



## Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки  
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Боцвін Ользі Сергіївні  
(прізвище, ім'я, по батькові)1. Тема роботи Розроблення вебзастосунку для розпізнавання продуктів харчування та подальшої генерації рецептів за допомогою мультимодальних моделей

затверджена наказом університету від 19 травня 2025 року № 381Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 23 травня 2025 р.

3. Вихідні дані до роботи наукові статті та технічна документація, матеріали конференцій, дані інтернет-мережі, мультимодальні моделі Gemini 2.0 Flash та o4-mini, мова програмування Python, бібліотека для створення інтерфейсу Gradio, середовище розробки Visual Studio Code, інструмент для створення діаграм та схем Mermaid.

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

1. Аналіз існуючих вебзастосунків для пошуку рецептів за продуктами харчування.2. Детальний аналіз існуючих методів розпізнавання продуктів харчування та подальшої генерації рецептів.3. Формування методики розпізнавання продуктів харчування та подальшої генерації рецептів.4. Вибір інструментальних засобів розроблення застосунку.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми розроблення вебзастосунку для розпізнавання продуктів харчування та генерації рецептів, постановка задачі, схеми роботи моделей, тестові зображення.

---



---



---



---



---



---



---

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

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	07.04.2025	
2	Аналіз завдання, підбір літератури	08.04.25-10.04.25	
3	Аналіз літератури з досліджуваної проблеми	11.04.25-14.04.25	
4	Аналіз існуючих методів розпізнавання	15.04.25-20.04.25	
5	Формування методики	21.04.25-27.04.25	
6	Програмна реалізація	28.04.25-11.05.25	
7	Оформлення пояснювальної записки	12.05.25-20.05.25	
8	Перевірка на нормоконтроль	21.05.25-01.06.25	
9	Перевірка на плагіат	21.05.25-01.06.25	
10	Рецензування	21.05.25-01.06.25	
11	Підготовка презентації та доповіді	21.05.25-18.06.25	
12	Занесення роботи в електронний архів	02.06.25-18.06.25	
13	Попередній захист кваліфікаційної роботи	02.06.25-18.06.25	

Дата видачі завдання 7 квітня 2025 р.

Здобувач \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ доц. Творошенко І. С.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 73 с., 3 табл., 39 рис., 35 джерел.

РОЗПІЗНАВАННЯ ПРОДУКТІВ ХАРЧУВАННЯ, ГЕНЕРАЦІЯ РЕЦЕПТІВ, ВЕБЗАСТОСУНОК, ШТУЧНИЙ ІНТЕЛЕКТ, GEMINI 2.0 FLASH, O4-MINI, PYTHON.

Об'єктом роботи є розроблення вебзастосунку для розпізнавання продуктів харчування та подальшої генерації рецептів.

Метою роботи є розроблення вебзастосунку, здатного визначати продукти на фотографіях за допомогою моделі Gemini 2.0 Flash та генерувати рецепти на основі розпізнаних інгредієнтів з використанням мовної моделі o4-mini.

Сформульовано вимоги до вхідних даних, здійснено порівняння моделей розпізнавання продуктів харчування з використанням багатокритеріального аналізу та тестування. Проаналізовано логіку взаємодії між модулями розпізнавання за допомогою моделі Gemini 2.0 Flash та генерації рецептів мультимодальною мовною моделлю o4-mini.

У результаті роботи розроблено вебзастосунок, що забезпечує розпізнавання продуктів з фото, формування списку інгредієнтів та генерацію рецептів з них.

FOOD RECOGNITION, RECIPE GENERATION, WEB APPLICATION, ARTIFICIAL INTELLIGENCE, GEMINI 2.0 FLASH, O4-MINI, PYTHON.

The object of the work is the development of a web application for food recognition and subsequent recipe generation.

The aim of the work is to develop a web application capable of identifying products in photographs using the Gemini 2.0 Flash model and generating recipes based on the recognized ingredients using the o4-mini language model.

Requirements for input data were formulated, and a comparison of food recognition models was conducted using multi-criteria analysis and testing. The logic of interaction between the recognition module using the Gemini 2.0 Flash model and the recipe generation module using the multimodal o4-mini language model was analyzed.

As a result, a web application was developed that ensures the recognition of products from photos, the formation of an ingredient list, and the generation of recipes from those ingredients.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	6
Вступ.....	7
1 Аналіз існуючих вебзастосунків для пошуку рецептів за продуктами харчування .....	8
1.1 Аналіз сучасних застосунків для пошуку рецептів за продуктами харчування в Україні та за кордоном.....	8
1.2 Аналіз літературних джерел щодо апробації результатів розроблення вебзастосунків для розпізнавання продуктів харчування і генерації рецептів за допомогою мультимодальних моделей.....	16
1.3 Постановка задачі .....	22
2 Особливості розроблення вебзастосунку для розпізнавання продуктів харчування та подальшої генерації рецептів за допомогою мультимодальних моделей.....	23
2.1 Детальний аналіз існуючих методів розпізнавання продуктів харчування та подальшої генерації рецептів .....	23
2.2 Формування методики розпізнавання продуктів харчування та подальшої генерації рецептів.....	32
3 Етапи розроблення вебзастосунку для розпізнавання продуктів харчування та подальшої генерації рецептів за допомогою мультимодальних моделей ..	42
3.1 Вибір інструментальних засобів розроблення вебзастосунку для розпізнавання продуктів харчування та подальшої генерації рецептів за допомогою мультимодальних моделей .....	42
3.2 Етапи програмної реалізації розпізнавання продуктів харчування та подальшої генерації рецептів .....	44
3.3 Тестування розробленого вебзастосунку та аналіз результатів.....	62
3.4 Перспективи подальшої роботи .....	67
Висновки .....	69
Перелік джерел посилання .....	70

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

HCI – Human-Computer Interaction (взаємодія людини та комп'ютера)

ШІ – штучний інтелект

LLM – Large Language Models (великі мовні моделі)

CNN – Convolutional Neural Network (згортова нейронна мережа)

EA-CNN – Enhanced Attention-Convolutional Neural Network (покращена згортова нейронна мережа з механізмом уваги)

XAI – Explainable Artificial Intelligence (пояснюваний штучний інтелект)

Embedding – вектор дробових чисел, що містить найважливіші ознаки зображення

Example mining – вибір найбільш інформативних прикладів для навчання

CLIP – Contrastive Language-Image Pre-training (модель спільного навчання перетворення зображень і тексту в спільний векторний простір)

Zero-shot – класифікація без донавчання на цільових об'єктах

Cross-modal attention – механізм обробки взаємозв'язків між різними типами даних

JPEG – Joint Photographic Experts Group (формат стиснення растрових зображень із втратами)

Ground truth – еталонні дані для навчання й оцінки моделей

Precision – точність класифікації

Recall – повнота класифікації

F1 Score – гармонійне середнє між точністю та повнотою

JSON – формат для зберігання та передачі структурованих даних у вигляді тексту

API – Application Programming Interface (інтерфейс програмного застосунку, використовується для обміну даними між клієнтом і сервером)

## ВСТУП

Застосування штучного інтелекту в повсякденному житті стає дедалі поширенішим, особливо у сфері побутових завдань, таких як приготування їжі. З розвитком цифрових технологій зростає потреба в інтелектуальних інструментах, здатних полегшувати повсякденні процеси та економити час. Одним із таких завдань є генерація рецептів на основі лише наявних інгредієнтів – і саме тут сучасні системи ШІ можуть забезпечити суттєве спрощення процесу.

Комбінація моделей комп'ютерного зору та генеративних мовних моделей відкриває нові підходи до вирішення задачі підбору рецептів. Аналізуючи зображення продуктів, такі системи здатні автоматично визначати наявні інгредієнти, а далі формувати зв'язний і практичний опис страв, які можна з них приготувати. Це дозволяє інтегрувати інтелектуальні рішення у звичайний щоденний процес приготування їжі – від розпізнавання вмісту холодильника до отримання готового рецепта, адаптованого під конкретні продукти користувача.

Актуальність цієї роботи полягає у практичному застосуванні сучасних досягнень штучного інтелекту для вирішення повсякденних побутових завдань. Завдяки поєднанню мультимодальних моделей, таких як Gemini 2.0 Flash для розпізнавання продуктів із фото та o4-mini для генерації кулінарних рецептів, з'являється можливість створення зручного вебзастосунку, що робить процес приготування їжі більш гнучким і персоналізованим. Такий підхід демонструє потенціал штучного інтелекту в побутовому середовищі та сприяє зростанню інтересу до інтелектуальних помічників у сфері харчування.

# 1 АНАЛІЗ ІСНУЮЧИХ ВЕБЗАСТОСУНКІВ ДЛЯ ПОШУКУ РЕЦЕПТІВ ЗА ПРОДУКТАМИ ХАРЧУВАННЯ

1.1 Аналіз сучасних застосунків для пошуку рецептів за продуктами харчування в Україні та за кордоном

У сучасному світі пошук рецептів за конкретними продуктами харчування стає дедалі актуальнішим. Багато людей стикаються з проблемою вибору страви на основі обмеженого набору інгредієнтів, наявних у них вдома. Часто користувачі мають бажання приготувати щось нове, використовуючи лише те, що є в холодильнику або на кухонних полицях. Саме тому пошук рецептів за інгредієнтами стає важливою функцією, яка дозволяє максимально використовувати наявні продукти та відкривати для себе нові кулінарні можливості.

Пошук рецептів за продуктами харчування стає все більш популярним не тільки в Україні, але й у всьому світі. Це зумовлено зміною підходу до кулінарії, де головною метою стає створення страв з наявних інгредієнтів, а не просто пошук нових рецептів. Користувачі все більше цінують гнучкість і можливість адаптувати рецепти під свої потреби та наявні продукти.

З розвитком технологій з'являється все більше вебзастосунків, які пропонують користувачам зручний інтерфейс для пошуку рецептів на основі введених або обраних інгредієнтів. Такі платформи дозволяють не тільки швидко знайти рецепт, але й адаптувати його під потреби користувача, вказавши доступні продукти. Але й у кожному такому застосунку можна знайти як переваги, що спрощують процес пошуку та приготування страв, так і недоліки, що обмежують зручність або функціональність.

Одним із популярних вебзастосунків для пошуку рецептів за обраними продуктами харчування є Cookpad (рис. 1.1).

Основна мета Cookpad – можливість вибору додаткових інгредієнтів, які потрібно додати або виключити із рецепта (рис. 1.2) [1].

