

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Штучного інтелекту  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розробка інформаційної системи для аналізу складу продуктів для людей  
з особливим типом харчування  
(тема)

Виконав:  
студент 2 курсу, групи СШМ-20-1  
Яричкіна Т.С.  
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту  
(повна назва спеціалізації)

Керівник доц. Чала Л.Е.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

В.О. Філатов  
(прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)  
Кафедра Штучного інтелекту  
(повна назва)  
Рівень вищої освіти другий (магістерський)  
Спеціальність 122 Комп'ютерні науки  
(код і повна назва)  
Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)  
Освітня програма Системи штучного інтелекту (СШІ)  
(повна назва)

ЗАТВЕРДЖУЮ:  
Зав. кафедри \_\_\_\_\_  
(підпис)  
«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Яричкіній Тетяні Сергіївні  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інформаційної системи для аналізу складу продуктів для людей з особливим типом харчування

затверджена наказом університету від 8 листопада 2021 р. № 1695Ст

2. Термін подання студентом роботи до екзаменаційної комісії 8 грудня 2021 р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проєктів щодо розробки та дослідження тематичного моделювання корпусів текстів, Python documentation

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1) Аналіз предметної галузі та постановка задачі

2) Штучні нейронні мережі

3) Технології розпізнавання тексту на зображеннях

4) Експериментальне моделювання та навчання моделі

5) Аналіз складу продуктів харчування

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1.1 – Шкідливі добавки у складі продуктів харчування, Рисунок 1.2 – Глікемічний індекс різних продуктів, Рисунок 1.3 – Основні алергени, Рисунок 2.1 – Приклад штрих-коду та розшифрування його цифр, Рисунок 2.2 – Коди країн-виробників, Рисунок 2.3 – Приклад QR-коду, Рисунок 2.4 – Схема будови перцептронів, Рисунок 2.5 – Схема побудови НМ Хопфілда, Рисунок 2.6 – Схема побудови НМ Машина Больцмана, Рисунок 2.7 – Схема побудови ЗНМ, Рисунок 2.8 – Співвідношення карт ознак, Рисунок 2.9 – Приклад ядра з навченою ознакою, Рисунок 2.10 – Опис згортки, Рисунок 2.11 – Демонстрація роботи алгоритму пошуку контуру, Рисунок 2.12 – Візуалізація алгоритму, Рисунок 4.1 – Схема БД, Рисунок 4.2 – Таблиця категорій, Рисунок 4.3 – Таблиця статусів, Рисунок 4.4 – Таблиця країн, Рисунок 4.5 – Таблиця добавок, Рисунок 4.6 – Таблиця компонентів складу, Рисунок 4.7 – Таблиця виробників, Рисунок 4.8 – Таблиця товарів, Рисунок 4.9 – Таблиця складу, Рисунок 4.19 – Відповідь виробника щодо ароматизатора, Рисунок 4.28 – Продукт для тестування «Чорний шоколад «Корона»», Рисунок 4.29 – Фото складу

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	Доц. каф. ШІ Чала Л.Е.		

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на дипломну роботу	01.09.2021	виконано
2	Аналіз предметної області	02.09.2021 – 07.09.2021	виконано
3	Формування постановки задачі	08.09.2021 – 21.09.2021	виконано
4	Аналіз конкурентів	21.09.2021 – 25.09.2021	виконано
5	Дослідження алгоритмів розпізнавання тексту	26.09.2021 – 01.10.2021	виконано
6	Формування алгоритму для програмної реалізації	01.10.2021 – 07.10.2021	виконано
7	Реалізація програми	08.10.2021 – 10.10.2021	виконано
8	Обробка результатів роботи програми	11.10.2021 – 12.10.2021	виконано
9	Підготовка матеріалів для пояснювальної записки	13.10.2021 – 23.11.2021	виконано
10	Попередній захист	01.12.2021	виконано
11	Захист перед ЕК	08.12.2021	

Дата видачі завдання 1 вересня 2021 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис) \_\_\_\_\_  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Записка пояснювальна: 71 с., 52 рис., 1 дод., 26 джерел.

БАЗА ДАНИХ, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, КОМПОНЕНТИ, МАШИННЕ НАВЧАННЯ, НЕЙРОННА МЕРЕЖА, РОЗПІЗНАВАННЯ ТЕКСТА, СКЛАД ПРОДУКТІВ ЖИВЛЕННЯ, ШТРИХ-КОД

Об'єкт дослідження – системи інтелектуального аналізу складу продуктів харчування.

Предмет дослідження – методи розпізнавання тексту на зображеннях та фото.

Мета роботи – визначення за зображеннями та знімками, або за штрих-кодом, складу продукту харчування та опису окремих компонентів за допомогою інтелектуальної системи.

Методи дослідження – машинне навчання, нейромереві методи класифікації.

Дана кваліфікаційна робота присвячена проблемі непрозорості складу продуктів харчування українського ринку. Вона покликана спростити визначення складу продуктів для людей, які потребують цього з різних причин: хвороба, етичні уподобання тощо.

## РЕФЕРАТ

Пояснительная записка: 71 с., 52 рис., 1 прил., 26 источников.

БАЗА ДАННЫХ, КЛАССИФИКАЦИЯ ИЗОБРАЖЕНИЙ, КОМПОНЕНТЫ, МАШИННОЕ ОБУЧЕНИЕ, НЕЙРОННАЯ СЕТЬ, РАСПОЗНАВАНИЕ ТЕКСТА, СОСТАВ ПРОДУКТОВ ПИТАНИЯ, ШТРИХ-КОД

Объект исследования – системы интеллектуального анализа состава продуктов питания.

Предмет исследования – методы распознавания текста на изображениях и фото.

Цель работы – определение по изображениям, снимкам или штрих-коду, составу продукта питания описание отдельных компонентов с помощью интеллектуальной системы.

Методы исследования – машинное обучение, нейросетевые методы классификации.

Данная квалификационная работа посвящена проблеме непрозрачности состава продуктов питания украинского рынка. Она призвана упростить определение состава продуктов для людей, нуждающихся в этом по разным причинам: болезнь, этические предпочтения и т.д.

## **ABSTRACT**

Explanatory note: 71 p., 52 fig., 1 ann., 26 sources.

**BARCODE, COMPONENTS, DATABASE, FOOD COMPOSITION,  
IMAGE CLASSIFICATION, MACHINE LEARNING, NEURAL NETWORK,  
TEXT RECOGNITION**

The object of the study – the systems of intellectual analysis of the composition of food products.

The subject of research is methods of text recognition in images and photos.

The purpose of the work is to determine the description of individual components from images, photographs or a barcode, the composition of a food product using an intelligent system.

Research methods – machine learning, neural network classification methods.

This certification work is devoted to the problem of the opacity of the composition of food products on the Ukrainian market. This work is designed to simplify the reading of composition for people who need it for reasons such as illness, ethical preferences and desire to understand what a person eats.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	9
Вступ.....	10
1 Аналіз предметної області.....	11
1.1 Що таке склад продуктів харчування.....	11
1.2 Кодові та хімічні позначення у складі продуктів харчування .....	13
1.3 Хвороби, за наявності яких важливий аналіз складу продуктів.....	16
1.3.1 Цукровий діабет .....	16
1.3.2 Алергія.....	18
1.3.3 Непереносимість лактози .....	19
1.4 Склад продуктів для веганів .....	19
1.5 Склад не харчових продуктів .....	20
1.6 Постановка задачі .....	21
1.7 Існуючі аналоги.....	22
2 Системи маркування товарів та методи розпізнавання тексту .....	24
2.1 Проблема маркування продуктів харчування.....	24
2.1.1 Що таке штрих-код та його недоліки.....	24
2.1.2 Можливість заміни штрих-коду на QR-код у майбутньому.....	26
2.2 Нейронна мережа та її типи .....	29
2.3 Опис алгоритму розпізнавання тексту .....	35
3 Вибір засобів реалізації .....	38
3.1 Вибір мови програмування.....	38
3.2 Вибір середовища та інструментів.....	39
3.2.1 Вибір СКБД.....	39
3.2.2 Вибір середовища для Python .....	40
4 Практична реалізація .....	42
4.1 Створення БД .....	42
4.1.1 Побудова схеми БД.....	42
4.1.2 Створення БД за складеною схемою.....	48

4.2 Приклад заповнення БД .....	52
4.4 Складання фінальної версії проекту .....	56
4.5 Тестування роботи проекту .....	61
Висновки .....	65
Перелік джерел посилання .....	67
Додаток А Відомість кваліфікаційної роботи магістра.....	71

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

– БД – база даних – це сукупність даних, що зберігаються відповідно до схеми даних, маніпулювання якими виконується відповідно до правил засобів моделювання даних;

– ЗНМ – згортоква нейронна мережа – спеціальна архітектура штучних нейронних мереж, запропонована Яном Лекуном в 1988 році і націлена на ефективне розпізнавання образів, входить до складу технологій глибокого навчання;

– НМ – нейронна мережа – це обчислювальні системи, натхнені біологічними нейронними мережами, що складають мозок тварин;

– Python – високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника і читання коду;

– Tesseract – це вільна програма для розпізнавання текстів з різних типів зображень.

## ВСТУП

Склад продуктів – це перелік інгредієнтів, який наводять всім харчовим продуктам, крім тих, які складаються з одного інгредієнта. Перед списком інгредієнтів має бути заголовок «Склад». Інгредієнти перераховують як зменшення масової частки на момент виготовлення харчового продукту.

Однак хоч наявність складу є обов'язковою, часто людина, яка читає його, зазнає труднощів або з його розумінням, або з його прочитанням, оскільки шрифт є надто дрібним для тих, хто має проблеми із зором. Труднощі з розшифровкою виникають у ситуаціях, коли інгредієнт або описаний хімічно, або зашифрований як E-код.

E-код означає, що добавка пройшла відповідну процедуру оцінки безпеки та схвалена до використання в ЄС, однак є люди, для яких ці добавки можуть бути шкідливими. Причиною для того, щоб звернути увагу на такий інгредієнт, може бути хвороба, алергія, ідеологічні причини та багато іншого, але часто покупець не може прямо на місці зрозуміти, що саме зашифровано у складі і чи може він вживати цей продукт.

Якщо повернутись до проблеми прочитання складу через маленький шрифт або незручне розташування, то можна помітити ще одну актуальну проблему: штрих-код, яким маркуються всі товари, не містить інформації про склад продукту. Сам по собі штрих-код – це застаріла система маркування. Після друку він не може більше доповнюватися, наприклад, складом продуктів, і зберігає не більше 50 байт інформації. У сучасному світі часто говорять про те, що час замінити штрих-код на QR-код, який може містити в рази більше інформації. Максимальний розмір QR-коду дає можливість зберігати в ньому: або 7089 чисел, або 4269 символів. Цього було б достатньо, щоб зберігати, як мінімум, посилання на повний склад продукту.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Метою даної роботи є розшифровка складу продуктів для людей, які цього потребують. Ця тема є актуальною, тому що існує безліч людей, які внаслідок хвороб змушені дотримуватися особливої дієти. Також все більше людей намагаються вести здоровий спосіб життя і тоді добавки, які містять багато продуктів, мають для них велике значення, оскільки не завжди зрозуміло, що саме виробник додав у свою продукцію. Крім цього, багато хто в наш час переходить на рослинний тип харчування або веганство, при якому людина намагається якнайбільше усунути себе від експлуатації тварин. До цього входять і продукти тваринного походження. Дуже часто, читаючи склад, можна випадково натрапити на зашифроване під чисельний код з літерою E спереду складове, що має тваринне походження. Також багато виробників не вказують, яким чином отримані ароматизатори або з чого складається барвник, доданий у продукт харчування. Причини, чому і для кого важливий склад можна перераховувати дуже довго, але проблемою є те, що не завжди людина може навіть прочитати цей склад, оскільки на упаковці або ціннику він розташований занадто незручно, закритий чимось зверху, зачеплений швом або просто надрукований таким дрібним шрифтом, що навіть людині з гарним зором важко розрізнити, що в ньому написано.

### 1.1 Що таке склад продуктів харчування

Складом продукту називають перелік інгредієнтів, який наводять всім харчовим продуктам, крім тих, які складаються з одного компонента. Перед цим переліком має стояти заголовок «Склад». Інгредієнти йдуть у порядку зменшення масової частки на момент виготовлення харчового продукту.

У разі коли масова частка складового компонента в готовому харчовому продукті становить менше 2%, допускається упустити перелік його інгредієнтів, що входять до складу, за винятком харчових або біологічно активних добавок та речовин, отриманих з генно-інженерно-модифікованих організмів або з їх використанням .

Вода, що входить у продукт, повинна вказуватись у списку інгредієнтів, за винятком тих випадків, коли вона є складовою відновлених продуктів, а також таких інгредієнтів, як розсіл, маринад, сироп, бульйон, тузлук та інших, що згадуються у списку інгредієнтів під власними найменуваннями [1], [2].

У списку інгредієнтів також не вказують:

- двоокис вуглецю (якщо не зазначено, що газований продукт);
- летючі компоненти, які у процесі виготовлення продукту тимчасово виділяються, проте потім повертаються назад, у кількісному відношенні не перевищуючи початковий зміст;
- допоміжні матеріали та речовини, які необхідні для виробництва, але не входять до складу продукту;
- харчові добавки, які містилися в інгредієнтах продукту, якщо в кінцевому продукті вони вже не мають технологічного ефекту;
- речовини, які використовуються як розчинники або носії.

При вказівці харчових добавок використовують групові назви:

- антиокислювачі;
- речовини для обробки борошна;
- речовини, що перешкоджають комкуванню;
- речовини, що сприяють збереженню забарвлення;
- вологоутримуючі агенти;
- глазувальники;
- желеутворювачі;
- загусники;
- кислоти;

- консерванти;
- барвники;
- наповнювачі;
- затверджувачі;
- піногасники;
- піноутворювачі;
- пропеленти;
- підсолоджувачі;
- розпушувачі;
- регулятори;
- стабілізатори;
- ущільнювачі;
- підсилювачі смаку та запаху;
- емульгатори;
- емульгуючі солі.

## 1.2 Кодові та хімічні позначення у складі продуктів харчування

Дуже часто споживач стикається з проблемою нерозуміння, що саме додано до харчового продукту. Найчастіше такими компонентами є числові коди з літерою Е, що стоїть попереду.

