Недоліки кваліфікаційної роботи бакалавра

У кваліфікаційній роботі бакалавра також присутні деякі недоліки, які варто відзначити. По-перше, більш докладні пояснення можуть бути введені в розділі огляду літератури для більш глибокого розуміння попередніх досліджень у цій галузі. Крім того, бажано більше практичних прикладів та досліджень для підтвердження ефективності розробленого методу.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи достатній рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «добре».

Рецензент

Мерцинюк В. В.