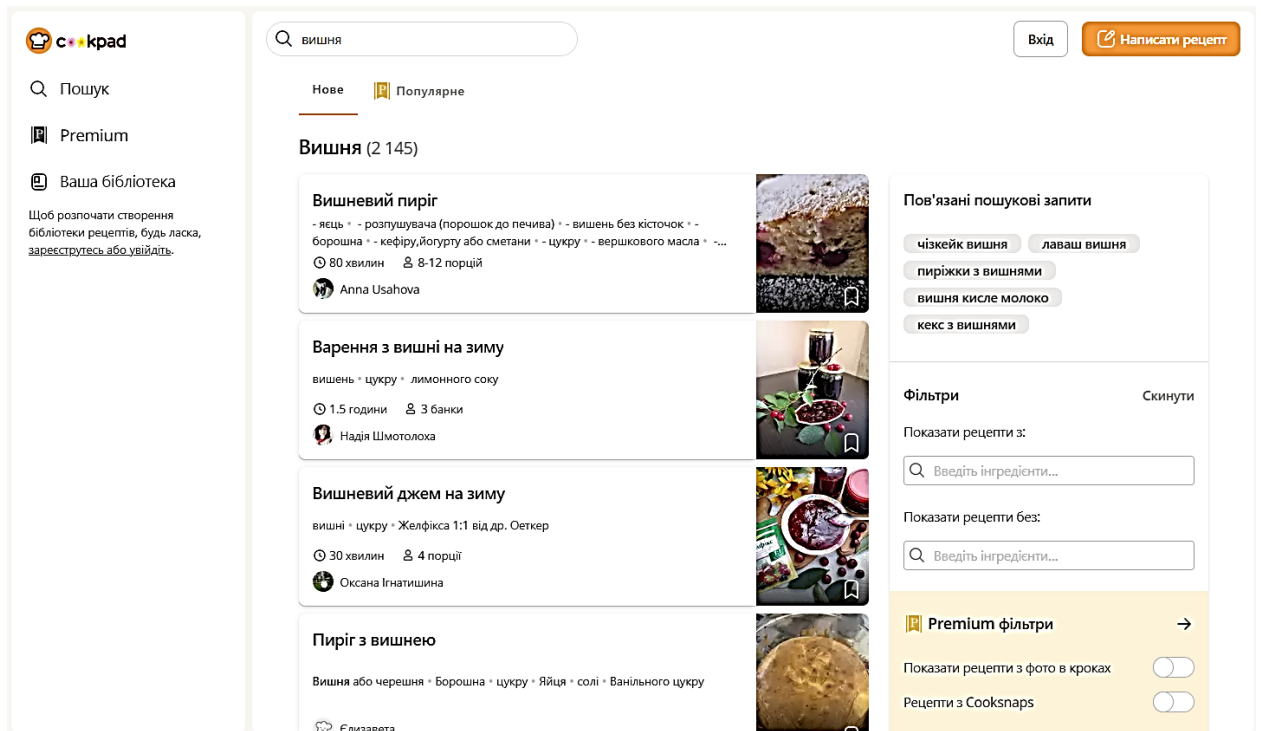


Рисунок 1.1 – Приклад пошуку рецептів за інгредієнтом у вебзастосунку Cookpad

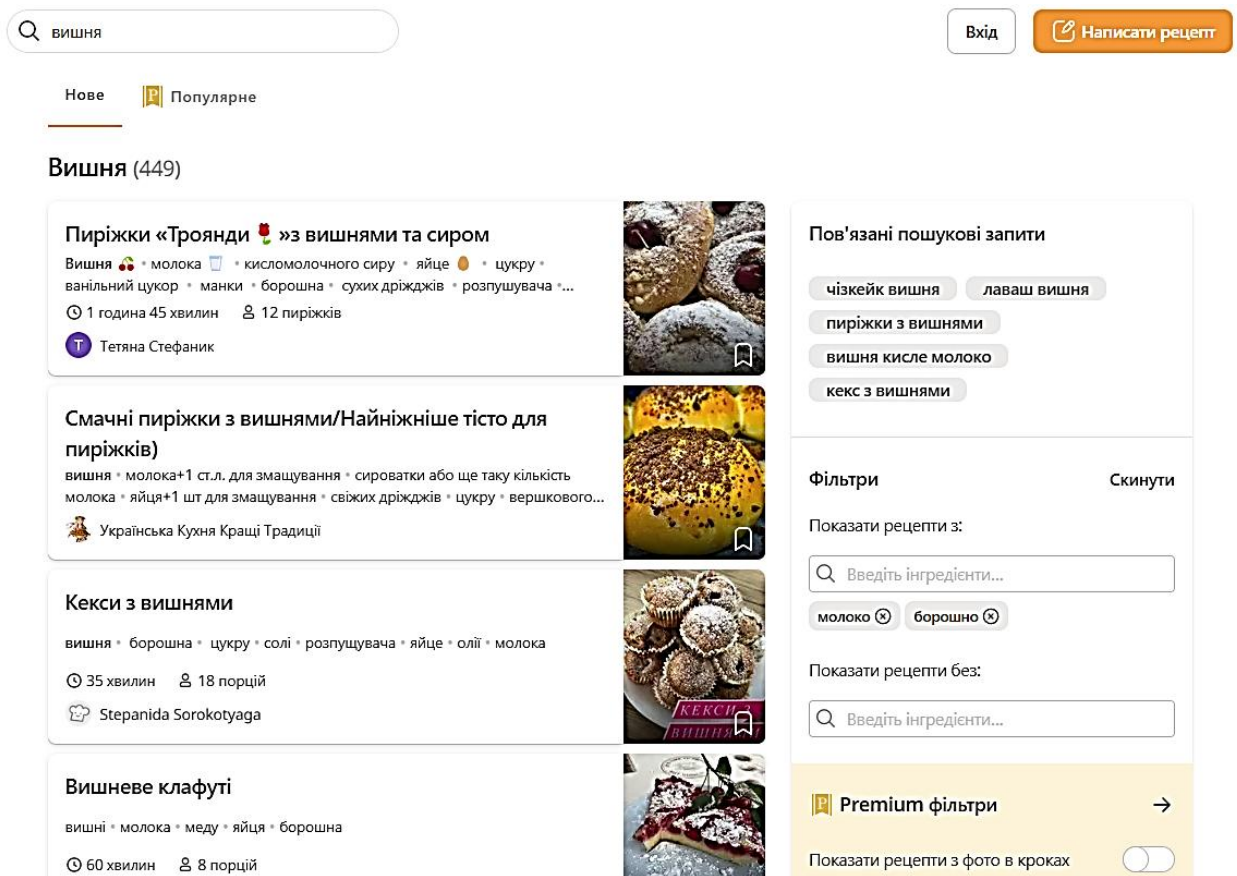


Рисунок 1.2 – Приклад пошуку рецептів з додатковими інгредієнтами

Перевагами вебзастосунку Cookpad є:

- зручний пошук рецептів за продуктами харчування з можливістю уточнення інгредієнтів;
- зручний та інтуїтивно зрозумілий інтерфейс;
- вказано час приготування та кількість порцій;
- можна побачити реальне фото готової страви перед приготуванням.

Недоліками вебзастосунку Cookpad є:

- пошук можливий лише шляхом введення головного та додаткових інгредієнтів;
- деякі функції, такі як зберігати рецепти у власній бібліотеці, доступні лише з преміум підпискою;
- незручний формат саме для приготування лише з наявних інгредієнтів.

Популярний вебзастосунок в Україні getsepty.online.ua [2] з можливістю пошуку рецепта за випадковим інгредієнтом (рис. 1.3, рис. 1.4).

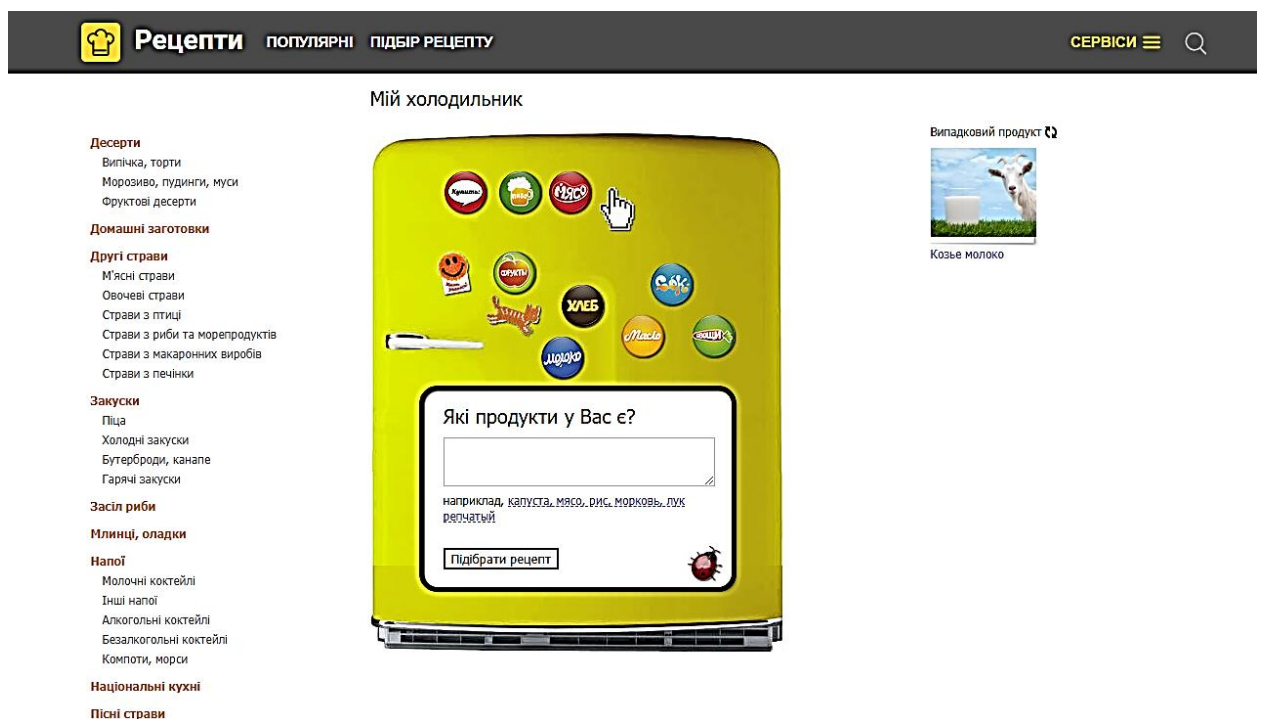


Рисунок 1.3 – Сторінка з функцією підбору рецепта

Домашні заготовки

**Другі страви**  
 М'ясні страви  
 Овочеві страви  
 Страви з птиці  
 Страви з риби та морепродуктів  
 Страви з макаронних виробів  
 Страви з печінки

**Закуски**  
 Піца  
 Холодні закуски  
 Бутерброди, канапе  
 Гарячі закуски

**Засіл риби**

**Млинці, оладки**

**Напої**  
 Молочні коктейлі  
 Інші напої  
 Алкогольні коктейлі  
 Безалкогольні коктейлі  
 Компоти, морси

**Національні кухні**

**Пісні страви**

**Перші страви**  
 Супи  
 Борщі  
 Холодні супи

**Рецепти з фотографіями**

**Салати**  
 М'ясні салати  
 Салати з риби та морепродуктів  
 Салати складні  
 Овочеві салати  
 Грибні салати  
 Фруктові салати

**Соуси, підливи**

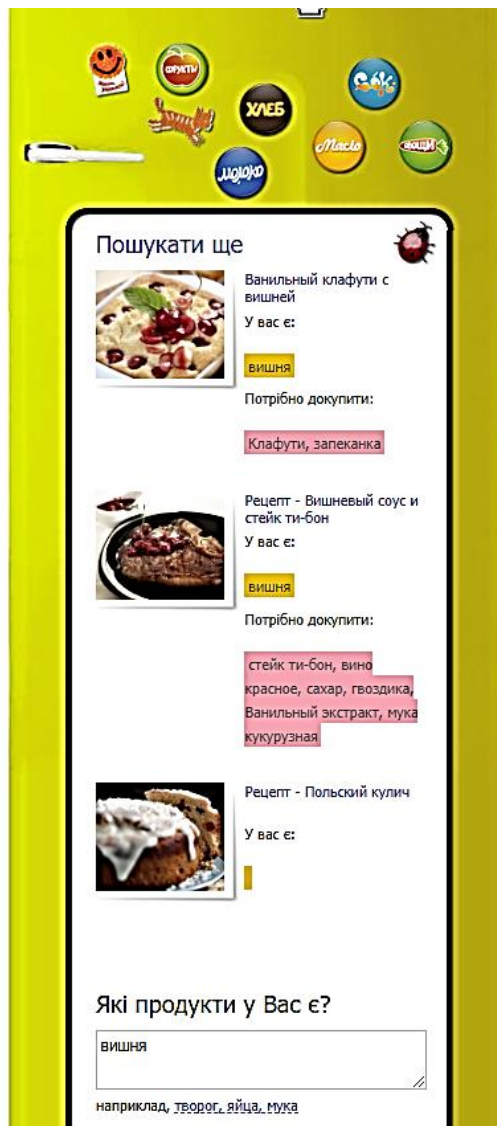
**Продукти**

**Етикет**

**Секрети**

**Посуд**

**Харчові добавки E**



Сыр Пармезан

Рисунок 1.4 – Приклад пошуку рецептів за інгредієнтом у вебзастосунку retsepty.online.ua

Перевагами вебзастосунку retsepty.online.ua є:

- цікавий інтерфейс у вигляді холодильника;
- можна обрати пошук рецепта за випадковим продуктом, що може урізноманітнити меню;

- пропонує готовий список різних продуктів для пошуку рецептів;
- до кожного рецепта є список інгредієнтів, які потрібно докупити.