Ці коди призначені для класифікації харчових добавок. Така система розроблена у Євросоюзі. Кожна така добавка має унікальний номер, який починається з літери Е. Система була допрацьована та прийнята для міжнародної класифікації «Кодекс Аліментаріус».

Речовини, які позначені цими кодами, іноді можуть нашкодити здоров'ю людини, яка не знає про те, що вона означає.

Усі «Е» поділяються на такі групи:

- барвники (Е100 – Е199);
- консерванти (Е200 – Е299);

- антиокислювачі (E300 – E339);
- стабілізатори, загусники та емульгатори (E400 – E499);
- регулятори кислотності та речовини (E500 – E599);
- підсилювачі смаку та аромату (E600 – E699);
- антибіотики (E700 – E799);
- резерв (E800 – E899);
- інші (E900 – E999);
- додаткові речовини (E1000 – E1999).

Найчастіше харчові добавки використовують для покращення стабільності та більш тривалого зберігання, для збереження харчової цінності продукту, для різноманітних цілей при виробництві, упаковці, обробці та зберіганні [3].

Часто можна побачити в новинах, що різні такі E-добавки викликають ракові пухлини, розлад шлунка, алергію та інші неприємні наслідки.

Однак потрібно враховувати, що ефекти залежать як від особливостей організму конкретної людини, так і кількості добавки, яку він вжив. Для кожної E-добавки є своя допустима добова доза, перевищення якої може призвести до негативних наслідків.

Наприклад, добавка E250 або нітрит натрію застосовують у ковбасах, хоча відомо, що ця речовина отруйна для ссавців. Для щура смертельна доза становить приблизно 180 міліграм на кілограм ваги тіла. Проте його не забороняють і використовують повсюдно, оскільки ця добавка забезпечує продукту товарний вигляд, а отже, його купує більше людей. Щоб це перевірити, достатньо порівняти колір ковбаси на прилавку магазину, де він буде яскравим і насиченим, з домашньою ковбасою, яку можна придбати на ринку, де колір буде темнішим і тьмянішим. Більше того, для копчених ковбас норма вмісту добавки E250 встановлена значно вищою, ніж для варених, оскільки вважається, що їх люди їдять у менших кількостях.

Також, з розвитком аналітичних методів, частина добавок, які раніше вважалися нешкідливими, були визнані надто небезпечними та у наш час заборонені. Прикладом такої добавки може бути E240 (формальдегід), який використовувався раніше в шоколадних батончиках, або E121 – у газованій воді. Також добавки, які нешкідливі для однієї людини, можуть завдати сильного шкідливого впливу на іншу. Лікарі радять людям похилого віку та алергікам бути особливо уважним до складу та змісту в ньому E-добавок. По можливості краще уникати таких продуктів.

Часто виробники використовують не E-код для позначення добавки у складі, а її хімічну назву. Робиться це з метою не відлякувати покупців наявністю E у складі товару [4].

Перелік шкідливих харчових добавок наведено на рисунку 1.1.

ТАБЛИЦЯ ШКІДЛИВИХ ХАРЧОВИХ ДОБАВОК	
<b>ДУЖЕ НЕБЕЗПЕЧНІ</b>	E123 E510 E513 E527
<b>НЕБЕЗПЕЧНІ</b>	E102 E110 E120 E124 E127 E129 E155 E180 E201 E220 E222 E223 E224 E228 E233 E242 E400 E401 E402 E403 E404 E405 E501 E502 E503 E620 E636 E637
<b>КАНЦЕРОГЕННІ</b>	E131 E142 E153 E210 E212 E213 E214 E215 E216 E219 E230 E240 E249 E280 E281 E282 E283 E310 E954
<b>ШЛУНКОВІ ЗАХВОРЮВАННЯ</b>	E338 E339 E340 E341 E343 E450 E461 E462 E463 E465 E466
<b>ШКІРНІ ЗАХВОРЮВАННЯ</b>	E151 E160 E231 E232 E239 E311 E312 E320 E907 E951 E1105
<b>РОЗЛАДИ КИШЕЧНИКА</b>	E154 E626 E627 E628 E629 E630 E631 E632 E633 E634 E635
<b>ТИСК</b>	E154 E250 E252
<b>НЕБЕЗПЕЧНІ ДЛЯ ДІТЕЙ</b>	E270
<b>ЗАБОРОНЕНІ</b>	E103 E105 E111 E121 E123 E125 E126 E130 E152 E211 E952
<b>ПІДОЗРІЛІ</b>	E104 E122 E141 E171 E173 E241 E477

Рисунок 1.1 – Шкідливі добавки у складі продуктів харчування

### 1.3 Хвороби, за наявності яких важливий аналіз складу продуктів

Існує багато захворювань, за яких потрібно уважно відстежувати склад продуктів. У цьому розділі розглянуто деяка з них, щоб показати, наскільки важливою для деяких людей є прозорість цього питання. Часто від цього залежить самопочуття людини та збереження її життя [5].

#### 1.3.1 Цукровий діабет

Першим захворюванням, на яке хотілося б звернути увагу, є цукровий діабет.

Цукровий діабет – це хронічне захворювання ендокринної системи, спричинене порушенням синтезу інсуліну та підвищення цукру в крові. Захворювання може призвести до ряду серйозних патологій [9].

Дане захворювання відноситься до тих, при яких слід суворо дотримуватись певної дієти. Продукти діабетичного харчування не повинні містити вуглеводів, що швидко засвоюються, тому що їх споживання може стати причиною різкого стрибка рівня цукру в крові. Раціон харчування повинен складатися переважно з їжі з високим вмістом клітковини, мінеральних речовин та вітамінів. Для цього чудово підходять сезонні фрукти, крупи, овочі та молочні продукти.

Цукровий діабет буває двох типів і залежно від діагнозу до харчування ставляться різні обмеження.

При діабеті першого типу цукор у крові підвищується за рахунок недостатнього вироблення інсуліну підшлунковою залозою. Людям з таким типом захворювання рекомендується дотримуватись основ здорового харчування та строгі обмеження харчування не ставляться.

При діабеті другого типу глюкоза практично не засвоюється і внаслідок цього, потрапляючи до організму, вона концентрується у крові, накопичуючись. Людям з таким типом захворювання радять

дотримуватися жорсткої низькокалорійної дієти та протипоказано вживання простих цукрів [6].

На сьогоднішній день для людей з цукровим діабетом випускаються спеціальні продукти харчування, але з ними все не так просто, як здається на перший погляд і їх все одно варто перевіряти на наявність у складі заборонених компонентів.

Насамперед люди з цим захворюванням повинні дивитися на глікемічний індекс продукту. Він показує, як швидко підвищується рівень цукру на крові після вживання вуглеводних продуктів. Для діабетиків рекомендовано продукти, у яких він якнайнижчий.

Рівень глікемічного індексу різних продуктів можна побачити на рисунку 1.2.



Снэки	Г.И.	Крахмалистые продукты	Г.И.	Овощи	Г.И.	Фрукты	Г.И.	Молокопродукты	Г.И.
Пицца	33	Свежая выпечка из белой муки	33	Брокколи	10	Вишни	22	Белый йогурт (без добавок)	14
Плитка шоколада	49	Белый рис	38	Перец	10	Яблоки	38	Обезжиренный йогурт	14
Фунтовый кекс	54	Макарони из рафинированной муки	38	Салат	10	Апельсины	43	Молоко	30
Попкорн	55	Картофель	44	Грибы	10	Виноград	46	Соевое молоко	31
Энергетический батончик	58	Белый хлеб	49	Огурцы	10	Киви	52	Кефир, Ряженка	49
Сладкая газировка	72	Коричневый рис	55	Зеленый горошек	48	Бананы	56	Какао	32
Пончик	76	Оладьи и блинчики	67	Морковь	49	Ананасы	66	Йогурт с наполнителям	36
Желейные конфетки	80	Ржаной хлеб	80	Свекла	64	Дыня	72	Заварной крем	43
Соломка, крекеры	83	Запеченный картофель	85	Лук	75	Финики	103	Мороженное	60

Рисунок 1.2 – Глікемічний індекс різних продуктів

Однак товари, які нібито призначені для людей з цукровим діабетом, часто містять компоненти з дуже високим глікемічним індексом. Основними з них, на які виробники заплющують очі, є крохмаль, біле борошно пшеничне, маргарин і пальмове масло. Це лише кілька з небезпечних для діабетиків складових, насправді їх набагато більше, включаючи деякі замінники цукру [7].

### 1.3.2 Алергія

Це не таке явне захворювання, як цукровий діабет, проте зустрічається не менш часто.

Алергією називають хронічне захворювання, спричинене неадекватною негативною реакцією імунної системи на вплив речовин, які в основному не призводять до захворювань та не завдають шкоди людині. Алергія може бути різною, але вона часто буває саме на продукти харчування.

Деякі, в основному зарубіжні компанії виділяють у складі своїх продуктів алергени жирним шрифтом, щоб люди, для яких це важливо, могли швидко виявити небажаний компонент і відмовитися від покупки продукту. Проте є компоненти, які неявно містять алергени і на які людина з алергією може не звернути увагу. Такі ситуації можуть бути небезпечними для життя людей, тому для них читати склад продуктів також дуже важливо [8].

Нижче представлені основні харчові алергени (рисунок 1.3).



Рисунок 1.3 – Основні алергени

### 1.3.3 Непереносимість лактози

Непереносимість лактози або лактозна недостатність – це стан, коли організм людини не може вільно перетравлювати лактозу та цукор, які містять молоко та молочні продукти.

У сучасному світі це досить часте захворювання і люди, схильні до нього, повинні бути дуже уважні, вибираючи продукцію в магазинах, тому, що навіть деякі товари, що мають позначку «без лактози», все ж таки містять її, наражаючи на небезпеку потенційних покупців.

Люди з непереносимістю лактози часто взагалі можуть вживати молочні продукти. Від цього в них погіршується як загальний стан, так й стан мікрофлори кишечника. Симптоми можуть бути різними: здуття, біль у животі чи коліки, бурчання в животі, надмірне газоутворення, діарея, нудота та багато іншого.

Крім того, що спеціалізованим товарам не завжди можна довіряти, лактоза може бути там, де людина її не чекає. Це може бути хліб, зернові сніданки, батончики мюслі, маргарин, а також багато ліків, такі як протизапальні або для лікування печії [10].

### 1.4 Склад продуктів для веганів

Для початку варто сказати, що веганство – це не дієта чи якесь особливе меню. Люди, які дотримуються такого способу життя, не вживають у їжу продукти тваринного походження і ті, які містять їх у собі.

Основними компонентами харчування таких людей є бобові, овочі, злаки, гриби, горіхи, насіння та фрукти. Також зараз дуже багато веганських альтернатив, таких як ковбаси, пельмені, котлети, морозиво, йогурти та багато іншого. Проте, навіть такі продукти харчування завжди необхідно перевіряти на наявність у складі інгредієнтів, що мають тваринне походження. Так відбувається тому, що іноді компанії, що

випускають такого роду продукцію, не до кінця розуміється на питанні веганства, і як наслідок, можуть додавати у свою продукцію те, чого там бути не повинно. Яскравим прикладом такого продукту може бути натуральний мед, який є продуктом експлуатації людиною бджіл.

Крім цього є й зворотні ситуації, коли продукти, які на перший погляд не веганські, є такими, оскільки часто у виробництві дешевше та простіше використовувати рослинні компоненти.

Для того щоб визначати харчову продукцію, яку можна чи не можна вживати, веганам потрібно уважно читати склад, розуміючи іноді дуже непрозорі позначення добавок [11].

### 1.5 Склад не харчових продуктів

Крім харчових продуктів, які явно потрапляють до організму людини, варто звертати увагу на склад тих речей, які роблять це не явно.

Хорошим прикладом може бути косметика. Багато людей використовують її щодня, не замислюючись наносячи такі засоби на свою шкіру чи волосся. Часто косметика може містити в собі шкідливі речовини або алергени, а ті ж самі шампуні – силікати, які негативно впливають на волосся. Це можуть бути також консерванти або ненатуральні барвники. Так само не бажаним компонентом косметики може бути спирт, оскільки він, крім антисептичного ефекту, може викликати роздратування. Крім того, він низькомолекулярний, тобто легко проникає крізь шкіру разом із хімічними речовинами, які можуть бути як корисними, так і шкідливими.

У наш час є багато засобів, що дозволяють перевіряти склад косметичних засобів, проте мало хто користується цим [12].

## 1.6 Постановка задачі

Метою даної кваліфікаційної роботи є розробка інформаційної системи, яка дозволить зробити склад прозорим для людей, які можуть цього потребувати через особливості свого харчування. Причиною цього може бути хвороба, ідеологічні мотиви, чи просто бажання виключити зі свого раціону конкретні продукти.

В організації отримання інформації про склад продуктів було виділено основні завдання:

- створення та заповнення бази даних з потенційно небезпечними компонентами для різних категорій споживача;
- створення системи, що отримує на вхід знімок складу з упаковки товару та віддає на виході перелік компонентів, що входять до цього списку;
- за допомогою системи, створеної на першому кроці, створити та заповнити БД продуктами зі складом, порівнянно їх штрих-коду;
- створити зручний та оптимізований пошук по товарах;
- додати можливість додавати раніше не додані до БД товари за допомогою системи з першого кроку та сканера штрих-коду;
- виділяти у виведеному складі продуктів компоненти, які можуть бути небажаними для споживачів різних категорій.

Таким чином, буде створено інтелектуальну систему, яку можна буде використовувати для створення різних типів додатків. Кінцевими користувачами будуть люди з особливим типом харчування, які мають труднощі з читанням складу з якихось причин. Це можуть бути як проблеми із зором, так і незнання розшифровок добавок та хімічних назв у складі продуктів.

## 1.7 Існуючі аналоги

На даний момент у світі існує багато додатків, які мають широкий функціонал і величезні довідники даних, які містять у собі розширену інформацію про склад продуктів.

Найвідоміші з них:

- Ecolabel Guide;
- Натурометр;
- Е код – харчові добавки;
- Open Food Facts;
- Fooducate Healthy Weight Loss & Calorie Counter.

Кожен із цих додатків містить у собі як можливість отримання інформації про складі продуктів, так й має зручні додаткові функції для користувача. Наприклад, Fooducate Healthy Weight Loss & Calorie Counter також допомагає налагодити систему правильного харчування і збалансувати вагу, а Open Food Facts містить інформацію про те, як утилізувати упаковку продукту [16].

Всі вони виглядають зручними, і багато людей використовують такі додатки у своєму повсякденному житті, проте для українського споживача є декілька недоліків.

Першим недоліком є те, що жодна з цих програм не підтримує українську мову.

Другий недолік впливає з першого – такі додатки не містять товарів, які виробляються в Україні чи лише для України. Це пов'язано з тим, що вони орієнтовані більше на російський чи американський ринок. Хоча розробники і дають можливість відправити їм фото складу продукту та його самого, щоб у майбутньому додати його до своєї бази, проте це надто трудомісткий процес для користувача. Коли, після завантаження програми, користувач виявить, що більшість місцевих продуктів відсутні, то додаток для нього втратить будь-який сенс.

Третій та основний недолік, який відноситься до кожного з цих додатків, полягає в тому, що розробники пропонують надсилати їм фото не знайдених товарів та їх складів, після чого вони додадуть його самі. Таким чином, користувач не може отримати одразу розшифрований склад, якщо йому це знадобиться.

Саме з цієї причини розробка запропонованої системи є актуальною. Розроблена система буде орієнтована на ринок України та міститиме більшість місцевих продуктів. Крім цього, в майбутньому додатку, який використовуватиме цю систему, можна буде додати свої особливі функції для того, щоб користувач мав можливість максимально зручно та ефективно використовувати цей засіб. Однією з цих функцій є можливість розшифрування складу по фото. Таким чином, відсутність продукту в БД не буде на заваді користувача при отриманні потрібної інформації.

## 2 СИСТЕМИ МАРКУВАННЯ ТОВАРІВ ТА МЕТОДИ РОЗПІЗНАВАННЯ ТЕКСТУ

У наш час необхідність автоматичного розпізнавання тексту з картинок чи фотографій трапляється досить часто. Це може бути пов'язано з перекладом або оцифруванням документів, представлених раніше лише у друкованому вигляді. Можливість швидко розпізнати текст значно спрощує безліч завдань, зокрема завдання цього проекту.

У рамках цієї роботи алгоритм буде реалізований за допомогою нейронної мережі.

### 2.1 Проблема маркування продуктів харчування

У світі більшість продукції, зокрема і продукти харчування, маркуються з допомогою штрих-коду. Ця технологія має безліч плюсів для технологічного процесу, проте для споживача цей вид маркування є не найкращим, тому що не дає можливості швидко отримати повну інформацію про товар [13].

#### 2.1.1 Що таке штрих-код та його недоліки

Штрих-код або штриховий код – це графічна інформація, яка наноситься на поверхню товару, маркування або його упаковку, і яку можна зчитати за допомогою технічних засобів.

Ці коди призначені для спрощення обліку товарів на складі або у магазині. Хоча більшість інформації міститься саме у графічному коді, деякі дані можна отримати, не використовуючи засіб для його читання. Прикладом такої інформації може бути країна-виробник [14].

Стандартний штрих-код відноситься до лінійного типу символіки. На відміну від двовимірних, вони читаються лише в одному напрямку – по

горизонталі. Приклад штрих-коду та тлумачення його цифр можна побачити нижче (рисунок 2.1).



Рисунок 2.1 – Приклад штрих-коду та розшифрування його цифр

Лінійні символіки дозволяють кодувати дуже невеликий обсяг інформації – лише до 20–30 числових символів.

За допомогою штрихового коду зашифрована інформація про деякі найбільш суттєві параметри продукції. Найбільш поширені: американський універсальний товарний код UPC та Європейська система кодування EAN. Найбільш поширеними є EAN/UCC товарні номери EAN-13, EAN-8, UPC-A, UPC-E та 14-розрядний код транспортної упаковки ITF-14. Також існує 128-розрядна система UCC/EAN-128. Відповідно до цієї чи іншої системи, кожному виду виробу надається свій номер, що складається найчастіше з 13 цифр.

Візьмемо, наприклад, цифровий код: 4820024700016. Перші дві цифри (482) означають країну походження (виробника або продавця) продукту, наступні 4 або 5 залежно від довжини коду країни (0024) – підприємство-виробник, ще п'ять (70001) – найменування товару, його споживчі властивості, розміри, масу, колір. Остання цифра (6) контрольна, яка використовується для перевірки правильності зчитування штрихів сканером.

Таблиця кодів країни виробника показана нижче (рисунок 2.2).

Штрих-код	Страна	Штрих-код	Страна	Штрих-код	Страна
00-09	США и Канада	54	Бельгія и Люксембург	779	Аргентина
30-37	Франція	560	Португалія	780	Чилі
380	Болгарія	569	Ісландія	786	Еквадор
383	Словенія	57	Данія	789	Бразилія
385	Хорватія	590	Польща	80-83	Італія
400-440	Германія	599	Венгрія	84	Іспанія
460-469	Росія и СНГ	600-601	ЮАР	850	Куба
471	Тайвань	611	Марокко	858	Словаччина
474	Естонія	613	Алжир	859	Чехія
475	Латвія	619	Туніс	860	Югославія
477	Литва	94	Нова Зеландія	869	Турція
482	Україна	64	Фінляндія	87	Нідерланди
484	Молдова	690	КНР	880	Южна Корея
489	Гонконг	70	Норвегія	885	Таїланд
45 и 49	Японія	729	Ізраїль	888	Сінгапур
50	Великобританія	73	Швеція	890	Індія
520	Греція	750	Мексика	893	В'єтнам
529	Кіпр	759	Венесуела	90-91	Австрія
535	Мальта	76	Швейцарія	93	Австралія
539	Ірландія	770	Колумбія	955	Малайзія

Рисунок 2.2 – Коди країн-виробників

### 2.1.2 Можливість заміни штрих-коду на QR-код у майбутньому

Штрих-код, як маркування, ідеально підходить для компаній-виробників та для магазинів, які використовують їх продукти, проте через те, що в ньому можна зберігати мало інформації, для споживача цей варіант не є найкращим. Натомість, в якості альтернативи, можна розглянути QR-код.

Якщо звернути на це увагу, можна помітити, що QR-код використовується все частіше і частіше. В умовах пандемії, його використовують навіть ресторани, для того, щоб людина могла скористатися електронним меню. Соціальні мережі, для поширення профілів, також пропонують використовувати цей вид символіки. Можна знайти ще безліч прикладів, які показують, наскільки популярні QR-коди для поширення інформації у зручному вигляді.

QR-код відноситься до двовимірної символіки. Так називають символіки, які розроблені для зберігання досить великого обсягу інформації, який може сягати кількох сторінок тексту. Розшифровка такого коду відбувається у двох вимірах: по вертикалі та по горизонталі.

Крім того, що такий код може зберігати в рази більше інформації, ніж звичайний штрих-код, він зчитується набагато швидше (майже вдесятеро). Також зараз практично на будь-якому пристрої з самого спочатку встановлено сканер QR-коду. Завдяки його швидкому зчитуванню та впізнаваності він здобув велику поширеність у різних сферах життя людини.

Стандартний QR-код складається з таких елементів:

- відступ – це біла рамка по зовнішньому краю. Без неї сканер не зможе визначити, що саме потрібно зчитувати – заважатимуть навколишні елементи;

- пошуковий узор – переважно у вигляді чорних квадратів, ідентифікатори для сканера, які повідомляють, що перед ним дійсно знаходиться QR-код;

- вирівнюючий узор – це квадрат меншого розміру, розташований ближче до правого нижнього кута. Він забезпечує читання коду навіть у перекошеному вигляді;

- смуги синхронізації – це г-подібна лінія, що пролягає між трьома квадратами пошукового візерунка. Вона допомагає сканеру визначати окремі квадрати в кодї та дозволяє зчитувати навіть пошкоджений код;

- версія – розташована у невеликому полі біля правого верхнього квадрата пошукового візерунка. Усього QR-код буває чотирьох версій;

- осередки даних – частина коду, що містить інформацію.

Нижче наведено приклад QR-коду (рисунок 2.3).



Рисунок 2.3 – Приклад QR-коду

Зараз багато виробників, крім штрих-коду, розміщують на упаковці своєї продукції і QR-код, що містить більш детальну інформацію про товар. Тим не менш, якщо ці дві символіки розміщені надто близько, то сканер штрих-коду намагається зчитати їх обоє відразу, видаючи помилку та приносячи багато проблем покупцю, особливо якщо він використовує касу самообслуговування.

Якщо замінити штрих-код QR-кодом, це дозволить виробнику робити інформацію про продукт прозорою для споживача. Таким чином, туди можна помістити повний склад та додаткові дані, які можуть бути корисними.

Через те, що штрих-код не містить інформації про склад, мало де є повна база даних із зіставленням значення коду та переліком компонентів продуктів. До того ж часто виробник може змінювати склад своєї продукції, що не завжди може бути швидко перенесено в потенційну БД. З цього випливає, що хоч для споживача і було б краще мати можливість отримати інформацію з QR-коду на упаковці, але на даний момент це дуже складно реалізовано [15].

## 2.2 Нейронна мережа та її типи

Нейронна мережа – це математична модель та її апаратна чи програмна реалізація, побудована за принципом функціонування та організації біологічних нейронних мереж. У результаті, на виході є продукт, що імітує мислення людини.

Насамперед, для завдання розпізнавання тексту, потрібно вибрати тип нейронної мережі, який і використовуватиметься у проекті.

Основні та базові архітектури нейронних мереж – це мережі прямого поширення та перцептрони (рисунок 2.4). Відмінною особливістю даних нейронних мереж є те, що вони дуже прямолінійні та інформація щодо них одразу передається від входу до виходу. Клітини шару таких мереж між собою не пов'язані, на відміну від сусідніх шарів, які, як правило, повністю пов'язані. НМ прямого поширення, переважно, навчаються шляхом зворотного поширення помилки. Завдяки такому навчанню НМ отримує безліч даних і на вхід, і на вихід. Такий процес називається навчанням із вчителем [17].

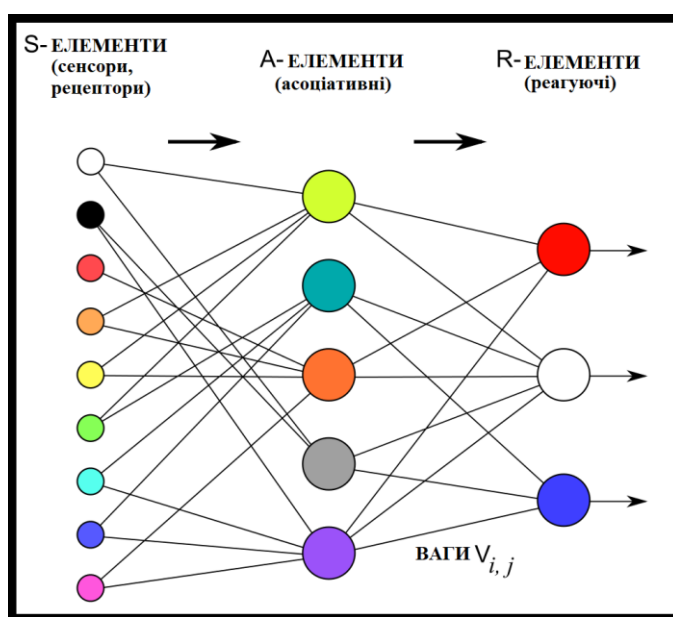


Рисунок 2.4 – Схема будови перцептрону

Наступним різновидом базових НМ є НМ Хопфілда (рисунок 2.5). Ця НМ використовується для відновлення зображень. Така НМ є пов'язаною із симетричною матрицею зв'язку. Те, що вона пов'язана, означає, що кожен нейрон передає свій вхідний сигнал іншим нейронам, включаючи себе. Кількість нейронів такої НМ визначається кількістю входів та виходів. Так само, як попередній тип НМ, мережа навчається з учителем. Це з тим, що з відновлення зображення НМ необхідно знати вихідні параметри відновлюваного зображення [18].

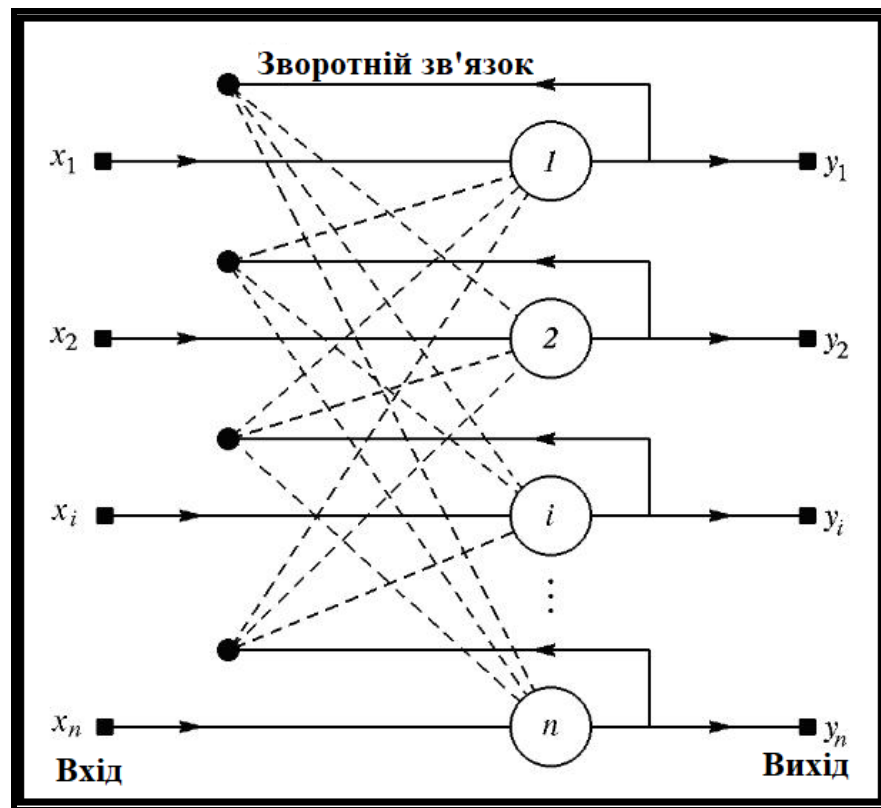


Рисунок 2.5 – Схема побудови НМ Хопфілда

Наступним різновидом базових НМ є Машина Больцмана (рисунок 2.6). У більшості випадків така НМ розглядається як генеративний стохастичний варіант НМ Хопфілда. Ця НМ є першою, здатною навчатися за внутрішнім уявленням та вирішувати складні комбінаторні завдання.