Недоліками вебзастосунку retsepty.online.ua є:

- пошук можливий лише шляхом введення інгредієнтів;
- не підходить під принцип приготування лише з наявних продуктів;

- іноді видає рецепти не пов'язані з введеними інгредієнтами;
- якщо рецептів забагато, перегляд їх у такому форматі може бути незручним.

Популярним за кордоном є вебзастосунок My Fridge Food (рис. 1.5) [3], він відрізняється від аналогів методом вибору інгредієнтів для пошуку рецептів (рис. 1.6).

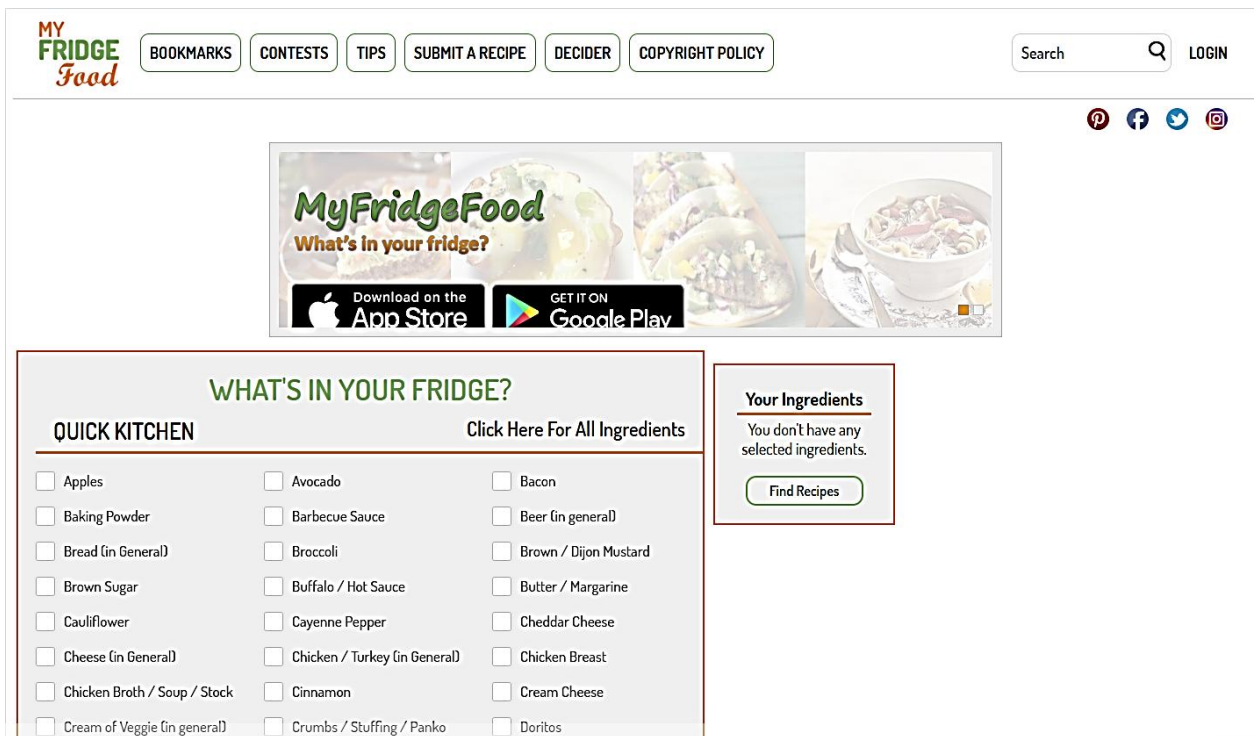


Рисунок 1.5 – Головна сторінка вебзастосунку My Fridge Food

Перевагами вебзастосунку My Fridge Food є:

- можливість обрати стислий або ж повний список інгредієнтів;
- у повному списку інгредієнти відсортовані по категоріям;
- можна відфільтрувати знайдені рецепти за різними категоріями;
- до кожного рецепта, якщо потрібно, є список інгредієнтів, які потрібно докупити;
- до кожного рецепта розраховано приблизний час на приготування (рис. 1.7).

PASTA / GRAINS		
<input type="checkbox"/> Angel Hair	<input type="checkbox"/> Barley	<input type="checkbox"/> Black Rice
<input type="checkbox"/> Bowtie Pasta	<input type="checkbox"/> Brown Rice	<input type="checkbox"/> Egg Noodles
<input type="checkbox"/> Elbow Macaroni	<input type="checkbox"/> Fettuccine Noodles	<input type="checkbox"/> Gnocchi
<input type="checkbox"/> Granola	<input type="checkbox"/> Lasagna Noodles	<input type="checkbox"/> Mac and Cheese / Mix
<input type="checkbox"/> Oats / Oatmeal	<input type="checkbox"/> Orzo	<input type="checkbox"/> Pasta Noodles (in General)
<input type="checkbox"/> Ramen	<input type="checkbox"/> Rice (in General)	<input type="checkbox"/> Rigatoni
<input type="checkbox"/> Rotini Noodles	<input type="checkbox"/> Shell Noodles	<input type="checkbox"/> Spaghetti Noodles
<input type="checkbox"/> Tortellini	<input type="checkbox"/> White Rice	<input type="checkbox"/> Wild Rice
<input type="checkbox"/> Ravioli		

PRODUCE		
<input type="checkbox"/> Apples	<input type="checkbox"/> Artichokes	<input type="checkbox"/> Asparagus
<input type="checkbox"/> Avocado	<input type="checkbox"/> Banana	<input type="checkbox"/> Beans
<input type="checkbox"/> Black Olives	<input type="checkbox"/> Blueberries	<input type="checkbox"/> Broccoli
<input type="checkbox"/> Brussel Sprouts	<input type="checkbox"/> Butternut Squash	<input type="checkbox"/> Cabbage
<input type="checkbox"/> Carrot	<input type="checkbox"/> Cauliflower	<input type="checkbox"/> Celery
<input checked="" type="checkbox"/> Cherries	<input type="checkbox"/> Cherry Tomatoes	<input type="checkbox"/> Chilis
<input type="checkbox"/> Coconut	<input type="checkbox"/> Corn	<input type="checkbox"/> Cranberries

**Your Ingredients**





[Clear All](#)

Cherries

[Find Recipes](#)

Рисунок 1.6 – Приклад пошуку рецептів за інгредієнтом у вебзастосунку My Fridge Food

## RECIPE RESULTS

<div style="text-align: center;"> <p>No Picture of this Recipe... YET</p> <p><b>To Win a Prize!!</b></p> <p>If you make this recipe:</p> <ol style="list-style-type: none"> <li>1. Take a Picture</li> <li>2. Post it on Twitter or Instagram with - the Recipe Title</li> <li>3. That's It! You're entered to win Prizes!</li> </ol> </div> <p><b>Basic Bakers</b></p> <p>Cooking Time: 10 minutes</p> <p>No Missing Ingredients!</p> <p style="text-align: center;"><a href="#">Log in to bookmark!</a></p>	<div style="text-align: center;"> <p>No Picture of this Recipe... YET</p> <p><b>To Win a Prize!!</b></p> <p>If you make this recipe:</p> <ol style="list-style-type: none"> <li>1. Take a Picture</li> <li>2. Post it on Twitter or Instagram with - the Recipe Title</li> <li>3. That's It! You're entered to win Prizes!</li> </ol> </div> <p><b>Quick Scratch Hash Browns</b></p> <p>Cooking Time: 10 minutes</p> <p>No Missing Ingredients!</p> <p style="text-align: center;"><a href="#">Log in to bookmark!</a></p>	 <p><b>Oven Baked Fried Fries</b></p> <p>Cooking Time: 45 minutes</p> <p>Missing Ingredients</p> <p style="text-align: center;"><a href="#">Add</a> Shortening / Oil (General)</p> <p style="text-align: right;"><a href="#">Log in to bookmark!</a></p>
 <p><b>Scored Potatoes</b></p> <p>Cooking Time: 50 minutes</p> <p>Missing Ingredients</p> <p style="text-align: center;"><a href="#">Add</a> Butter / Margarine</p> <p style="text-align: right;"><a href="#">Log in to bookmark!</a></p>	 <p><b>Homemade Fries</b></p> <p>Cooking Time: 30 minutes</p> <p>Missing Ingredients</p> <p style="text-align: center;"><a href="#">Add</a> Olive Oil</p> <p style="text-align: right;"><a href="#">Log in to bookmark!</a></p>	 <p><b>Mashed Potato Muffins</b></p> <p>Cooking Time: 15 minutes</p> <p>Missing Ingredients</p> <p style="text-align: center;"><a href="#">Add</a> Butter / Margarine</p> <p style="text-align: right;"><a href="#">Log in to bookmark!</a></p>

**Filter Results +**

**By Ingredients**

Beef (in General)

Potatoes

**Categories**

Salads and Sides

Fried Food

Appetizer

Unapproved Recipes

**Calories**

Any ▼

[Filter Recipes](#)

[Can't decide?](#)

Рисунок 1.7 – Результат пошуку рецептів за інгредієнтами

Недоліками вебзастосунку My Fridge Food є:

- інтерфейс не зовсім зручний та зрозумілий для користувача;
- пошук можливий лише шляхом вибору інгредієнтів зі списку;
- пошук необхідних інгредієнтів займає багато часу;
- мала база рецептів;
- до деяких рецептів відсутнє фото.

Також доволі популярний вебзастосунок Supercook (рис. 1.8) [4], який відрізняється від інших своїм зручним функціоналом.

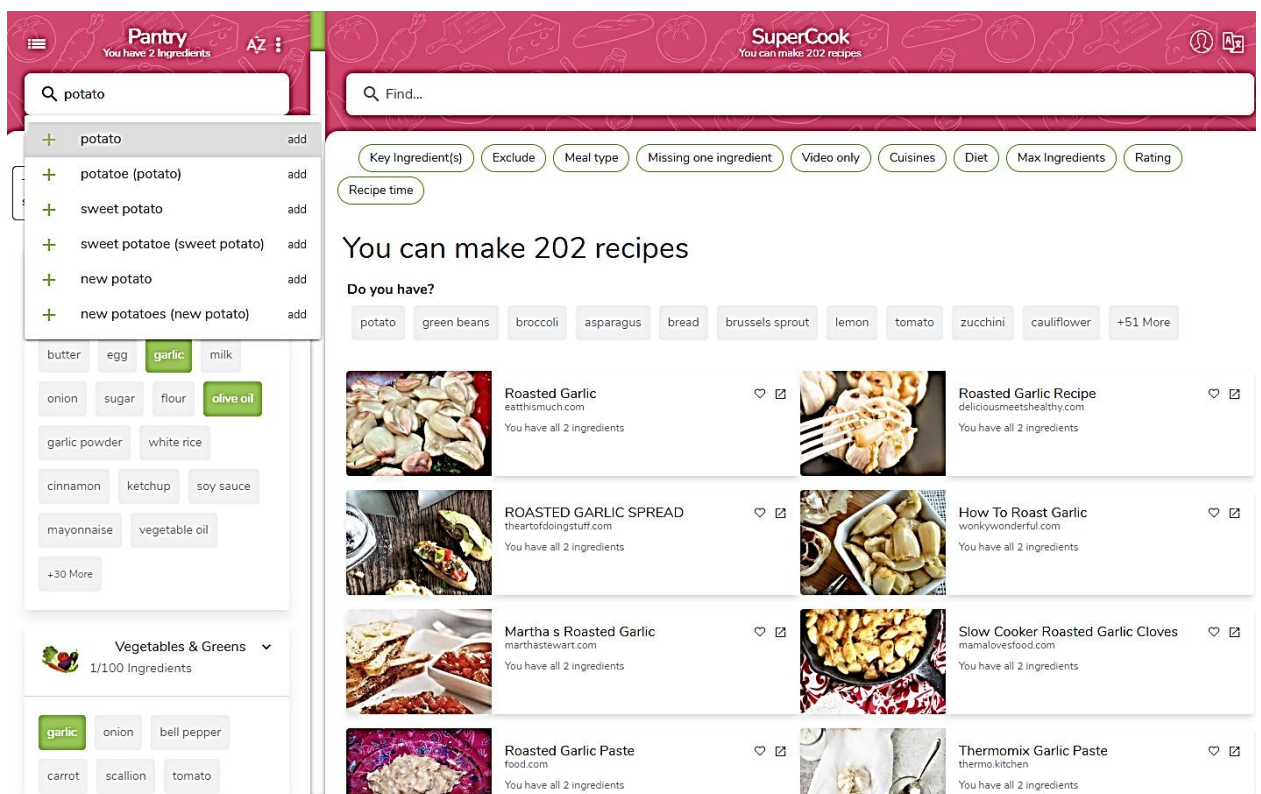


Рисунок 1.8 – Приклад вибору інгредієнтів у вебзастосунку Supercook

Перевагами вебзастосунку Supercook є:

- зручний та інтуїтивно зрозумілий інтерфейс;
- можливість пошуку інгредієнтів шляхом вводу інгредієнтів або вибору зі списку;
- інгредієнти відсортовані по категоріям (рис. 1.9);
- можна відфільтрувати знайдені рецепти за різними категоріями;

- до кожного рецепта є фото (рис. 1.10);
- є список інгредієнтів, яких не вистачає, якщо потрібно;
- велика база рецептів.

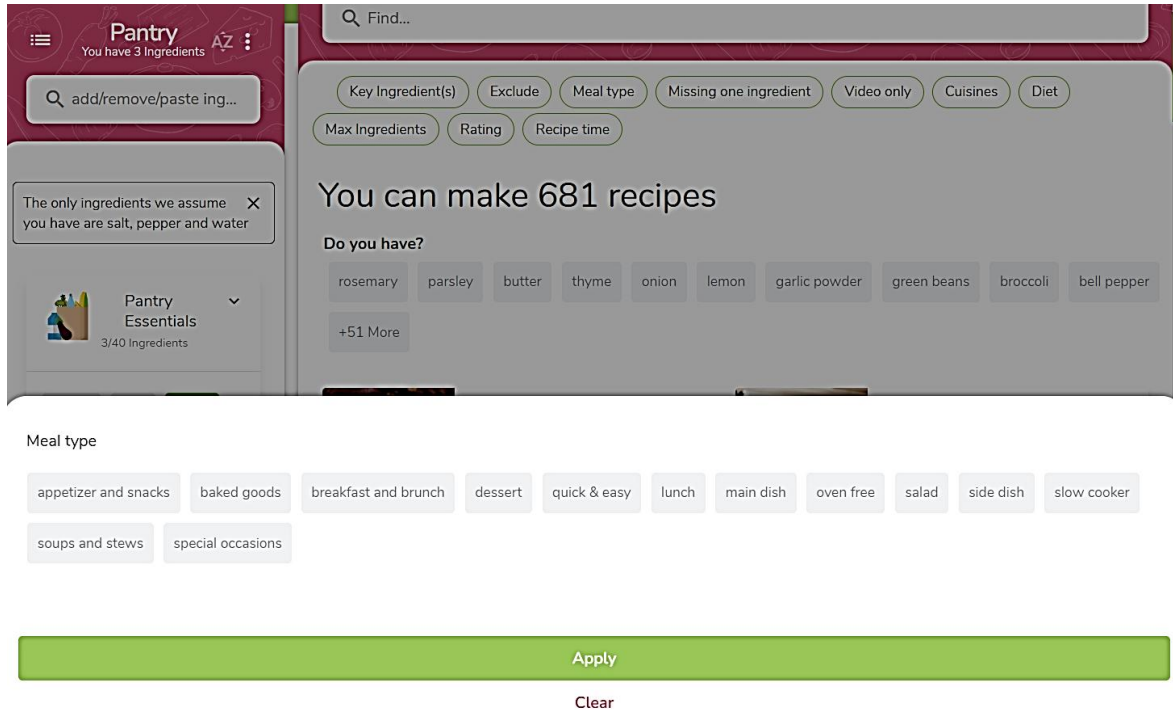


Рисунок 1.9 – Приклад вибору категорій для пошуку рецептів

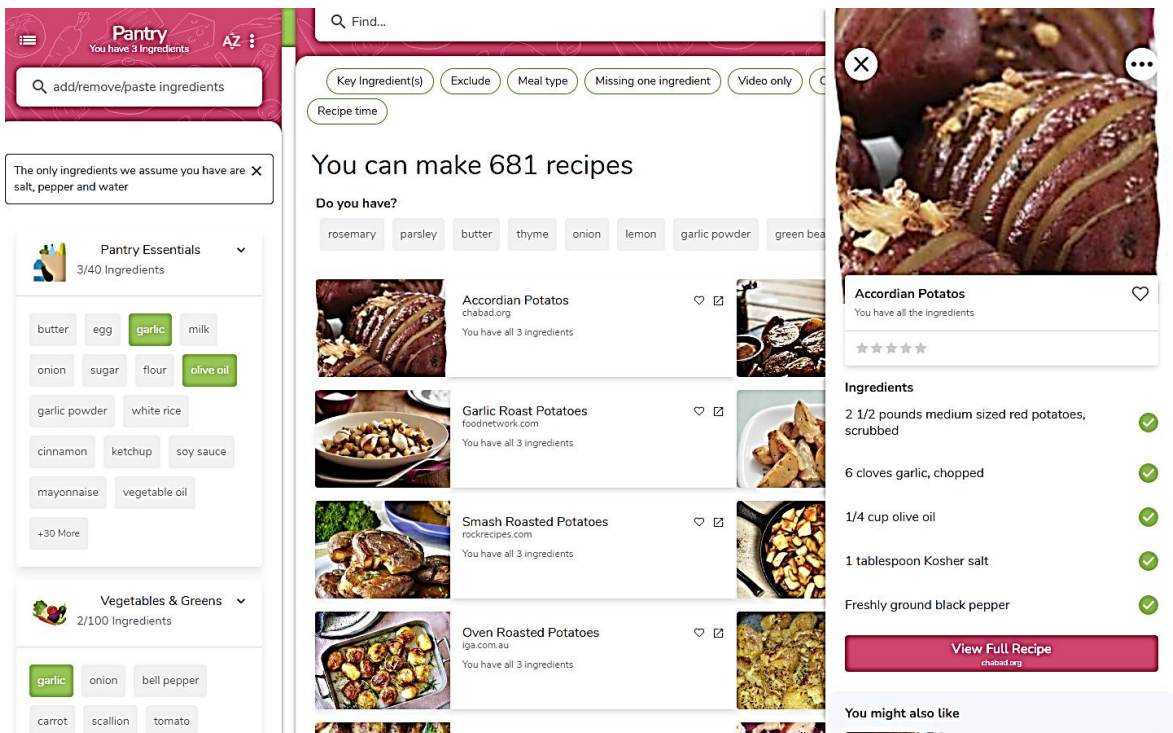


Рисунок 1.10 – Приклад відображення знайденого рецепта

Недоліками вебзастосунку Supercook є:

- щоб подивитися повний рецепт, потрібно переходити на сторонні сайти;
- пошук необхідних інгредієнтів може займати час.

У результаті аналізу встановлено, що вебзастосунки [1–4] мають суттєвий недолік – пошук рецептів відбувається за співпадінням із списком інгредієнтів, таким чином, в деяких рецептах враховуються складові, які не були обрані користувачем. Через це виникає проблема, що потрібні додаткові інгредієнти для приготування тієї чи іншої страви. Також ці вебзастосунки мають недолік вибору інгредієнтів для пошуку (ручний ввід або обрання зі списку), що займає багато часу та може бути незручним для користувача.

1.2 Аналіз літературних джерел щодо апробації результатів розроблення вебзастосунків для розпізнавання продуктів харчування і генерації рецептів за допомогою мультимодальних моделей

У роботі [5] автори підкреслюють важливість харчування для здоров'я та добробуту людини, особливо в сучасному світі, де все більше поширюються захворювання, пов'язані з неправильним харчуванням. З огляду на це, все більшої популярності набувають мобільні застосунки для відстеження харчування, які потребують ефективних систем класифікації продуктів. Для вирішення цієї проблеми автори пропонують модель розпізнавання їжі на основі глибоких згорткових нейронних мереж з механізмом самонавчання. Модель була навчена на різних структурах і застосовувала регуляризацію для уникнення надмірної складності. Її протестували на двох нових наборах даних: Food-101 та MA Food-121, де вона показала високу точність: 96,40 % і 95,11 % відповідно. Модель перевершує базові моделі навчання з перенесенням на 8,12 % та демонструє

здатність до узагальнення на випадкових зображеннях з Інтернету, що робить її ефективною для задач класифікації продуктів.

У роботі [6] автори описують систему рекомендацій рецептів Chef Dalle, що використовує мультимодельні та мультимодальні методи НСІ для надання персоналізованих кулінарних порад. У сучасному світі, де харчові звички значно впливають на здоров'я, Chef Dalle пропонує технологічне рішення для вибору страв, що відповідають індивідуальним потребам. Система інтегрує кілька сучасних технологій, зокрема перетворення голосу в текст через Whisper та розпізнавання зображень інгредієнтів за допомогою GPT-Vision. Це дозволяє користувачам вводити інгредієнти голосом або фотографіями для отримання відповідних рецептів. Використання таких моделей, як OpenAI, Google Gemini, Claude та Anthropic API, забезпечує точність і врахування дієтичних обмежень. Крім того, Chef Dalle генерує зображення готових страв через DALL-E 3, що підвищує залученість користувачів. Система використовує зворотний зв'язок для постійного вдосконалення рекомендацій, демонструючи свою адаптивність. Chef Dalle має широкий спектр застосувань – від домашніх кухонь до продуктових магазинів і ресторанів, сприяючи популяризації здорового харчування. Таким чином, автори підкреслюють важливість мультимодальних методів НСІ для поліпшення кулінарного досвіду та створення інноваційних рішень у цій сфері.

У роботі [7] автори досліджують новий підхід до автоматизованої оцінки раціону харчування на основі мультимодальної моделі GPT-4V, що демонструє суттєве поліпшення у порівнянні з традиційними методами самозвітів або інтерв'ю під наглядом дієтологів. Традиційні підходи є суб'єктивними, часом неточними і потребують значних ресурсів. Попередні рішення на основі штучного інтелекту (ШІ) вирішували завдання лише частково, наприклад, зосереджуючись на розпізнаванні окремих продуктів або визначенні розмірів порцій. Проте ці методи мали обмежену здатність узагальнювати різноманітні категорії продуктів, кулінарні звички та

враховувати культурні особливості. Впровадження мультимодальних моделей, таких як GPT-4V, показало нові можливості в розв'язанні складних завдань, таких як розуміння сцен і створення підписів до зображень. У цьому дослідженні автори застосовують GPT-4V для оцінки раціону харчування, використовуючи швидке інжиніринг-підказок і методи пасивного моніторингу. Як набір даних було зібрано 16 епізодів прийому їжі в реальному житті, зафіксованих за допомогою натільних камер. Результати експериментів демонструють, що GPT-4V ефективно розпізнає їжу в складних умовах без додаткового навчання на спеціальних наборах даних про їжу. Модель показала здатність адаптуватися до різних кулінарних контекстів, наприклад, розпізнаючи не лише загальні продукти, такі як рис і хліб, але й регіональні страви, такі як банку та угалі, за допомогою специфічних мовних підказок. GPT-4V також демонструє контекстну обізнаність, використовуючи навколишні об'єкти для визначення розмірів порцій, що полегшує процес оцінки харчування. Це дослідження показує потенціал GPT-4V для автоматизованого моніторингу харчування, що може стати важливим інструментом у сфері дієтології та харчування.