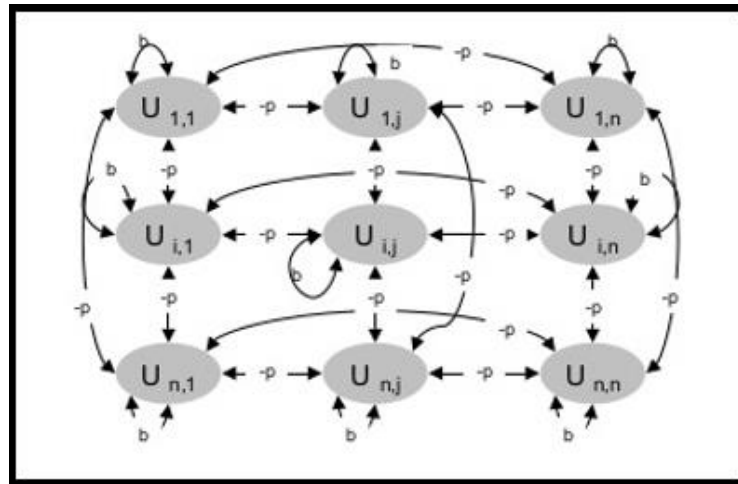


Рисунок 2.6 – Схема побудови НМ Машина Больцмана

Останнім розглянутим різновидом НМ будуть згорткові нейронні мережі або ЗНМ (рисунок 2.7). Такі нейронні мережі дуже відрізняються від тих, які були розглянуті раніше. ЗНМ найчастіше використовують для класифікації зображень. Головною особливістю даної НМ є так званий сканер, який зчитує зображення частинами, і процес згортки, який зменшує розмір матриці ознак зображення.

ЗНМ складається з різних типів шарів: згорткових, субдискретизуючі та шари звичайної НМ [19].

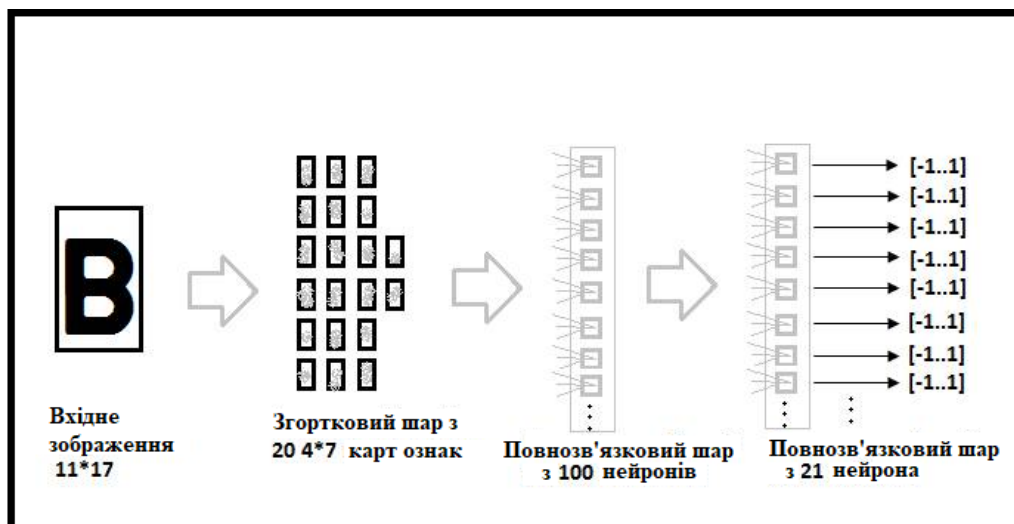


Рисунок 2.7 – Схема побудови ЗНМ

Згорткові та субдискретизуючі шари, чергуючись, формують вхідний вектор ознак для багат шарового персептрону.

Особливу увагу варто приділити згортковому шару, завдяки якому НМ отримала свою назву. Він являє собою набір карт або карт ознак, які мають синаптичне ядро або фільтр. Кількість карток визначається тим, які вимоги були поставлені до завдання, оскільки збільшення їх кількості підвищує точність визначення, проте збільшує обчислювальну складність, і навпаки.

У більшості завдань рекомендується брати співвідношення один до двох – кожна карта попереднього шару має бути пов'язана з двома картами згорткового шару, як показано нижче (рисунок 2.8).

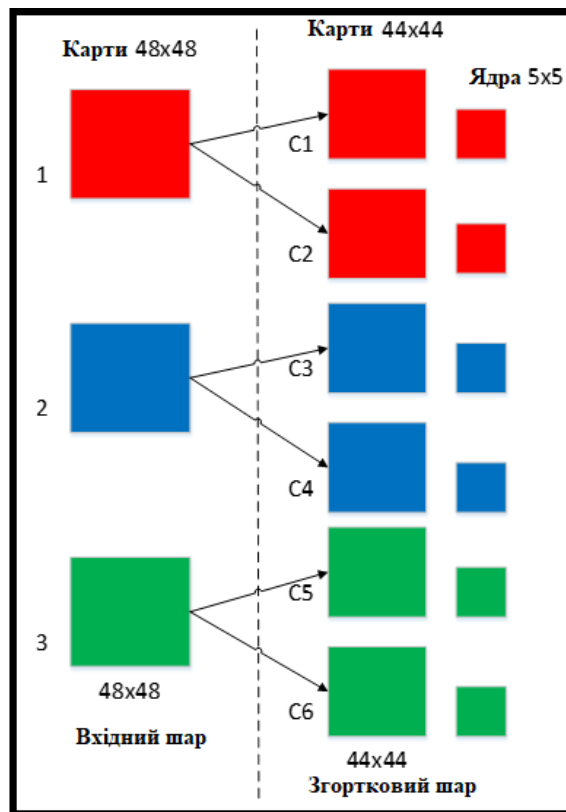


Рисунок 2.8 – Співвідношення карт ознак

Розмір всіх карт згорткового шару обчислюється за однією для всіх згідно з формулою (2.1).

$$(w, h) = (mW - kW + 1, mH - kH + 1), \quad (2.1)$$

де  $(w, h)$  – обчислюваний розмір згорткової картки;

$mW$  – ширина попередньої карти;

$mH$  – висота попередньої карти;

$kW$  – ширина ядра;

$kH$  – висота ядра.

Ядро представляється як вікно або фільтр, який проходить по всій області попередньої карти та знаходить певні ознаки об'єктів. Якщо його розмір буде занадто маленьким, воно не зможе виділяти ознаки, а якщо занадто великим – збільшиться кількість зв'язків між нейронами. Розмір вибирається так само, щоб розмір карт був парним. Це дає можливість не втрачати інформацію при зменшенні розмірності в шарі підвибірки.

Ядро представляє систему ваг, що розподіляються, і синапсів. Це одна з основних особливостей ЗНМ. Загальні ваги дозволяють скоротити кількість зв'язків і дають можливість знаходити ту саму ознаку по всій області зображення (рисунок 2.9).



Рисунок 2.9 – Приклад ядра з навченою ознакою

Початкові значення кожної картки згорткового шару мають значення 0. Значення ваги задаються випадково у вибраному діапазоні. Ядро проходить по попередній карті і виготовляє вёртку (формула 2.2).

$$(f * g)[m, n] = \sum_{k, l} f[m - k, n - l] * g[k, l], \quad (2.2)$$

де  $f$  – вихідна матриця;

$g$  – ядро згортки.

Також цю операцію можна описати таким чином: фільтр розміру ядра  $g$  обходить із заданим кроком все зображення  $f$ , і на кожному кроці поелементно множиться вміст на ядро  $g$ . Результат підсумовується та записується в результуючу матрицю (рисунок 2.10).

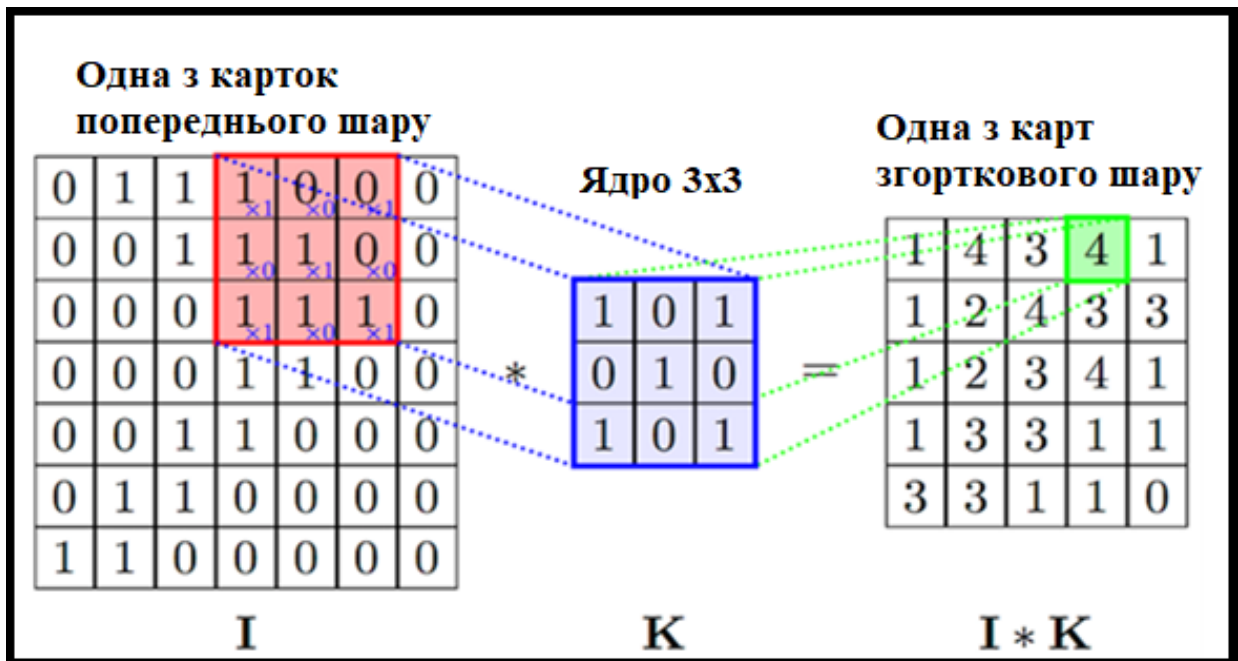


Рисунок 2.10 – Опис згортки

### 2.3 Опис алгоритму розпізнавання тексту

Основні кроки для реалізації функції розпізнавання тексту:

- навчання НМ для розпізнавання українського тексту;
- визначення на зображенні конкретних букв;
- розбиття зображення на дрібніші зображення з конкретними літерами;
- визначення літери на зображенні;
- постановка літери в порядку слів, що йдуть;
- виділення слова;
- розставлення прогалини.

Так як така задача, як розпізнавання тексту, відноситься до задачі класифікації, то для побудови алгоритму було обрано ЗНМ.

За таким принципом працює бібліотека для розпізнавання тексту на Python. Проте перед тим, як подавати зображення безпосередньо в НМ, його потрібно привести до потрібного вигляду.

Щоб виділити на зображенні літери, його потрібно перевести в чорно-біле. Для досягнення цього зображення спочатку переводиться в сірий та його відтінки, а потім, залежно від темряви кольору пікселів, кожен з них набуває чорного або білого кольору.

Після того, як зображення приведено до потрібного вигляду, наступним кроком є визначення контурів літер (рисунок 2.11). Це потрібно для того, щоб у майбутньому визначати межі вступних даних, які вводитимуться.

Алгоритм визначення контурів має такий вигляд:

- знаходження двох найвіддаленіших одна від одної точок контуру;
- знаходження найвіддаленішої точки контуру від відрізка, який був утворений на попередньому кроці;
- такий крок повторюється до виконання умови про довжину порогу.

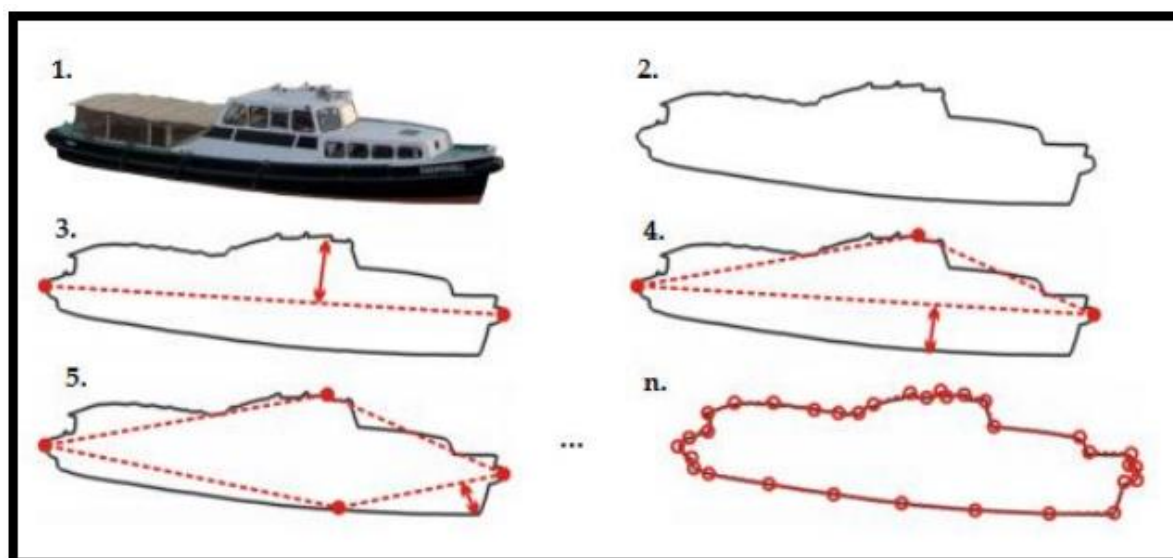


Рисунок 2.11 – Демонстрація роботи алгоритму пошуку контуру

Після того, як контур визначено, область вирізається таким чином, щоб весь контур входив у квадратну форму.

Потім зображення варто призвести до конкретного розміру, оскільки це підвищить ефективність розпізнавання у майбутньому. Після цього готове зображення подається на вхід ЗНМ, де відбудеться його віднесення до конкретного класу, що належить літері українського алфавіту.

Після того, як літери визначені, вони виводяться зліва направо, виходячи з їхнього розташування на початковому зображенні. Пробіл ставиться в тому випадку, якщо відстань між останньою координатою одного зображення та першою – наступного більше або дорівнює третині від розміру зображення до осі X [23].