У цьому дослідженні [8] йдеться про використання ШІ як інструменту для створення нових кулінарних рецептів. Метою було дослідити, наскільки ефективно ШІ може допомагати професійним шеф-кухарям у генерації нових ідей та створенні нестандартних поєднань інгредієнтів. Дослідники вирішили перевірити, чи може ШІ створити рецепти з незвичними інгредієнтами – полуницею, куркою та виноградним листям – без додаткових вказівок. ШІ згенерував три різні рецепти з використанням цих інгредієнтів. Щоб оцінити, наскільки споживачі сприймають створені страви, провели так званий «гедоністичний тест», де 118 непідготовлених експертів пробували ці страви та виставляли оцінки. Результати показали, що всі рецепти отримали прийнятні оцінки (середній бал  $\geq 6,0$  з 9 можливих), причому найнижча оцінка була  $6,58 \pm 1,61$ . Це дослідження демонструє, що ШІ може ефективно підтримувати професійних шеф-кухарів у скороченні процесу творчого

пошуку нових кулінарних рішень, а також створювати оригінальні й несподівані поєднання інгредієнтів. Дослідження є новаторським у вимірюванні здатності ШІ до кулінарної творчості, а також показує, як споживачі сприймають страви, створені за допомогою штучного інтелекту, що відкриває нові можливості для інновацій у кулінарному мистецтві.

У дослідженні [9] автори представляють новий підхід до надання рекомендацій щодо харчування на основі штучного інтелекту – фреймворк ChatDiet, який орієнтований на забезпечення персоналізованих, пояснюваних і інтерактивних рекомендацій. Традиційні методи часто не враховують індивідуальні особливості харчування, не забезпечуючи достатньої гнучкості та інтерпретації, тому цей фреймворк розроблений для подолання цих обмежень. ChatDiet використовує LLM для створення інтерпретованих рекомендацій, однак їх автономне використання не дає можливості досягти справжньої персоналізації. Тому фреймворк поєднує персональну та популяційну моделі, які доповнюються оркестратором для більш комплексної обробки даних. Персональна модель застосовує методи виявлення причинно-наслідкових зв'язків для оцінки ефектів харчування, специфічних для кожного користувача. У той час популяційна модель надає загальну інформацію про поживну цінність продуктів харчування на основі узагальнених даних. Оркестратор об'єднує результати обох моделей, передаючи їх до LLM, щоб створити індивідуальні рекомендації з урахуванням потреб користувача. ChatDiet вирізняється своєю здатністю надавати динамічні рекомендації щодо харчування, які не тільки адаптуються до індивідуальних вподобань користувача, але й зрозуміло пояснюють свої рекомендації. Ефективність ChatDiet було продемонстровано за допомогою тематичного дослідження, в якому розробили персоналізовану модель для оцінки впливу харчування на конкретну людину. Оцінка ефективності рекомендаційного механізму показала результат у 92%, підтверджуючи, що система здатна надавати точні та персоналізовані поради. Ілюстративні приклади діалогів демонструють високу здатність ChatDiet до пояснення та

інтерактивності, що робить його ефективним інструментом для підтримки здорового харчування.

У роботі [10] автори представили систему NutriFlow, призначену для створення персоналізованих планів харчування, орієнтованих на підтримку здоров'я користувачів та врахування їхніх індивідуальних потреб. Сучасний спосіб життя ускладнює підтримку фізичної форми, особливо для людей, які страждають від хронічних захворювань, таких як серцеві хвороби чи інші медичні проблеми. NutriFlow пропонує рішення, використовуючи передові методи збору даних про фізичний стан і медичну історію користувача для створення відповідних дієтичних планів. В основі системи лежить потужний алгоритм ChatGPT 3.5, який забезпечує персоналізовану взаємодію з користувачем і надає індивідуальні рекомендації щодо харчування. NutriFlow не лише генерує плани харчування, але й пропонує комплексну підтримку, включаючи вправи з йоги та рецепти, які відповідають рекомендованим продуктам, забезпечуючи більш інтегрований підхід до здорового способу життя. Додатково система має модуль для відстеження рівня гідратації, який допомагає користувачам контролювати споживання води, підтримуючи оптимальний баланс рідини в організмі. Завдяки цьому NutriFlow сприяє не лише правильному харчуванню, але й загальному фізичному добробуту. Таким чином, NutriFlow є ефективною системою для створення індивідуальних планів харчування, які відповідають фізичному стану та цілям користувача, допомагаючи вести здоровий та збалансований спосіб життя.

У роботі [11] автори представляють Smart Cuisine – інноваційну систему, що використовує передові технології штучного інтелекту для оптимізації процесу приготування їжі. Smart Cuisine пропонує низку унікальних функцій, спрямованих на покращення кулінарних процесів, включаючи розрахунок пропорцій інгредієнтів, створення нових рецептів за допомогою ШІ та інтерактивний чат для обговорення питань харчування та здоров'я. Система використовує генератор рецептів на основі мета-моделі

Facebook, що робить її доступною для широкого кола користувачів. Цей генератор дозволяє створювати смачні та збалансовані рецепти, використовуючи ті інгредієнти, що є під рукою, мінімізуючи харчові відходи та сприяючи сталому використанню ресурсів. Додаткова функція чату, розроблена на основі OpenAI API та Flutterflow, надає користувачам можливість ставити запитання на природній мові про їжу, дієти та здоров'я. Smart Cuisine також забезпечує користувачів точними розрахунками пропорцій інгредієнтів, що сприяє приготуванню страв із найменшою кількістю відходів та економічному використанню продуктів. Автори підкреслюють важливість цієї системи для підтримки сталого розвитку, допомагаючи користувачам максимально ефективно використовувати наявні продукти. Завдяки поєднанню технологій штучного інтелекту, розрахунку пропорцій та інтерактивного чату, Smart Cuisine є потужним інструментом для тих, хто прагне зменшити харчові відходи, підтримувати здорове харчування та сприяти екологічно свідомому приготуванню їжі. Система має потенціал для значного впливу на харчову культуру, роблячи її більш ефективною, здоровою та стійкою. Дослідження описує процес розробки Smart Cuisine, підкреслюючи її значення як нового етапу в управлінні продуктами харчування на основі штучного інтелекту, що сприяє здоров'ю людей та збереженню навколишнього середовища.

Таким чином, проаналізувавши джерела [5–11], зроблено висновок, що сучасні вебзастосунки для розпізнавання їжі та створення рецептів активно використовують штучний інтелект. Вони допомагають точно визначати продукти на фото, пропонують індивідуальні поради щодо харчування та навіть вигадують нові страви. У таких системах, як Chef Dalle, ChatDiet, NutriFlow і Smart Cuisine, користувач може легко спілкуватися з програмою – наприклад, поставити запитання або отримати зрозуміле пояснення, чому запропоновано саме той рецепт чи пораду. Це робить дані застосунки зручними, корисними в повсякденному житті та корисними як для звичайних людей, так і для кухарів чи дієтологів.

Усе це підтверджує, що GPT-Vision та подібні технології мають великий потенціал у сфері здорового харчування та кулінарії.

### 1.3 Постановка задачі

Автоматичне розпізнавання продуктів харчування на зображеннях та генерація рецептів є актуальним завданням у контексті розвитку мультимодальних систем штучного інтелекту. Тому ставиться завдання створення вебзастосунку, що дозволяє розпізнавати продукти на фото та формувати рецепти страв із використанням сучасних генеративних моделей.

Об'єктом роботи є розроблення вебзастосунку для розпізнавання продуктів харчування та подальшої генерації рецептів.

Метою роботи є розроблення вебзастосунку, здатного визначати продукти на фотографіях за допомогою моделі Gemini 2.0 Flash та генерувати рецепти на основі розпізнаних інгредієнтів з використанням мовної моделі o4-mini.

Для досягнення мети необхідно вирішити такі завдання:

- проаналізувати сучасні підходи до розпізнавання продуктів харчування;
- сформулювати вимоги до вхідних даних і критерії вибору моделей;
- порівняти доступні моделі розпізнавання за допомогою багатокритеріального аналізу та тестування;
- дослідити можливості o4-mini для генерації рецептів;
- розробити вебзастосунок, що поєднує модуль розпізнавання зображень із модулем генерації рецептів;
- протестувати працездатність системи на прикладах та оцінити її ефективність.

## **2 ОСОБЛИВОСТІ РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКУ ДЛЯ РОЗПІЗНАВАННЯ ПРОДУКТІВ ХАРЧУВАННЯ ТА ПОДАЛЬШОЇ ГЕНЕРАЦІЇ РЕЦЕПТІВ ЗА ДОПОМОГОЮ МУЛЬТИМОДАЛЬНИХ МОДЕЛЕЙ**

### 2.1 Детальний аналіз існуючих методів розпізнавання продуктів харчування та подальшої генерації рецептів

Точне розпізнавання продуктів харчування на фотографії є ключовим етапом у конвеєрі «зображення → список інгредієнтів → рецепт». Якщо модель буде неправильно визначати продукти, то згенерований рецепт може виявитися недоречним або нездійсненним через відсутність таких інгредієнтів, що призведе до втрати часу користувача. Саме тому в системах розпізнавання, що в подальшому покладаються на ChatGPT для генерації рецептів, надзвичайно важливо застосувати методи, які здатні одночасно виявляти десятки різноманітних продуктів у різних умовах (освітлення, фон, ракурс) із високою точністю й у реальному часі. У контексті вебзастосунків для повсякденного використання це особливо актуально, адже користувачі очікують швидкої та точної відповіді. Якісне розпізнавання продуктів харчування напряду впливає на рецепт, тому саме цей етап визначає загальну ефективність усього проєкту.

Метод класифікації зображень за допомогою звичайних CNN полягає в навчанні згорткової нейронної мережі на зображеннях окремих продуктів на однотонному фоні. Архітектура складається з послідовності конволюційних шарів із операціями підвибірки після яких ідуть повнозв'язні шари, що перетворюють витягнуті ознаки на ймовірності належності до певних класів (рис. 2.1) [12].

У дослідженні [13] представлено набір даних Fruits-360, що містить 90483 зображення 131 класу фруктів, овочів та горіхів. Вони показали, що базова CNN досягає 98,66 % точності на тесті, що є відмінним результатом.

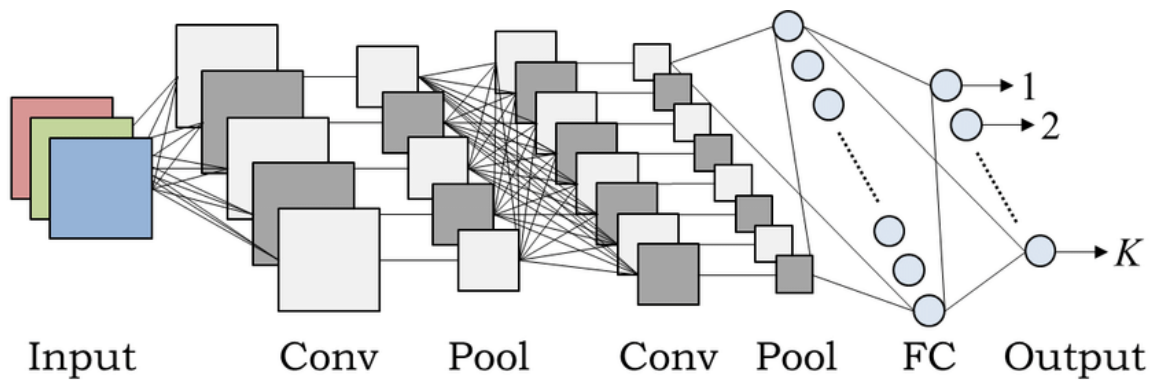


Рисунок 2.1 – Приклад архітектури CNN

Метод Enhanced Attention-CNN (EA-CNN) з Explainable AI (XAI) полягає в тому, що після стандартних згорткових блоків із операціями підвибірки мережа застосовує механізм уваги для зважування найбільш інформативних регіонів зображення. Далі вбудований модуль XAI генерує теплові карти важливості пікселів, що забезпечує точність прийнятого рішення (рис. 2.2) [14].