Повний алгоритм представлений у вигляді схеми нижче (рисунок 2.12).

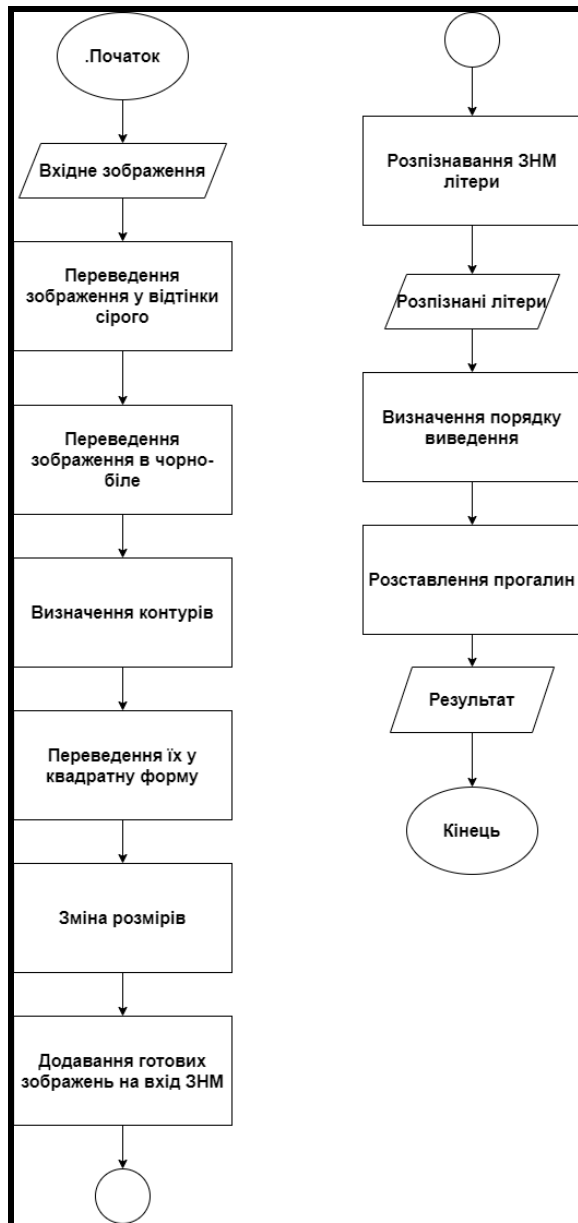


Рисунок 2.12 – Візуалізація алгоритму

Таким чином, було розроблено алгоритм розпізнавання тексту за допомогою ЗНМ. Цей алгоритм зможе досить швидко та точно розпізнавати букви та окремі слова у тому порядку, у якому вони представлені на зображенні.

## 3 ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ

### 3.1 Вибір мови програмування

Для створення ЗНМ можна використовувати будь-яку мову програмування. Тим не менш, якісь з них є зручними для завдань машинного навчання, а якісь – не дуже, оскільки призначені для інших цілей.

Першою та найпопулярнішою мовою програмування в машинному навчанні є Python. Вона має корисні якості, які і вивели її в лідери. Вона дуже універсальна, тобто її можна використовувати як написання ЗНМ, так і хмарного сервісу. Крім цього ця мова має дуже простий синтаксис, що робить її особливо привабливою для новачків або як додаткова мова. І найважливішим для машинного навчання якістю є те, що Python має активну спільноту і безліч інструментів, які дуже допомагають у вирішенні завдань ML. Але крім плюсів, ця мова має і мінуси: вона споживає досить багато пам'яті, що утруднює її використання на малопотужних машинах, а так само у неї досить низька швидкість виконання програм.

Другою популярною мовою для ML є R. Однак вона рідко коли використовується одна – в основному у зв'язці з Python. Тим не менш, вона відмінно підходить для статичної обробки даних та для роботи з графіками. Для неї існує багато пакетів для машинного навчання. В основному R використовується для вирішення складних завдань, і з одного боку це гарна якість, проте з іншого це значно ускладнює вирішення простих завдань із її використанням.

Третьою розглянутою мовою є C++. Ця мова є достатньо потужною для того, щоб програми нею працювали швидко і стабільно. Однак у наш час для багатьох вона здається надто складною у синтаксисі, а велика кількість абстрактних функцій ще більше відлякує розробників від роботи з нею. Саме тому, навіть незважаючи на те, що C++ дає можливість вручну

керувати оперативною пам'яттю та вилученням даних, вона не така популярна і зручна, як попередні дві розглянуті мови.

Останньою розглянутою мовою є Java. Її синтаксис схожий на C++, тобто не такий простий, як Python. Ця мова має непогані бібліотеки для вирішення завдань машинного навчання. Тим не менш, їх набагато менше, ніж у першій розглянутій мові і вони набагато менш функціональні та зручні у використанні. Java краще використовувати для ML тоді, коли розробник знає її на високому рівні, але хоче увійти в машинне навчання, не вивчаючи особливо нічого нового. Для новачків краще вибрати іншу мову.

Після розгляду всіх мов було прийнято рішення використовувати в проєкті Python. Це зумовлено тим, що розв'язуване завдання, з погляду машинного навчання, не настільки складне, щоб підключати R, при цьому, навіть у майбутньому, не повинне бути настільки ресурсно витратним, щоб була необхідність використовувати Java або C++.

## 3.2 Вибір середовища та інструментів

Після того, як було прийнято рішення використовувати Python як основну мову в даному проєкті, наступним завданням стояли вибір засобів та інструментів для реалізації. Крім ЗНМ для розпізнавання тексту, вимагалось створити БД, з якої вибиратися дані. Таким чином, крім того, що потрібно було визначитися із середовищем розробки Python, необхідно було вибрати СКБД.

### 3.2.1 Вибір СКБД

Спочатку було зрозуміло, що для проєкту буде використано реляційну БД. Це було виправдано тим, що в даному проєкті є кілька таблиць, які певним чином пов'язані між собою.

Таким чином, вибір був із таких СКБД:

- MySQL;
- PostgreSQL;
- Oracle;
- MS SQL Server.

Кожен із цих засобів має свої переваги та недоліки.

Перша СКБД – MySQL. Її основна перевага – це простота використання та те, що вона абсолютно безкоштовна. Її переважно використовують для розробки сайтів. Наразі підтримується компанією Oracle [24].

Друга СКБД – PostgreSQL. Вона також є безкоштовною. Її переважно використовують на UNIX-подібних платформах для реалізації проектів різного розміру.

Третя СКБД – Oracle. В основному цю СКБД використовують великі компанії. Це виправдано тим, що вона коштує досить дорого, проте натомість вона дає широкі можливості та високу надійність.

Четверта СКБД – MS SQL Server. Так само, як і попередня, ця система використовується, переважно, великими компаніями. Тим не менш, вона досить зручна, хоч і має менше можливостей і слабкіший захист, порівняно з Oracle.

Після розгляду всіх реляційних СКБД було прийнято рішення використовувати MS SQL Server. Це пов'язано з тим, що цей засіб відмінно поєднується з мовою Python, досить надійний і при цьому дає можливість безкоштовного використання. Таким чином, в даному проекті для створення та заповнення БД було обрано СКБД MS SQL Server [25].

### 3.2.2 Вибір середовища для Python

Для цієї мови програмування існує досить багато середовищ розробки, як і практично для будь-якої популярної мови. Однак у цьому

випадку продукт компанії Jet Brains – PyCharm – підходив ідеально, тому інші засоби не розглядалися як альтернативні. Цей засіб є одним з найкращих повнофункціональних IDE, які призначені для Python. У ньому є можливість безпосередньо із середовища, не використовуючи консоль, завантажувати потрібні бібліотеки, а також дає можливість запускати та налагоджувати проекти. Крім того, як доповнення, цей засіб дає можливість безпосередньо використовувати системи контролю версій, такі як Git [26].

Також невід’ємним плюсом даного засобу є можливість оформлення студентської ліцензії. Таким чином, кожний, хто навчається у вищому навчальному закладі, може застосовувати PyCharm для навчання або для реалізації великих навчальних робіт. Для студентів Jet Brains дають таку ліцензію, що відкриває всі можливості середовища. З цієї причини PyCharm був обраний як основний інструмент для створення системи у даній роботі.

## 4 ПРАКТИЧНА РЕАЛІЗАЦІЯ

### 4.1 Створення БД

#### 4.1.1 Побудова схеми БД

Першим кроком у реалізації проекту було створення зручної та продуманої БД. Для цього було побудовано схему, яка описувала загальну структуру та зв'язки таблиць між собою (рисунок 4.1).

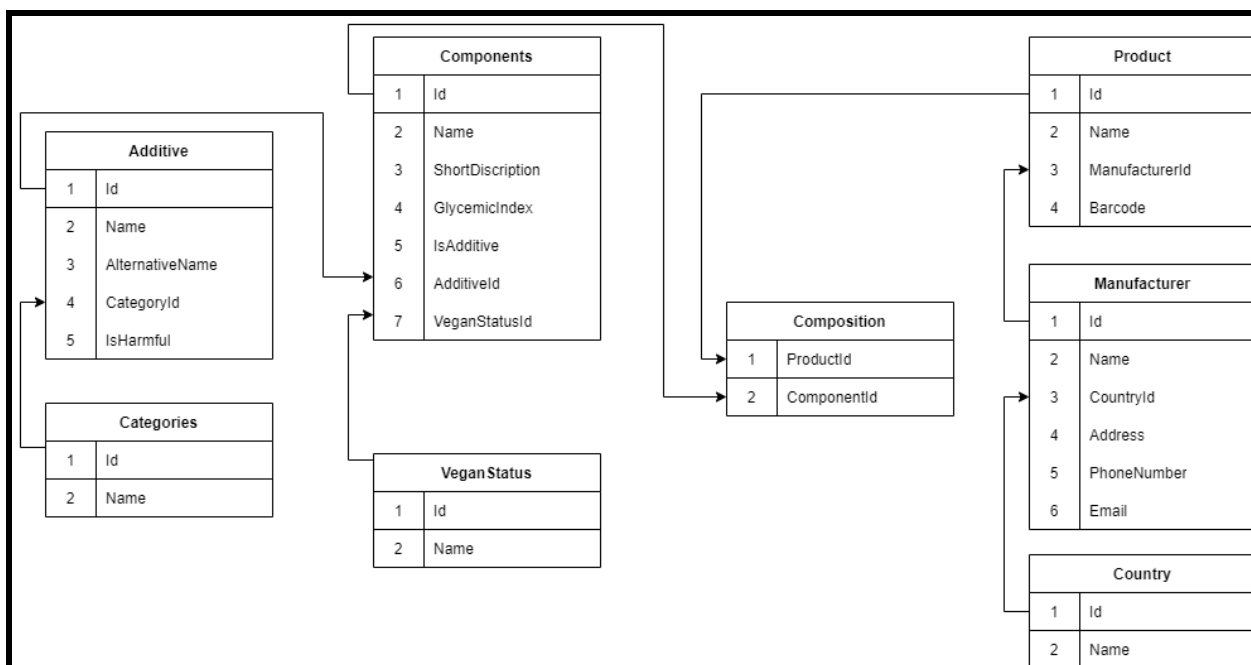


Рисунок 4.1 – Схема БД

Можна помітити, що спроектована БД складається із восьми таблиць. Перша, яку варто розглянути – таблиця категорій (рисунок 4.2).

Categories	
1	Id
2	Name

Рисунок 4.2 – Таблиця категорій

Ця таблиця містить у собі категорії добавок. Як було зазначено раніше, їх існує досить багато, і кожна з них має свою назву. Таблиця має два поля: код та найменування. За схемою можна побачити, що код цієї таблиці є зовнішнім ключем до таблиці з добавками. Таким чином, організовується зв'язок між ними, де код категорії добавки береться з даної таблиці.

Наступна таблиця – таблиця статусів компонентів (рисунок 4.3).

VeganStatus	
1	Id
2	Name

Рисунок 4.3 – Таблиця статусів

Ця таблиця містить у собі статус компонентів. Їх лише три:

- рослинного походження;
- тваринного походження;
- спірно.

Останній статус надається тим компонентам, які можуть мати як рослинне, і тваринне походження. Такими, в основному, є ароматизатори та добавки. У разі надання компоненту такого статусу його обов'язково потрібно перевірити, щоб розуміти, яке походження він має. Це можна

зробити, зв'язавшись із виробником та поставивши йому відповідні питання.

Таблиця, як і попередня, містить поле з кодом статусу та його найменуванням згідно зі списком. Якщо подивитися на схему, можна помітити, що код даної таблиці є зовнішнім ключем до таблиці з компонентами. Таким чином, кожен з компонентів має свій статус, взятий з цієї таблиці.

Наступна розглянута таблиця – таблиця країн (рисунок 4.4).

Country	
1	Id
2	Name

Рисунок 4.4 – Таблиця країн

Ця таблиця зберігає у собі список країн. Як і попередні дві, має два поля – код та найменування. Ця таблиця потрібна для того, щоб її код використовувався як зовнішній ключ до таблиці виробника, що можна побачити на загальній схемі. Таким чином, у цій таблиці зберігаються дані про країну, де знаходиться виробник товару.

Наступна таблиця – таблиця добавок (рисунок 4.5).

Additive	
1	Id
2	Name
3	AlternativeName
4	CategoryId
5	IsHarmful

Рисунок 4.5 – Таблиця добавок

Ця таблиця призначена для того, щоб зберігати інформацію про добавки. Вона містить такі поля: код, найменування, альтернативне ім'я, код категорії та логічний параметр, що вказує на те, шкідлива ця добавка чи ні. Альтернативне ім'я потрібне для добавок, що має позначення, яке складається з E та числового коду, і являє собою хімічну назву. Код цієї таблиці є первинним ключем для таблиці з компонентами і дає можливість отримати докладнішу інформацію про добавки, зазначені у складі продукту харчування.

Наступна таблиця – таблиця компонентів (рисунок 4.6).

Components	
1	Id
2	Name
3	ShortDiscription
4	GlycemicIndex
5	IsAdditive
6	AdditiveId
7	VeganStatusId

Рисунок 4.6 – Таблиця компонентів складу

Ця таблиця є однією з основних у цій БД. Вона складається з таких полів, як код, найменування, короткий опис, глікемічний індекс, ознака добавки, код добавки та статус, що вказує на походження компонента. Дані в таблиці потрібні для того, щоб розшифрувати склад для користувача. Короткий опис дає більше розуміння, що міститься в продукті харчування. Глікемічний індекс – це необхідна інформація для діабетиків різних типів, а також для людей, які намагаються дотримуватися правильного харчування.

Якщо подивитися на схему БД, можна помітити, що ця таблиця безпосередньо пов'язані з таблицею складу продукту. Первинний ключ є зовнішнім до таблиці складу продукту, даючи можливість за коротким числовим кодом отримати інформацію про складові продуктів.

Наступна таблиця – таблиця виробників (рисунок 4.7).

Manufacturer	
1	Id
2	Name
3	CountryId
4	Address
5	PhoneNumber
6	Email

Рисунок 4.7 – Таблиця виробників

Ця таблиця зберігає у собі основну інформацію по виробнику. Вона містить такі поля: код, найменування, код країни, адресу виробника, номер телефону та електронну пошту. Вона пов'язана за кодом із таблицею товарів. Дані цієї таблиці особливо важливі для того, щоб користувач міг уточнювати будь-які моменти за складом у виробника. Це потрібно тому, що часто у складі не точно вказані ароматизатори та походження добавок. Це особливо важливо для тих, хто дотримується веганства, тому можливість отримання контактів виробника – важлива частина проекту.

Наступна таблиця – таблиця товарів (рисунок 4.8).

Product	
1	Id
2	Name
3	ManufacturerId
4	Barcode

Рисунок 4.8 – Таблиця товарів

Ця таблиця – це друга важлива таблиця БД, що стоїть на рівні з компонентами складу продукту. Вона містить такі поля, як код, найменування, код виробника та штрих-код. Вона потрібна для того, щоб зіставляти лічений код із товаром і швидко знаходити його склад через зв'язки бази даних. Так само зв'язок з таблицею виробників дає можливість швидко та просто отримати інформацію про нього. Повертаючись до загальної схеми БД, можна помітити, що дана таблиця пов'язана за кодом з таблицею складу.

Наступна таблиця – таблиця складу (рисунок 4.9).

Composition	
1	ProductId
2	ComponentId

Рисунок 4.9 – Таблиця складу

Ця таблиця, хоч і має у собі лише два поля – найважливіша у цьому проекті. Вона поєднує у собі дві великі частини – товари та компоненти. Завдяки їй можна отримати розширену інформацію як про складові продукту, так і про виробника. Вона не має первинного ключа, проте має унікальне обмеження для комбінації коду товару та коду компонента. Дані

в ній будуються таким чином: на один код товару може бути один або більше кодів компонентів. Таким чином, виходить загальний склад.

#### 4.1.2 Створення БД за складеною схемою

Для створення та заповнення БД використовувалася СКБД SQL Server. Вибір цього засобу обумовлений тим, що взаємодія з Python встановлюється просто і без проблем.

Насамперед створюються таблиці. Їх слід створювати в порядку: від довідників до основних таблиць. Це пов'язано з тим, що таблиці вищого рівня використовують низькорівневі таблиці, і якщо точніше – їх первинні ключі, як свої зовнішні. Через це не вийде їх відразу зв'язати, а якщо таблиця вже міститиме якісь дані, то це може бути взагалі неможливо без видалення або додавання даних у відповідні таблиці.

Таким чином, порядок створення таблиць виглядає так:

- таблиця категорій добавок;
- таблиця статусів походження;
- таблиця добавок;
- таблиця компонентів;
- таблиця країн;
- таблиця виробників;
- таблиця товарів;
- таблиця складу.

Первинні ключі таблиць завжди будуть мати цілий числовий тип, а найменування, яке є практично у всіх таблицях – строковий різних розмірностей. Таким чином, в описі створення таблиць будуть описані тільки ті поля, які унікальні для об'єкта, що розглядається і мають особливий тип даних або по суті, або за розмірністю.

Створення таблиці категорій добавок харчування відбувається як показано нижче (рисунок 4.10).

```
CREATE TABLE Categories (
    Id INT NOT NULL,
    Name nvarchar(200) NOT NULL,
    CONSTRAINT PK_Categories PRIMARY KEY (Id)
);
```

Рисунок 4.10 – Створення таблиці категорій

Створення таблиці статусів харчування відбувається так (рисунок 4.11).

```
CREATE TABLE VeganStatus (
    Id INT NOT NULL,
    Name nvarchar(200) NOT NULL,
    CONSTRAINT PK_VeganStatus PRIMARY KEY (Id)
);
```

Рисунок 4.11 – Створення таблиці статусів

Створення таблиці добавок продукту харчування в такий спосіб як показано нижче (рисунок 4.12).

```
CREATE TABLE Additive (
    Id INT NOT NULL,
    Name nvarchar(200) NOT NULL,
    AlternativeName nvarchar(200),
    CategoryId INT,
    IsHarmful bit,
    CONSTRAINT PK_Additive PRIMARY KEY (Id),
    CONSTRAINT FK_AdditiveCategories FOREIGN KEY (CategoryId) REFERENCES Categories(ID)
);
```

Рисунок 4.12 – Створення таблиці добавок

Ця таблиця містить такі унікальні поля, як: альтернативне ім'я, яке має рядковий тип розмірності 200 символів, і ознака шкідливості, що має логічний тип даних, який може набувати значення 0 – брехня, 1 – правда, і пусте значення.

Створення таблиці компонентів продукту відбувається так (рисунок 4.13).

```
CREATE TABLE Components (  
  Id INT NOT NULL,  
  Name nvarchar(200) NOT NULL,  
  ShortDiscription nvarchar(500),  
  GlycemicIndex INT,  
  IsAdditive bit,  
  AdditiveId INT,  
  VeganStatusId INT,  
  CONSTRAINT PK_Components PRIMARY KEY (Id),  
  CONSTRAINT FK_ComponentsAdditive FOREIGN KEY (AdditiveId) REFERENCES Additive(ID),  
  CONSTRAINT FK_ComponentsVeganStatus FOREIGN KEY (VeganStatusId) REFERENCES VeganStatus(ID)  
);
```

Рисунок 4.13 – Створення таблиці компонентів

Ця таблиця містить такі унікальні поля, як: короткий опис (має рядковий тип на 200 символів), глікемічний індекс (має цілий тип) і ознака добавки (має логічний тип).

Пов'язана з таблицями добавок та статусів за первинними ключами даних таблиць.

Створення таблиці країн, до яких належить виробник продукту харчування, відбувається в такий спосіб (рисунок 4.14).

```
CREATE TABLE Country (  
  Id INT NOT NULL,  
  Name nvarchar(200) NOT NULL,  
  CONSTRAINT PK_Country PRIMARY KEY (Id)  
);
```

Рисунок 4.14 – Створення таблиці країн

Створення таблиці виробників продуктів харчування відбувається в такий спосіб (рисунок 4.15).

```

CREATE TABLE Manufacturer (
  Id INT NOT NULL,
  Name nvarchar(200) NOT NULL,
  CountryId INT,
  Address nvarchar(500),
  PhoneNumber nvarchar(100),
  Email nvarchar(100),
  CONSTRAINT PK_Manufacturer PRIMARY KEY (Id),
  CONSTRAINT FK_ManufacturerCountry FOREIGN KEY (CountryId) REFERENCES Country(ID)
);

```

Рисунок 4.15 – Створення таблиці виробників

Ця таблиця містить такі унікальні поля, як: адреса (має рядковий тип розмірністю 500 символів), номер телефону (має рядковий тип розмірністю 100 символів для можливості вказівки коду країни та додаткових символів) та електронну пошту (має рядковий тип розмірністю 100 символів).

Також дана таблиця пов'язані з таблицею країн за її первинним ключем.

Створення таблиці товарів відбувається в такий спосіб (рисунок 4.16).

```

CREATE TABLE Product (
  Id INT NOT NULL,
  Name nvarchar(200) NOT NULL,
  ManufacturerId INT,
  Barcode nvarchar(100),
  CONSTRAINT PK_Product PRIMARY KEY (Id),
  CONSTRAINT FK_ProductManufacturer FOREIGN KEY (ManufacturerId) REFERENCES Manufacturer(ID)
);

```

Рисунок 4.16 – Створення таблиці товарів

Ця таблиця має таке унікальне поле, як штрих-код, який має рядковий тип розміру 100 символів.

Також дана таблиця пов'язана зовнішнім ключем з таблицею виробників з її первинному ключу.

Створення таблиці складу продуктів харчування відбувається в такий спосіб (рисунок 4.17).

```
CREATE TABLE Composition (  
    ProductId INT NOT NULL,  
    ComponentId INT NOT NULL,  
    CONSTRAINT UNQ_Composition UNIQUE(ProductId, ComponentId)  
);
```

Рисунок 4.17 – Створення таблиці складу

Ця таблиця не містить унікальних полів, однак і не містить первинного ключа. Мета цієї таблиці – об'єднання двох інших важливих таблиць проекту – таблиці компонентів та таблиці товарів. Вона має унікальне обмеження для первинних ключів-кодів цих двох таблиць. Це потрібно для того, щоб у одного продукту могли бути лише унікальні складові, які не повторюватимуться. Це виключає можливість виникнення помилок у даних. Унікальне обмеження, як і первинний ключ, створює індекс для таблиці, завдяки чому доступ до складу буде відбуватися максимально швидко, проте використання первинного ключа для даної таблиці було зайвим, оскільки поля даних таблиць за замовчуванням не можуть бути порожніми.

#### 4.2 Приклад заповнення БД

Після того, як БД була створена за схемою, зазначеною вище, потрібно заповнити її даними. Наприклад було взято певний товар, після чого його склад, як і дані про нього, було занесено до таблиць.

Заповнення таблиць даними має відбуватися приблизно в такому ж порядку, як і їх створення. Як і в попередньому випадку, це пов'язано з тим, що зв'язки між таблицями передбачають такий порядок, щоб уникнути майбутніх помилок.

Заповнення відбувалося таким чином (рисунок 4.18).

```

INSERT INTO Categories VALUES (1, 'Емульгатор');
INSERT INTO VeganStatus VALUES (1, 'Рослинного походження');
INSERT INTO VeganStatus VALUES (2, 'Животного походження');
INSERT INTO VeganStatus VALUES (3, 'Спірно');
INSERT INTO Additive VALUES (1, 'E476', 'Полігліцерин', 1, 0);
INSERT INTO Components VALUES (1, 'какао терте', 'Перетерті какао-боби', 24, 0, NULL, 1);
INSERT INTO Components VALUES (2, 'какао-масло', 'Жир з тертого какао', NULL, 0, NULL, 1);
INSERT INTO Components VALUES (3, 'какао-порошок зі зменшеним вмістом жиру',
    'Перетерті какао-боби зі зменшеним вмістом жиру',
    24, 0, NULL, 1);
INSERT INTO Components VALUES (4, 'лецитин соєвий', NULL, 15, 0, NULL, 1);
INSERT INTO Components VALUES (5, 'E476', 'Додаток-емульгатор', NULL, 1, 1, 1);
INSERT INTO Components VALUES (6, 'ароматизатор', NULL, NULL, 0, NULL, 3);
INSERT INTO Country VALUES (1, 'Україна');
INSERT INTO Manufacturer VALUES (1, 'ПрАТ "Монделіс"', 1, '42600, Україна, Сумська обл.,
    м.Тростянець, вул. Набережна, 28А', '+ (373 228) 55-678', NULL);
INSERT INTO Product VALUES (1, 'Чорний шоколад "Корона"', 1, '7622210815385');
INSERT INTO Composition VALUES (1, 1);
INSERT INTO Composition VALUES (1, 2);
INSERT INTO Composition VALUES (1, 3);
INSERT INTO Composition VALUES (1, 4);
INSERT INTO Composition VALUES (1, 5);
INSERT INTO Composition VALUES (1, 6);

```

Рисунок 4.18 – Заповнення БД

Як приклад було взято товар «Чорний шоколад «Корона»». Додаткову інформацію про компоненти складу було взято з інтернету.

Можна помітити, що склад цього товару містить у собі такий продукт, як «ароматизатор». У таблиці статусів він має статус «спірно» і це не так. Це пов'язано з тим, що ароматизатором називають ефірну олію, олеорезин, есенцію або екстракт, гідролізат білка, дистилят або будь-який продукт кальцинації, нагрівання або ензимолізу, який містить ароматизуючі елементи, витягнуті зі спеції, фрукту або фруктового соку, овочів або овочевого соку, харчових дріжджів, трави, кори, нирки, кореня, листя або аналогічного рослинного матеріалу, а також з м'яса, морських водоростей, дичини, яєць, молочних продуктів або ферментованих продуктів. Простіше кажучи, ароматизатор не завжди буває тваринного походження.

Саме тому важливо уточнювати такі питання у виробника, після чого, у більшості випадків, можна отримати таку відповідь (рисунок 4.19).



Рисунок 4.19 – Відповідь виробника щодо ароматизатора

Однак у БД було прийнято рішення залишити статус даного компонента спірним, оскільки різні виробники можуть так само вказати у складі ароматизатор без більш детальної інформації. Надалі структуру може бути переглянуто для додавання ознаки перевіреного товару, проте на даний момент він не був передбачений.

Зрештою було складено приблизний запит для отримання даних у проєкті (рисунок 4.20).

```

SELECT cm.id,
       cm.Name AS ComponentName,
       cm.ShortDiscription,
       'Глікемічний індекс = '+ISNULL(CAST(cm.GlycemicIndex AS nvarchar(3)), '~'),
       CASE cm.IsAdditive
         WHEN 1 THEN 'Це додаток '+a.Name+' або '+
                   a.AlternativeName+' категорії '+
                   ct.Name+'. Шкідливий: '+
                   CASE a.IsHarmful
                     WHEN 1 THEN 'Так'
                     ELSE 'Ні' END
         ELSE 'Не є додатком'
       END AS IsAdditive,
       v.Name AS VStatusName
FROM Composition c,
     Product p,
     Components cm,
     VeganStatus v,
     Additive a,
     Categories ct
WHERE c.ProductId = p.id |
     AND c.ComponentId = cm.id
     AND cm.VeganStatusId = v.id
     AND cm.AdditiveId = a.id
     AND a.CategoryId = ct.id
     AND p.Barcode = '7622210815385'

```

Рисунок 4.20 – Запит для отримання даних

Результат запиту БД виглядає наступним чином (рисунок 4.21).