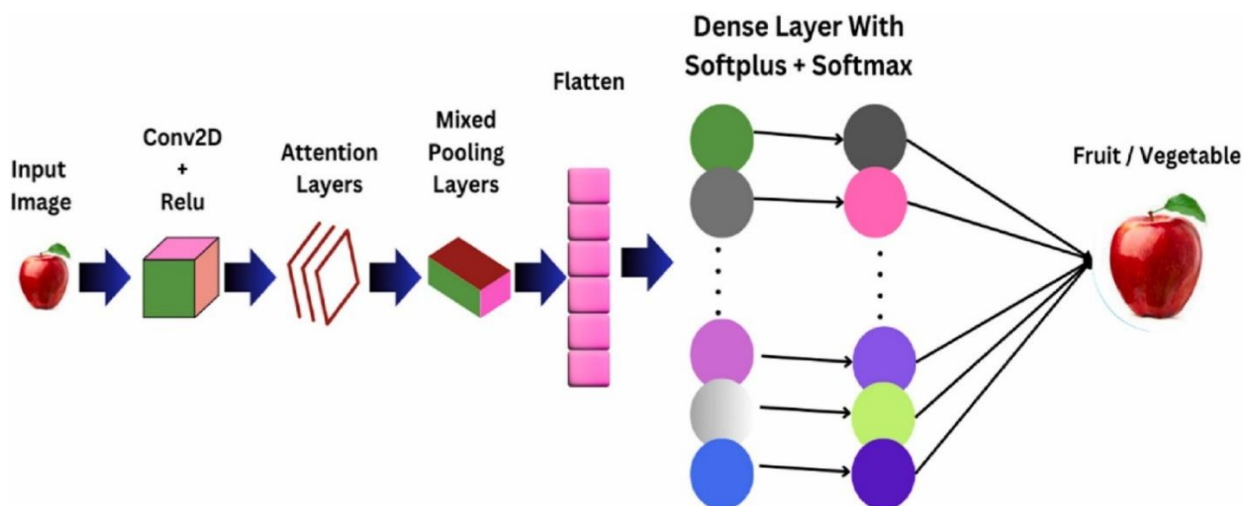


Рисунок 2.2 – Загальна схема архітектури запропонованої нейронної мережі Enhanced Attention-Convolutional Neural Network (EA-CNN)

Згідно з дослідженням [14] на публічному наборі даних Fruits-360 модель EA-CNN показала 98,1 % точності класифікації, а на наборі Fruit Recognition – 96 %, що доводить ефективність даного методу.

Метод трансферного навчання на основі згорткової нейронної мережі Inception v3 полягає у повторному використанні знань, зданих моделлю під час навчання на великому узагальненому наборі даних ImageNet, для розв'язання нових, вузких завдань класифікації.

Модель уже навчена розпізнавати загальні ознаки об'єктів – контури, кольори, форми, текстури – і тому не потребує повного навчання з нуля. У процесі адаптації для розв'язання задачі її вихідний класифікаційний шар замінюють на новий, що відповідає конкретним категоріям, після чого роблять тонке налаштування моделі на новій вибірці (рис. 2.3) [15].

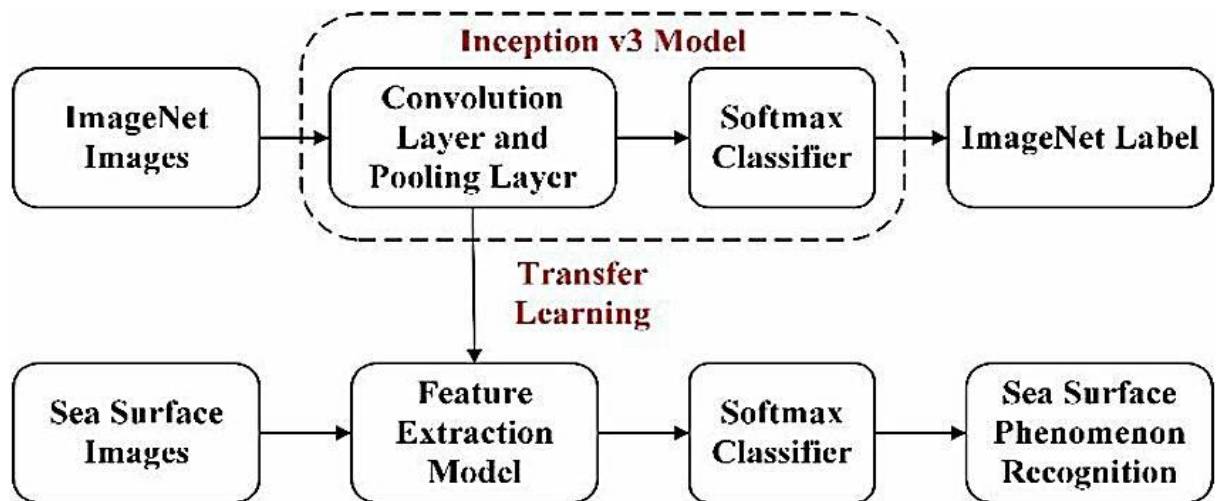


Рисунок 2.3 – Архітектура трансферного навчання на основі моделі Inception v3

Це значно зменшує обсяги потрібних даних і часу на навчання, зберігаючи високу якість розпізнавання. Inception v3, як глибока CNN, поєднує ефективність із високою точністю та добре масштабується під різні візуальні задачі.

У дослідженні [16] автори запропонували архітектуру, в якій старший класифікатор на базі Inception v3 пересилає кожне вхідне зображення далі до трьох із десяти спеціалізованих CNN-класифікаторів.

Вони використали 1920 зображень продуктів і товарів із супермаркетів для навчання алгоритму за 131 класом.

Під час тестування помилка для топ-1 прогнозу становила лише кілька відсотків, частка хибно-позитивних відповідей серед трьох найімовірніших варіантів – 25,36 %, а межа похибки для топ-5 – 20,28 %. Результати тестування підтверджують ефективність використання методу трансферного навчання на основі CNN Inception v3 для розпізнавання продуктів харчування.

Метод Instance segmentation з embedding-класифікатором полягає в розпізнаванні продуктів харчування на зображенні поетапно.

На першому етапі використовується сегментація екземплярів – підхід комп’ютерного зору, що дозволяє виділити контурні маски кожного окремого об’єкта на зображенні. Далі виділені об’єкти локалізуються як окремі області інтересу.

Кожна така область передається в embedding-модель – нейронну мережу, яка перетворює зображення об’єкта на вектор ознак для подальшої класифікації. Ця модель навчається зі стратегією example mining, яка автоматично добирає найбільш інформативні зразки для підвищення якості класифікації (рис. 2.4) [17].

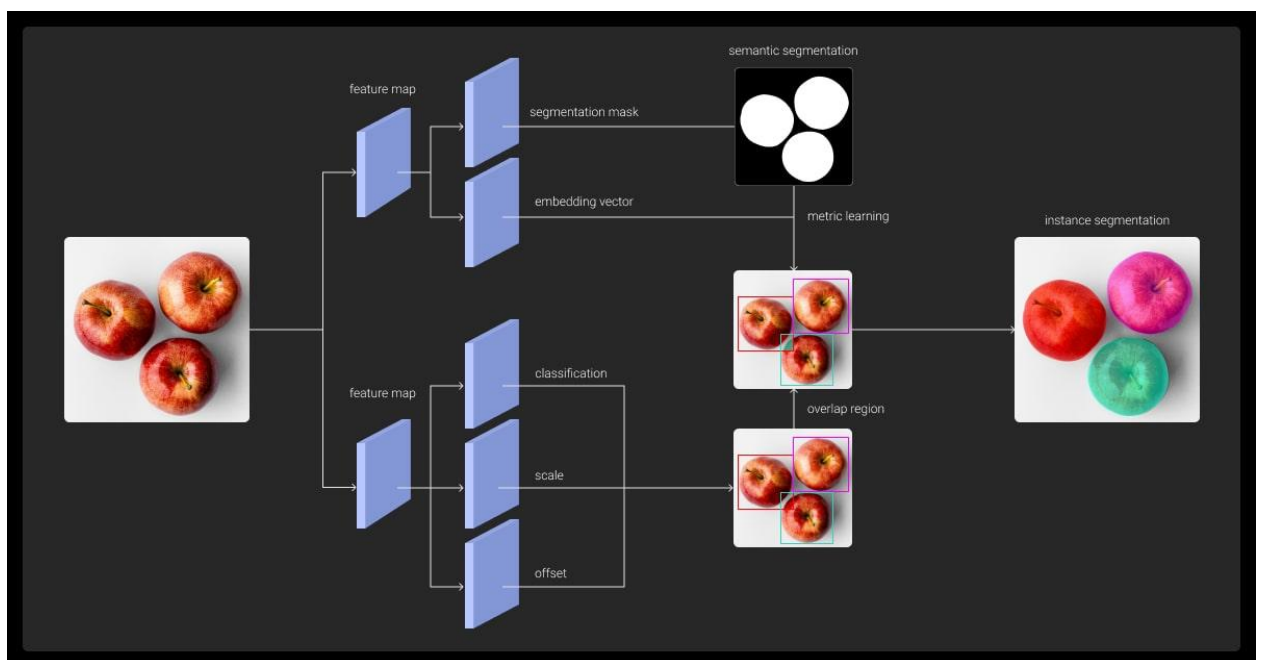


Рисунок 2.4 – Процес сегментації екземплярів

У дослідженні [18] продемонстровано, що застосування *example mining* у *embedding*-класифікаторі дозволяє підвищити точність класифікації з 93,1 % до 96,3 % на внутрішньому наборі даних, а також з 80,4 % до 85,8 % на відкритому наборі даних *Grocery products*. Загалом, дана модель добре працює на зображеннях, подібних до навчальних даних, і досягає середнього показника точності визначення об'єктів 85,0 % та 60,8 % при перенесенні на інші набори даних. Це свідчить про потенціал використання методу в умовах домашнього розпізнавання продуктів.

Описані вище методи продемонстрували ефективність у вирішенні задач розпізнавання продуктів харчування, проте більшість із них мають обмеження, пов'язані з необхідністю попереднього навчання на спеціалізованих наборах даних.

Водночас розвиток мультимодальних моделей штучного інтелекту (наприклад, CLIP, GPT-4o та Gemini 2.0) відкриває нові можливості у цій сфері. Завдяки здатності до *zero-shot* класифікації та обробки як зображень, так і тексту, ці моделі виявляються більш універсальними і гнучкими у контексті розпізнавання продуктів харчування з фотографій у домашніх умовах. Тому особливу увагу при аналізі було приділено саме мультимодальним підходам.

Метод *zero-shot* класифікації продуктів харчування за допомогою моделі CLIP ґрунтується на попередньому спільному навчанні двох модулів кодування – один обробляє зображення, інший – текстові підписи.

У процесі навчання модель отримує близько 400 мільйонів пар «зображення – опис» та створює єдиний векторний простір, де векторні представлення відповідних пар максимально наближені, а невідповідні віддалені. Під час класифікації продуктів харчування користувач надає зображення, а система формує список можливих назв (наприклад, «яблуко», «банан», «помідор»), кожен з яких перетворює на текстовий вектор. Фото також кодується у вигляді вектора, після чого модель обчислює косинус подібності між вектором зображення та векторами всіх назв (рис. 2.5) [19].