ID	NAME	SHORTDISCRPTION	GLYCEMICINDEX
1	какао терте	Перетерті какао-боби	Глікемічний індекс = 24
2	какао-масло	Жир з тертого какао	Глікемічний індекс = ~
3	какао-порошок зі зменшеним вмістом жиру	Перетерті какао-боби зі зменшеним вмістом жиру	Глікемічний індекс = 24
4	лецитин соєвий		Глікемічний індекс = 15
5	E476	Додаток-емульгатор	Глікемічний індекс = ~
6	ароматизатор		Глікемічний індекс = ~
ISADDITIVE	VSTATUSNAME		
Не є додатком	Рослинного походження		
Не є додатком	Рослинного походження		
Не є додатком	Рослинного походження		
Не є додатком	Рослинного походження		
Це додаток E476 або Полігліцерин категорії Емульгатор. Шкідливий: Ні	Рослинного походження		
Не є додатком	Спірно		

Рисунок 4.21 – Результат запиту в БД

### 4.3 Використання Tesseract

Бібліотека Tesseract є ЗНМ, навченою розпізнавати текст. Її використання було зумовлено тим, що склад товарів часто буває різних кольорів і на різному тлі, а також може мати різний шрифт і форму. Ця

проблема не дозволила знайти відповідну та досить велику вибірку для навчання нової ЗНМ розпізнаванню української мови. Tesseract вже навчений досить точно розпізнавати різні шрифти – точність розпізнавання в нормальних умовах 70%. У разі сильно пошкодженого зображення – 30%. Цього цілком достатньо для того, щоб розпізнати склад тексту з упаковки товару фото.

Крім цієї бібліотеки використовувалася також cv2 для обробки вхідного зображення.

#### 4.4 Складання фінальної версії проекту

Після того, як обидві великі частини проекту готові, потрібне їхнє об'єднання. Воно включає створення взаємодії між підключеною БД і результатом розпізнавання тексту з використанням Tesseract.

Насамперед потрібно перетворити зображення до потрібного вигляду та встановити режим розпізнавання української мови (рисунок 4.22).

```
pytesseract.pytesseract.tesseract_cmd = 'C:\\tess\\tesseract.exe'

img = cv2.imread('test_6.jpg')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

config = r'--oem 3 --psm 6'

text = pytesseract.image_to_string(img, config=config, lang='ukr')
data = pytesseract.image_to_data(img, config=config, lang='ukr')
print()
print('Результат розпознавання:')
print(text)
```

Рисунок 4.22 – Підключення Tesseract та CV2

Наступним важливим кроком є обробка результату розпізнавання составу (рисунок 4.23).

```
N = text.find('Склад:') + 7

textForList = text[N:]
|
print(textForList)

textForList = textForList[0:textForList.find('.')]

print()
print('Результат:')
print(textForList)

print()
print('Масив:')
list = textForList.split(", ")

print(list)
```

Рисунок 4.23 – Вилучення з розпізнаного тексту масив із компонентами

Текст, отриманий у результаті розпізнавання з фотографії, у будь-якому разі матиме багато зайвого. По-перше – це оточуючий склад текст та зайві зображення, які можуть бути розпізнані як будь-які літери. Таким чином, щоб виділити з отриманого тексту лише склад продукту, використовуються засоби Python для роботи з рядками.

Насамперед шукається так звана «точка входу» – це слово «Склад» та знак «:». Операція `text.find` знаходить входження пошукового слова та повертає його першу позицію у рядку. Після цього до отриманої позиції додається кількість символів, що займає «Склад:» – це 7 символів. Після цього ліва частина тексту відсікається. У рядку, що залишився, відсікається права частина, починаючи з точки. Після цього результат кроку готовий до подальшої обробки – розподілу на масив. Це робиться за допомогою засобу `split`. Він розбиває на масив рядок, вважаючи роздільником символ «,». Таким чином, у результаті виходить масив з усіма компонентами складу продукту.

Наступним кроком є підключення БД до проекту (рисунок 4.24).

```
serverName = "BRISTOL"  
dbName = "master"  
  
connection = pyodbc.connect('Driver={SQL Server};'  
                             '~Server=BRISTOL;'  
                             '~Database=master;')  
  
cursor = connection.cursor()
```

Рисунок 4.24 – Підключення до БД

Підключення БД, створеної за допомогою СКБД SQL Server, здійснюється за допомогою бібліотеки Python – pyodbc. Вона дає можливість отримати дані та працювати з ними. Для підключення використовується ім'я СКБД, ім'я сервера – в даному випадку комп'ютера – та ім'я самої БД.

Після того, як з'єднання встановлено, потрібно скласти запит, який витягуватиме дані з БД. Для наочності запит було перенесено у сам проект, хоча практично було б краще використовувати цих цілей об'єкт представлення. Він є запитом, який зберігається, за фактом, як таблиця. Уявлення часто використовуються під час роботи з БД і дають можливість захвати від користувачів чи інших розробників деталі вилучення даних із таблиць.

Для цього проекту використовувався запит наступного виду (рисунок 4.25).

```

query = ("""
        SELECT cm.id,
               cm.Name AS ComponentName,
               ISNULL(cm.ShortDiscription, ''),
               ISNULL(CAST(cm.GlycemicIndex AS nvarchar(3)), '~'),
               cm.IsAdditive,
               a.Name as AdditiveName,
               a.AlternativeName,
               ct.Name as CtName,
               a.IsHarmful,
               v.Name AS VStatusName
        FROM Composition c,
             Product p,
             Components cm,
             VeganStatus v,
             Additive a,
             Categories ct
        WHERE c.ProductId = p.id
              AND c.ComponentId = cm.id
              AND cm.VeganStatusId = v.id
              AND (cm.AdditiveId = a.id or cm.AdditiveId is null)
              AND a.CategoryId = ct.id
              AND p.Barcode = '7622210815385'
        """)

cursor.execute(query)
resultsQ = cursor.fetchall()

print(resultsQ)

```

Рисунок 4.25 – Кінцевий запит для отримання даних

Можна помітити, що для цього запиту не використовувалися таблиці виробника та країн. Надалі, при використанні створеної системи, використання саме запиту дає можливість змінити кількість даних і розширити їх полями з таблиць, що не використовуються.

Наступним кроком є зв'язок розпізнавання та складання масиву з вилученими з БД даними (рисунок 4.26).

```

print()
print('Склад з описом:')

for i in results0:
    for el in list:
        if (i[1] == el) or (i[1].startswith(el)) or (el.startswith(i[1])):
            if i[4]:
                addStr = 'Додаток ' + i[5] + ' або ' + i[6] + ' категорії ' + i[7] + '. Шкідливий: ' + str(i[8]) + '. '
            else:
                addStr = 'Не є додатком. '

            print(str(i[0]) + ' Назва компоненту: ' + i[1] + '. Опис: ' + i[2] + 'Глекімічний індекс = ' + str(i[3]) + '. ' + addStr + i[9])

```

Рисунок 4.26 – Зіставлення масиву складу та даних із БД

Для того, щоб отримувати додаток, до проекту додано перевірку та складання повідомлення для таких випадків. У Python конкатенація рядків відбувається за допомогою символу +.

Для того, щоб показати, як система знаходить текст на зображенні, було додано візуальну демонстрацію даного процесу (рисунок 4.27).

```

for i, el in enumerate(data.splitlines()):
    if i == 0:
        continue

    el = el.split()
    try:
        x, y, w, h = int(el[6]), int(el[7]), int(el[8]), int(el[9])
        cv2.rectangle(img, (x, y), (w + x, h + y), (0, 255, 0), 2)
    except IndexError:
        print("Операция была пропущена")

cv2.imshow('Result', img)
cv2.waitKey(0)

```

Рисунок 4.27 – Наочна демонстрація процесу розпізнавання

В подальшому цю демонстрацію можна використовувати для того, щоб прямо із зображення переходити до конкретного компонента складу.

#### 4.5 Тестування роботи проекту

Тестування роботи створеної системи проводилося на продукті компанії «Монделіс Україна» – «Чорний шоколад «Корона»» (рисунок 4.28).



Рисунок 4.28 – Продукт для тестування «Чорний шоколад «Корона»»

Першим кроком для тестування був зроблений знімок складу на звичайний смартфон, який би використовувався для цих цілей кінцевим користувачем (рисунок 4.29).

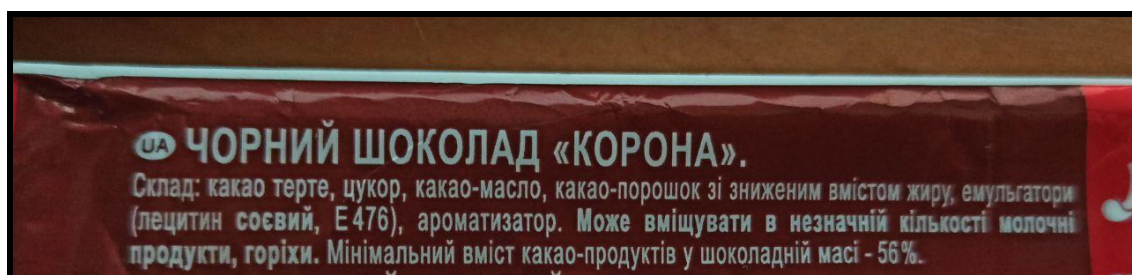


Рисунок 4.29 – Фото складу

Після того, як зображення було подано на вхід, було отримано результати наступного типу:

- результат розпізнавання тексту (рисунок 3.30).;
- результат виділення складу з отриманого тексту (рисунок 3.31);
- результат перетворення на масив (рисунок 3.32).;
- результат отриманого розширеного складу (рисунок 3.33).

Результат розпізнавання:

ін

А я уч нь

- ФЧОРНИЙ ШОКОЛАД «КОРОНА».

Склад: какао терте, цукор, какао-масло, какао-порошок зі зниженим вмістом жиру, емульгатори - (лецитин соєвий, E476), ароматизатор. Може вмішувати в незначній кількості молочні продукти, горіхи. Мінімальний вміст какао-продуктів у шоколадній масі - 56.5. б.

какао терте, цукор, какао-масло, какао-порошок зі зниженим вмістом жиру, емульгатори - (лецитин соєвий, E476), ароматизатор. Може вмішувати в незначній кількості молочні продукти, горіхи. Мінімальний вміст какао-продуктів у шоколадній масі - 56.5. б.

Рисунок 4.30 – Результат розпізнавання

Результат:

какао терте, цукор, какао-масло, какао-порошок зі зниженим вмістом жиру, емульгатори - (лецитин соєвий, E476), ароматизатор

Рисунок 4.31 – Результат виділення складу

Масив:

['какао терте', 'цукор', 'какао-масло', 'какао-порошок зі зниженим вмістом жиру', 'емульгатори -\n(лецитин соєвий', 'E476)', 'ароматизатор']

Рисунок 4.32 – Результат перетворення на масив

Склад з описом:

1 Назва компоненту: какао терте. Опис: Перетерті какао-бобиГлекімічний індекс = 24. Не є додатком. Рослинного походження  
2 Назва компоненту: какао-масло. Опис: Жир з тертого какаоГлекімічний індекс = -. Не є додатком. Рослинного походження  
5 Назва компоненту: E476. Опис: Додаток-емульгаторГлекімічний індекс = -. Додаток E476 або Полігліцерин категорії Емульгатор. Шкідливий: False. Рослинного походження  
6 Назва компоненту: ароматизатор. Опис: Глекімічний індекс = -. Не є додатком. Спирно

Рисунок 4.33 – Результат розширеного складу

Крім цього, отримано результат наочної демонстрації пошуку об'єктів на зображенні (рисунок 4.34).

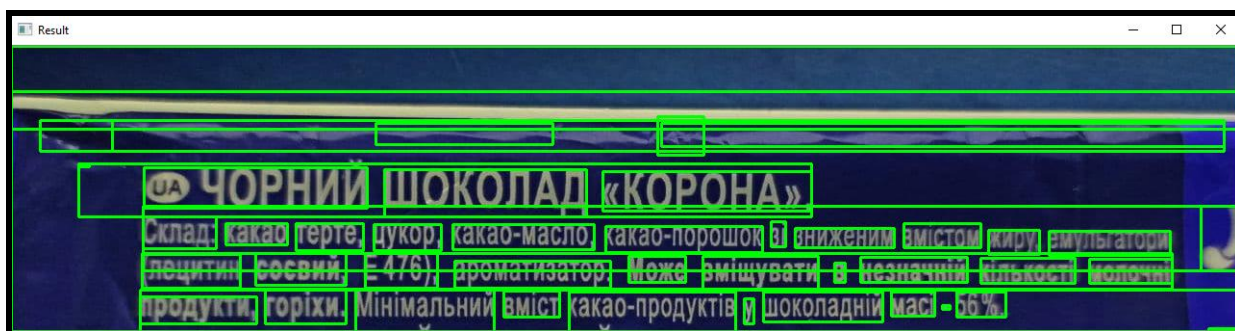


Рисунок 4.34 – Наочна демонстрація пошуку об'єктів на зображенні

Наступний приклад роботи системи – склад товару «Томатний сік» (рисунок 4.35).