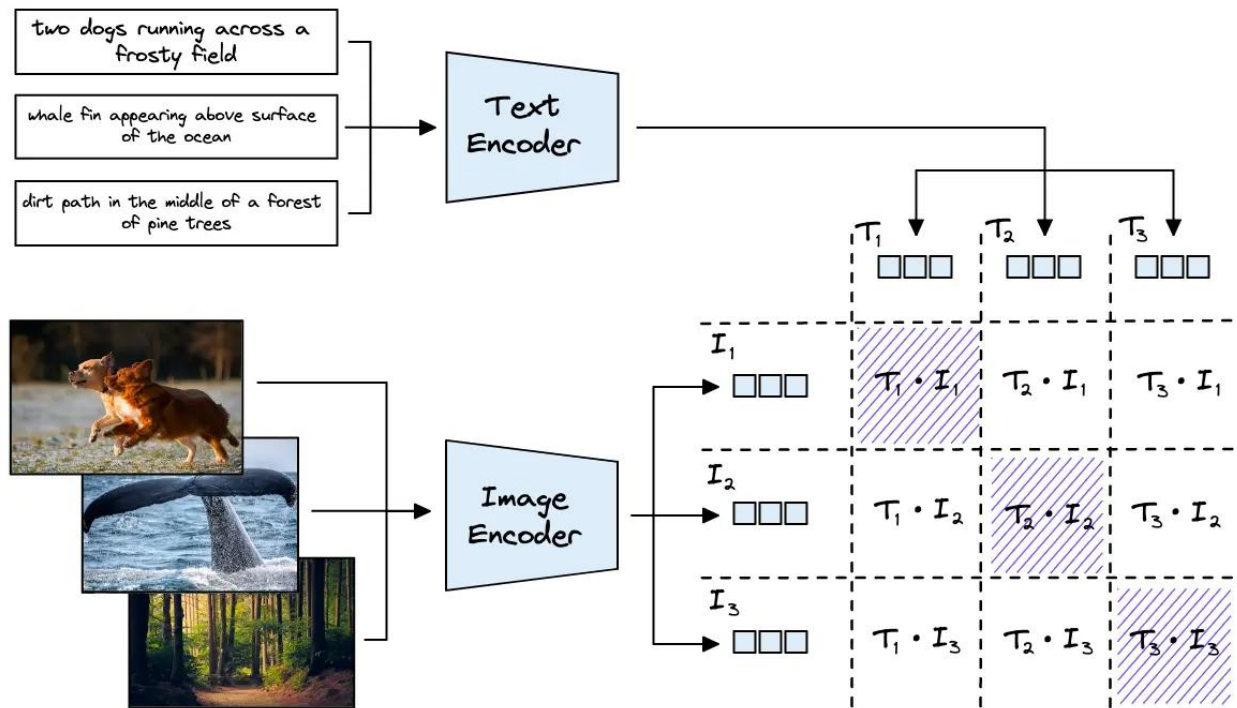


Рисунок 2.5 – Приклад роботи моделі CLIP

Категорія з найвищим показником вважається результатом розпізнавання. Оскільки модель CLIP була навчена на широкому наборі відкритих пар «зображення – опис», вона здатна працювати без донавчання на конкретних продуктах, тобто в zero-shot режимі. Але головним мінусом даного методу є те, що при виборі його розробнику доведеться вручну прописувати всі можливі описи зображень для того, щоб в подальшому модель могла зіставити пари «зображення – опис».

Модель GPT-4o реалізує мультимодальний підхід, що об'єднує обробку зображень і тексту в межах єдиного трансформерного конвеєра. Такий підхід дозволяє здійснювати розпізнавання продуктів харчування без попереднього донавчання, працюючи в zero-shot режимі, що є особливо корисним для вебзастосунків. На початковому етапі вхідне зображення розбивається на фіксовані області (патчі), кожна з яких обробляється через візуальний модуль, заснований на архітектурі Vision Transformer.

У результаті формується послідовність векторних представлень, які містять як візуальні, так і позиційні характеристики патчів.

Паралельно текст запиту (наприклад, «Що зображено на фото?» або «Які продукти видно на знімку?») перетворюється на послідовність текстових токенів. Токен – це мінімальна одиниця тексту, яку обробляє модель трансформерного типу. Ці одиниці, разом із візуальними поданнями, подаються в один обчислювальний граф трансформера GPT-4o, у якому реалізовано спільну обробку інформації (зображення та тексту) (рис. 2.6).

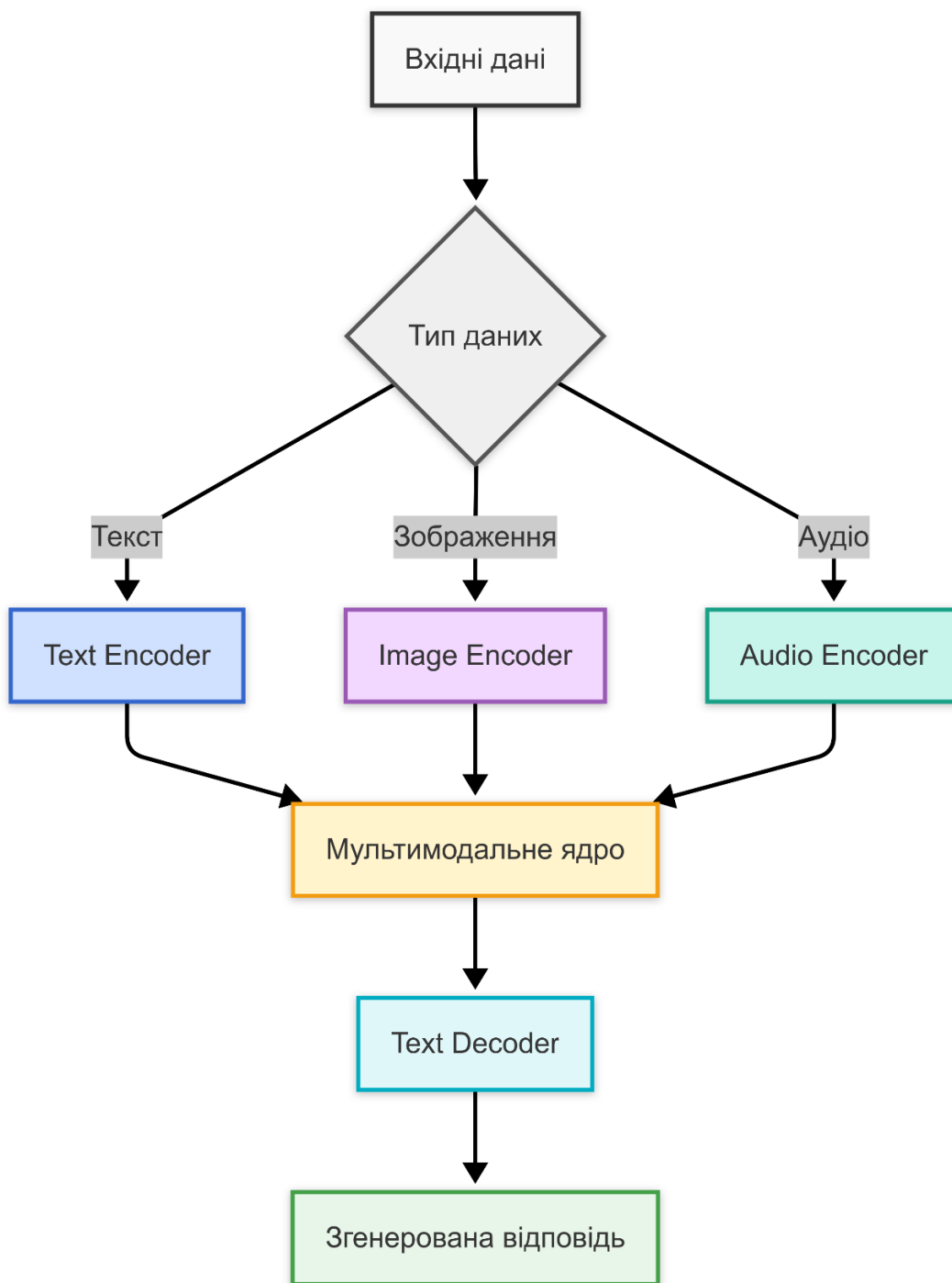


Рисунок 2.6 – Схема роботи моделі GPT-4o

Особливістю архітектури є механізм *cross-modal attention*, який забезпечує взаємозв'язок між візуальним і текстовим контекстами на рівні кожного шару. Завдяки притаманному трансформерним моделям механізму покрокової генерації тексту, модель формує узгоджену текстову відповідь, яка включає розпізнані продукти харчування, їхні назви, а за потреби – подальші інструкції, які розробник може додатково вказати. GPT-4o здатна розпізнавати як сирі продукти (фрукти, овочі, зелень), так і упаковані товари (молоко, соки, макарони) навіть за часткової візуальної видимості або побутових спотворень, що робить її придатною для розпізнавання продуктів харчування у домашніх умовах.

Метод розпізнавання продуктів харчування на основі моделі Gemini 2.0 передбачає використання сучасної мультимодальної трансформерної архітектури, розробленої компанією Google, що забезпечує оброблення різних типів даних – тексту, зображення, аудіо та відео – у межах єдиного середовища. Зокрема, для завдання класифікації сирих або упакованих продуктів у побутових умовах модель працює з вхідними зображеннями, які проходять попереднє перетворення через візуальний модуль кодування, що функціонує за принципом *Vision Transformer* – трансформера, адаптованого для аналізу візуальної інформації.

На першому етапі зображення розбивається на регулярну сітку патчів, кожен із яких кодується у векторне подання за допомогою багат шарової моделі з механізмами самоуваги (*self-attention*). Після додавання просторових координат фрагментів отримані вектори утворюють послідовність візуальних токенів. Ці токени далі узгоджуються з текстовими токенами в межах єдиної трансформерної архітектури, що дозволяє моделі здійснювати *cross-modal attention* – тобто оброблення зображення та тексту в єдиному інформаційному просторі. На відміну від традиційних систем, що мають окремі компоненти для візуальної та мовної обробки, Gemini 2.0 виконує об'єднане моделювання, в якому як візуальні, так і текстові дані представлені в спільному векторному просторі (рис. 2.7).