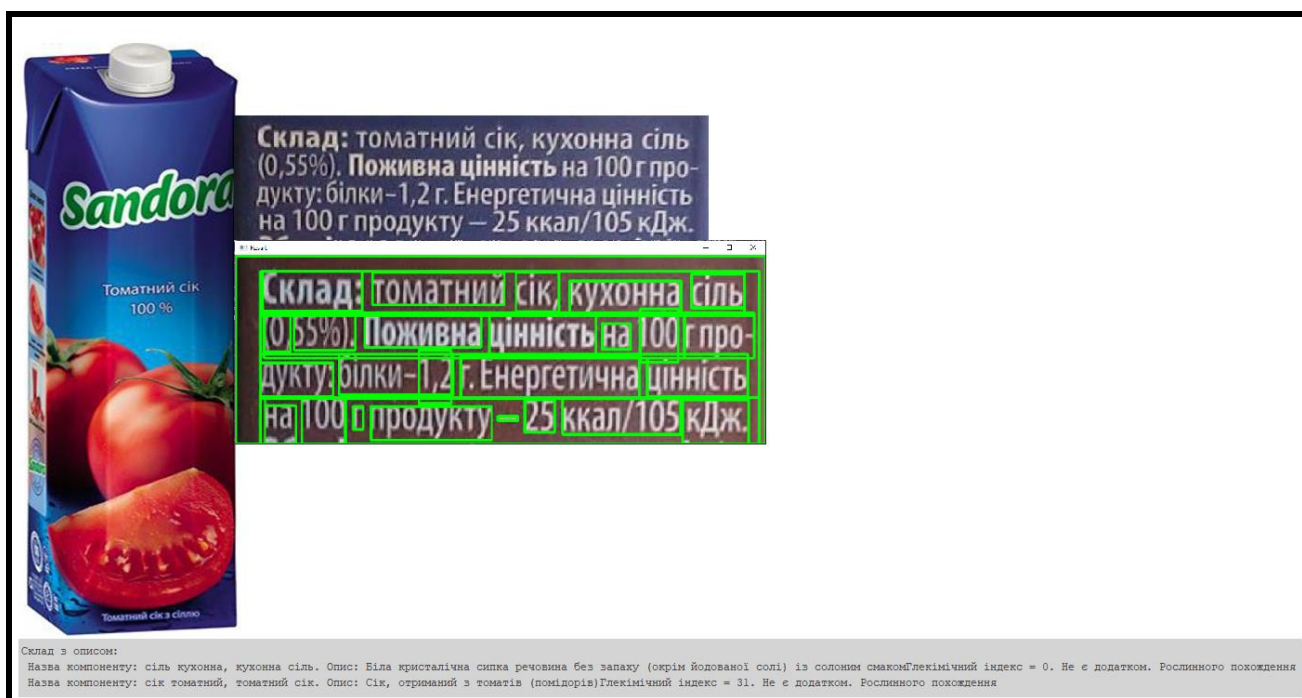


Рисунок 4.35 – Склад товару «Томатний сік»

Наступний приклад роботи системи – склад товару «Хлібці» (рисунок 4.36).

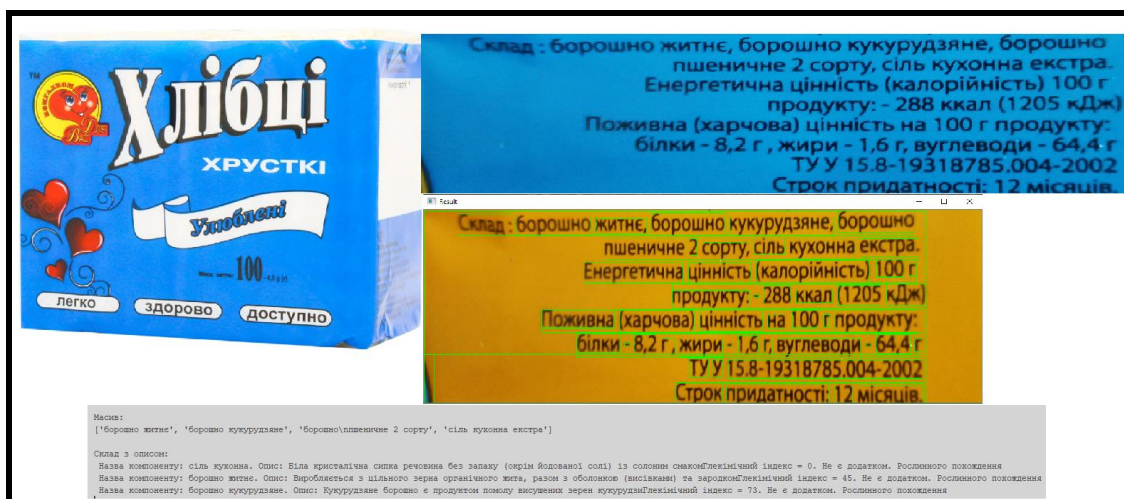


Рисунок 4.36 – Склад товару «Хлібці»

Наступний приклад роботи системи – склад товару «Квасоля біла у томатному соусі» (рисунок 4.37).



Рисунок 3.37 – Склад товару «Квасоля біла у томатному соусі»

Таким чином, можна побачити, що розроблена система досить точно визначає склад товару за зображенням або фотографією, а також, згідно з описом, що зберігається у БД, дає його повну розшифровку. Якщо товар визначається не за штрих-кодом, а тільки по фотографії складу – це не є проблемою, бо тоді система знайде компонент за розпізнаною назвою і дасть користувачу розширену інформацію по ньому.

## ВИСНОВКИ

В результаті дослідження теми даної роботи було детальніше розглянуто проблему недоступності повної інформації про склад продуктів. Насправді ця проблема стосується кожного – через стан здоров'я, принципи або просто бажання розуміти, що людина вживає в їжу. Розуміння складу продукту та окремих його компонентів – це ключ до усвідомленого споживання продуктів харчування. Це дає можливість відмовитися від шкідливих добавок та зменшувати витрати на продукти, які можуть зашкодити здоров'ю.

Тісно це питання стосується веганства та людей, які дотримуються цього типу харчування. Для таких людей важливо розуміти, що написано у складі та чи не заклав виробник у черговий зашифрований компонент продукти тваринного походження. Таким чином, проект дає можливість зробити склад прозорим. Завдяки цьому більше людей можуть подумати про те, що експлуатація тварин – не вихід для сучасного світу. На даний момент існує безліч аналогів, які можуть стати повноцінною заміною продуктів тваринного походження. Разом з цим, варто звернути увагу на те, що м'ясні вироби, здебільшого, можуть сприяти розвитку раку та інших різноманітних захворювань. Навіть з цього погляду веганський тип харчування є найкращою альтернативою для довгого та здорового життя.

Ця робота орієнтована те що, щоб надати користувачам можливість без проблем розуміти склад продуктів харчування. Крім людей, які страждають на будь-які захворювання, це може бути корисно, наприклад, для тих, у кого погіршується зір. Виробники часто друкують склад зовсім дрібним шрифтом, і через це прочитання здається неможливим навіть тим, хто має ідеальний зір.

Для реалізації використовувалася реляційна база даних та згортована нейронна мережа для класифікації зображень, а якщо точніше – для розпізнавання тексту.

Складністю реалізації була відсутність структурованих даних про склад продуктів харчування і про них самих. Через те, що в магазинах, для зручності зберігання та продажу, використовується штрих-код, інформація про склад ніде не зберігається, окрім виробника. Це ускладнює завдання заповнення бази даних, оскільки силами однієї людини дуже складно заповнити базу, в якій повинні зберігатися тисячі продуктів і ще більше компонентів цих продуктів.

Ще однією проблемою було те, що для навчання ЗНМ розпізнаванню української мови не було достатньо великої вибірки. Однак це було вирішено за допомогою засобу Tesseract, який містить у собі НМ, навчену розпізнавати український текст, навіть деформований. Це значно спростило реалізацію проекту.

В результаті була реалізована досить точно працююча система, яка дає можливість отримати розширений склад продуктів двома шляхами: штрих-кодом і за допомогою фото складу. При пошуку по штрих-коду система використовує виключно ті дані, які містяться в БД. По зв'язках складається повний перелік компонентів із описом. У другому випадку система отримує на вхід фото складу, шукає перелік інгредієнтів серед усього результату, за допомогою функцій роботи з рядками, після чого створює масив компонентів продуктів. Після цього підключається БД. Беруться дані про всі можливі компоненти і включення імені компонента в найменування отриманого, або навпаки, виходить вся інформація про кожен з них. Після цього відбувається виведення результату.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Склад товару. URL: [https://normative\\_reference\\_dictionary.academic.ru/71704/%D0%A1%D0%BE%D1%81%D1%82%D0%B0%D0%B2\\_%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D0%B0](https://normative_reference_dictionary.academic.ru/71704/%D0%A1%D0%BE%D1%81%D1%82%D0%B0%D0%B2_%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D0%B0) (дата звернення: 20.10.2021).

2. Як правильно читати склад продуктів? URL: <http://www.vitaprom.com/blog/articles/kak-pravilno-chitat-sostav-produktov> (дата звернення: 20.10.2021).

3. Харчові добавки (E\*\*\*). Опис, позначення, класифікація та розшифрування харчових добавок. URL: <https://spravka.dobroest.com/spravochnik/sokrashheniya/pishhevaya-i-himicheskaya-promyishlennost/pishhevyie-dobavki-e-opisanie-oboznachenie-klassifikatsiya-i-rasshifrovka-pishhevyih-dobavok.html> (дата звернення : 20.10.2021).

4. Таблиця харчових E-добавок. URL: <https://dobavkam.net/additives> (дата звернення: 20.10.2021).

5. Дієтотерапія. Які хвороби можна скоригувати харчуванням? URL: [https://aif.ru/health/food/dietoterapiya\\_kakie\\_bolezni\\_mozhno\\_skorrektirovat\\_pitanie](https://aif.ru/health/food/dietoterapiya_kakie_bolezni_mozhno_skorrektirovat_pitanie) (дата звернення: 20.10.2021).

6. Вибираємо продукти для діабетиків: на що звертати увагу. URL: <http://www.zdorovka.ru/o-zdorovom-pitanii/vybiraem-produkty-dlya-diabetikov.html> (дата звернення: 20.10.2021).

7. Чи підходять діабетичні продукти для людей з діабетом? URL: <https://roscontrol.com/community/article/podhodyat-li-diabeticheskie-produkti-dlya-lyudey-s-diabetom/> (дата звернення: 20.10.2021).

8. Дієта при гастриті. Від чого треба відмовитись і чому? І які продукти варто звернути увагу. URL: <https://ivan-pole.ru/info-pole/pitanie/diety/dieta-pri-gastrite-ot-chego-nuzhno-otkazatsya-i-pochemu-i-na-kakie-produkty-stoit-obratit-vnimanie/> (дата звернення: 20.10.2021).

9. Що таке цукровий діабет і як він виникає? URL: <https://mcplus.com.ua/ru/chto-takoe-saharnyj-dyabet-y-kak-on-voznukaet/> (дата звернення: 20.10.2021).

10. Лактозна непереносимість. URL: <https://medicover.ua/ru/blog/laktoznaja-neperenosymost.html> (дата звернення: 20.10.2021).

11. Вчимося читати склад. URL: <https://vegetarian.ru/articles/uchimsya-chitat-sostav.html> (дата звернення: 20.10.2021).

12. Як правильно читати етикетки косметичних засобів. URL: <https://www.the-village.ru/service-shopping/beauty/235919-ingredients> (дата звернення: 20.10.2021).

13. Маркування продукції штрих-кодом. URL: <https://www.vostok.dp.ua/infa1/marking/mpcod/> (дата звернення: 20.10.2021).

14. Штрих-коди товарів та їх розшифровка. URL: <https://mozp.org/main/spravochnik-potrebitelya/shtrix-kod-produktov> (дата звернення: 20.10.2021).

15. Штрихкод проти QR-коду: Що краще використовувати для вашого бізнесу? URL: <https://pageloot.com/ua/barcode/%D1%88%D1%82%D1%80%D0%B8%D1%85-%D0%BA%D0%BE%D0%B4-vs-qr-%D0%BA%D0%BE%D0%B4/> (дата звернення: 20.10.2021).

16. Е-сканнер: 5 програм для визначення складу продуктів харчування і не тільки. URL: <https://zza.delo.ua/know/e-skanner-chto-prjachetsja-za-krasivoj-upakovkoj-348435/> (дата звернення: 20.10.2021).

17. Класифікація та типи нейронних мереж. URL: <http://datascientist.one/class-type-nn/> (дата звернення: 20.10.2021).

18. Використання нейронної мережі Хопфілда для вирішення найпростішого завдання. URL: <https://habr.com/ua/post/80730/> (дата звернення: 20.10.2021).

19. Що таке згортоква нейронна мережа. URL: <https://habr.com/ua/post/309508/> (дата звернення: 20.10.2021).

20. OpenCV API Reference. URL: <https://docs.opencv.org/2.4/modules/refman.html> (дата звернення: 20.10.2021).

21. Python. URL: <https://www.python.org/doc/> (дата звернення: 20.10.2021).

22. Нейронна мережа на практиці з Python та Keras. URL: <https://pythonru.com/primery/nejronnaja-set-na-praktike-s-python-i-keras> (дата звернення: 20.10.2021).

23. Розпізнавання тексту на зображенні. URL: [https://neerc.ifmo.ru/wiki/index.php?title=%D0%A0%D0%B0%D1%81%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\_%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%B0\\_%D0%BD%D0%B0\\_%D0%B8%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B8#D0.90.D0.BB.D0.B3.D0.BE.D1.80.D0.B8.D1.82.D0.BC.D1.8B\\_.D1.80.D0.B0.D1.81.D0.BF.D0.BE.D0.B7.D0.BD.D0.B0.D0.B2.D0.B0.D0.BD.D0.B8.D1.8F\\_.D1.81.D0.B8.D0.BC.D0.B2.D0.BE.D0.BB.D0.BE.D0.B2](https://neerc.ifmo.ru/wiki/index.php?title=%D0%A0%D0%B0%D1%81%D0%BF%D0%BE%D0%B7%D0%BD%D0%B0%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%B0_%D0%BD%D0%B0_%D0%B8%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B8#D0.90.D0.BB.D0.B3.D0.BE.D1.80.D0.B8.D1.82.D0.BC.D1.8B_.D1.80.D0.B0.D1.81.D0.BF.D0.BE.D0.B7.D0.BD.D0.B0.D0.B2.D0.B0.D0.BD.D0.B8.D1.8F_.D1.81.D0.B8.D0.BC.D0.B2.D0.BE.D0.BB.D0.BE.D0.B2) (дата звернення: 20.10.2021).

24. Що таке «натуральні ароматизатори»? URL: <https://vegan.ru/info/detail.php?ID=2290> (дата звернення: 20.10.2021).

24. Вибираємо базу даних. URL: <https://medium.com/nuances-of-programming/%D0%BA%D0%B0%D0%BA-%D0%B2%D1%8B%D0%B1%D1%80%D0%B0%D1%82%D1%8C-%D0%BF%D1%80%D0%B0%D0%B2%D0%B8%D0%BB%D1%8C%D0%BD%D1%83%D1%8E-%D0%B1%D0%B0%D0%B7%D1%83-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85-62b0747f7f47> (дата звернення: 20.10.2021).

25. ТОП 5 популярних систем керування базами даних (СУБД) у 2020. URL: <https://info-comp.ru/top-popular-database-management-systems> (дата звернення: 20.10.2021).

26. Найкращі IDE та редактори коду для Python. URL: <https://tproger.ru/translations/python-ide/> (дата звернення: 20.10.2021).