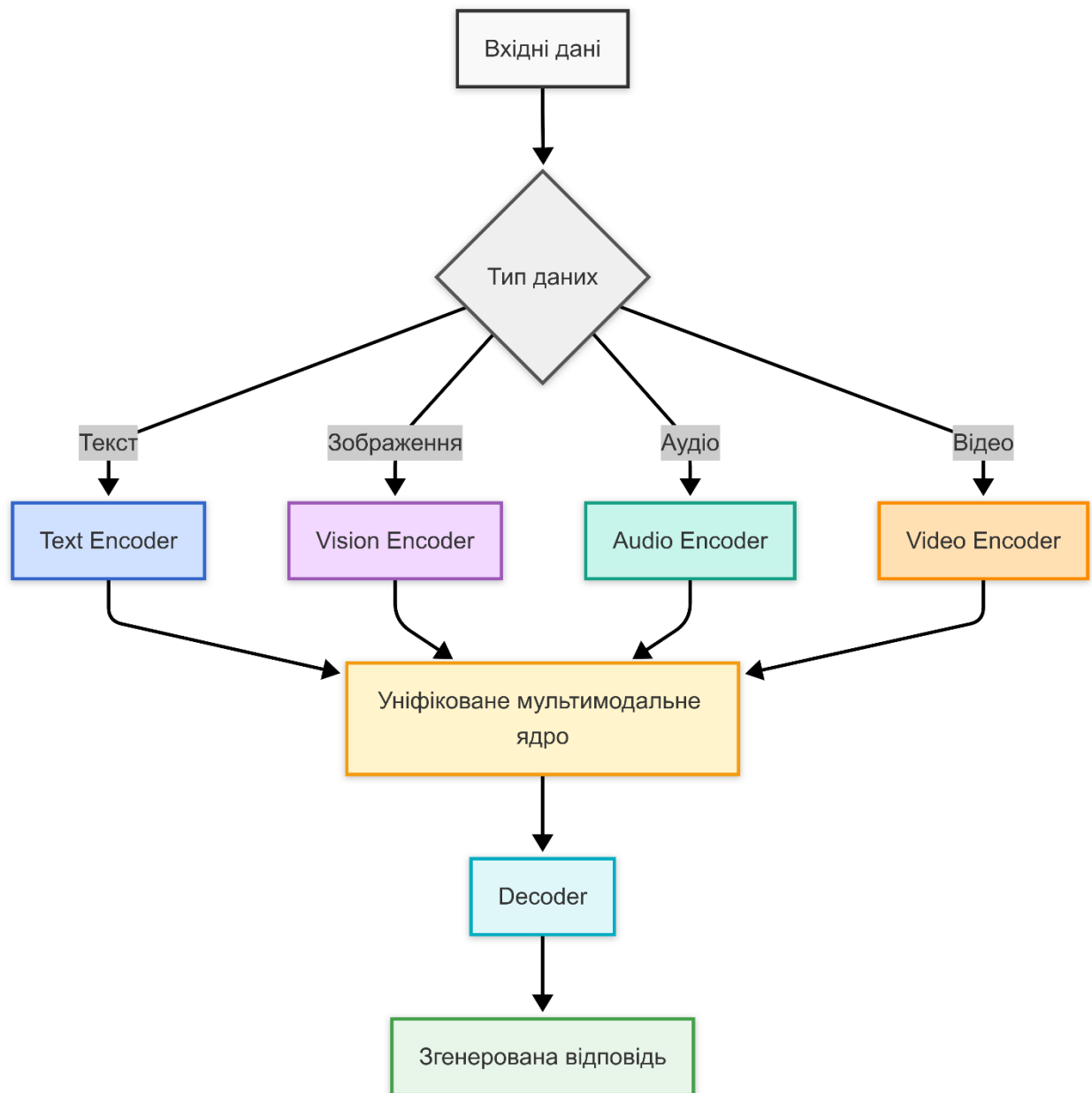


Рисунок 2.7 – Схема роботи моделі Gemini 2.0

Завдяки цьому модель не потребує точно сформульованих запитів або спеціалізованих шаблонів. У zero-shot режимі вона здатна інтерпретувати побутові фото з холодильника, кухонного столу чи супермаркету, ідентифікуючи на них харчові продукти.

Модель формує текстову відповідь за принципом автодоповнення, прогножуючи наступні токени на основі всього наявного контексту, що включає як текстову, так і візуальну інформацію.

У результаті виводиться узгоджена текстова відповідь, що включає визначені об'єкти (наприклад, «томати», «зелена цибуля», «твердий сир»), їхні категорії та, за необхідності, подальші рекомендації щодо використання (наприклад, рецепти на основі знайдених інгредієнтів).

## 2.2 Формування методики розпізнавання продуктів харчування та подальшої генерації рецептів

Ефективна робота системи розпізнавання продуктів харчування та генерації рецептів значною мірою залежить від якості вхідних зображень, що надходять до моделі. Тому до них висувається низка технічних вимог для забезпечення належної якості розпізнавання зображень.

Фотографії мають бути збережені у форматі JPEG із роздільною здатністю не менш ніж  $720 \times 1280$  пікселів та не перевищувати 5 мегабайт. Зйомку бажано здійснювати при денному світлі або за умови рівномірного штучного освітлення, оскільки перепади яскравості, глибокі тіні або засвічені ділянки можуть негативно вплинути на якість розпізнавання.

Фон зображення рекомендується обирати однотонним, без зайвих деталей, які могли б створювати перешкоди для моделі при відокремленні продуктів. У кадрі повинні бути чітко видимі лише ті об'єкти, які підлягають розпізнаванню, тобто сирі або паковані продукти харчування. Також варто уникати накладання об'єктів один на одного або часткового перекриття.

Оптимальна відстань зйомки становить 30 – 40 сантиметрів без використання масштабування, який може знизити чіткість зображення.

Особливу увагу слід приділити налаштуванню фокуса – розмиті зображення, що виникають унаслідок неправильно сфокусованої камери, значно ускладнюють розпізнавання, оскільки модель не здатна точно ідентифікувати об'єкти без чітких контурів.

Дотримання зазначених вимог забезпечує стабільну роботу моделі розпізнавання і підвищує ймовірність формування точного набору інгредієнтів.

У підрозділі 2.1 були розглянуті різні методи розпізнавання продуктів харчування, однак з урахуванням завдання та вимог до гнучкості й універсальності було прийнято рішення надати перевагу сучасним мультимодальним моделям.

У межах подальшого дослідження було здійснено порівняльний аналіз можливостей моделей Gemini 2.0 Flash, GPT-4o та GPT-4.1, щоб визначити метод розпізнавання для поставленої задачі. Створено власний набір даних, що містить 107 фотографій – 84 із продуктами харчування (рис. 2.8) та 23 без них (рис. 2.9).

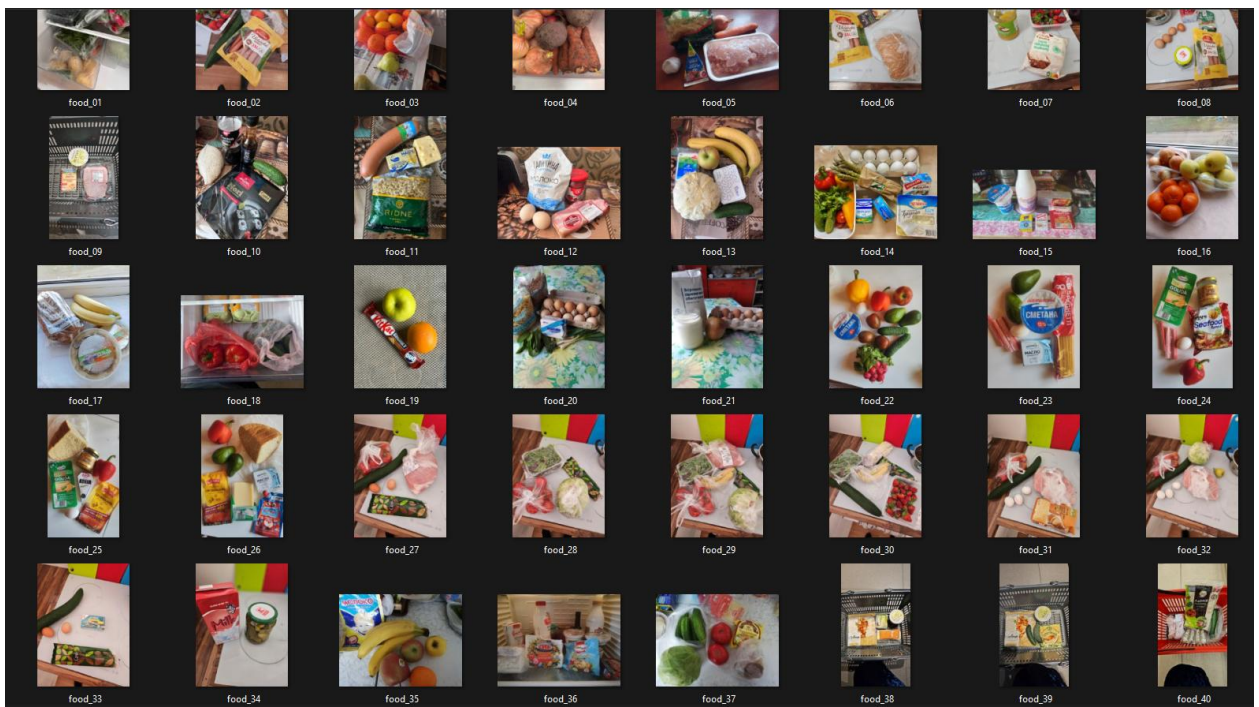


Рисунок 2.8 – Фрагмент створеного набору даних, що містить фотографії з продуктами харчування

До кожного зображення були створені анотації («ground truth») із зазначенням фактичного вмісту кадру, що дозволило об'єктивно оцінити результати розпізнавання шляхом порівняння з відповідями моделей.

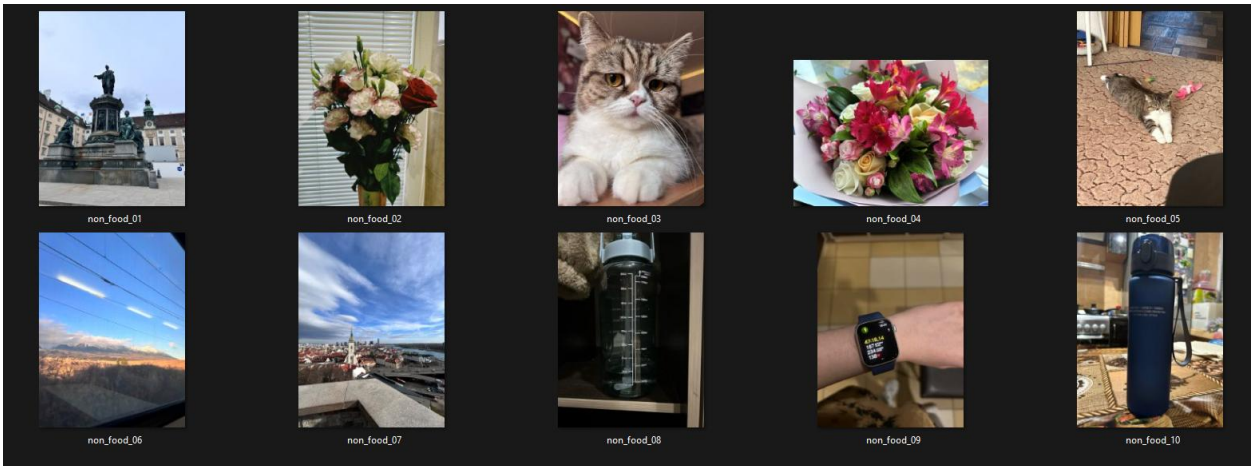


Рисунок 2.9 – Фрагмент створеного набору даних, що містить фотографії без продуктів

Кожне зображення набору даних було надіслано моделям разом із текстовим запитом, після чого фіксувалися результати розпізнавання: основні метрики точності (precision, recall, F1 score), середній час обробки, вартість запиту, вартість обробки набору даних та точність класифікації «їжа/не їжа» (рис. 2.10).

```

Ініціалізація моделі Gemini 2.0...
Ініціалізація моделі GPT-4o...
Ініціалізація моделі GPT-4.1...

Тестування моделі Gemini 2.0...
Тестування Gemini 2.0: 100%
Результати для Gemini 2.0 збережено в results/final_test1/gemini_2.0_results.csv

Загальні метрики для Gemini 2.0:
Precision: 0,800
Recall: 0,780
F1 Score: 0,790
Точність класифікації їжа/не їжа: 0,960
Середній час обробки: 4,66 секунд
Вартість одного запиту: $0,000123
Вартість обробки всього датасету: $0,0062

Тестування моделі GPT-4o...
Тестування GPT-4o: 100%
Результати для GPT-4o збережено в results/final_test1/gpt-4o_results.csv

Загальні метрики для GPT-4o:
Precision: 0,820
Recall: 0,790
F1 Score: 0,800
Точність класифікації їжа/не їжа: 0,980
Середній час обробки: 5,86 секунд
Вартість одного запиту: $0,003075
Вартість обробки всього датасету: $0,1537

Тестування моделі GPT-4.1...
Тестування GPT-4.1: 100%
Результати для GPT-4.1 збережено в results/final_test1/gpt-4.1_results.csv

Загальні метрики для GPT-4.1:
Precision: 0,740
Recall: 0,760
F1 Score: 0,750
Точність класифікації їжа/не їжа: 0,960
Середній час обробки: 5,85 секунд
Вартість одного запиту: $0,002460
Вартість обробки всього датасету: $0,1230

Всі результати збережено в results/final_test1/all_results.csv
Зведена таблиця збережена в results/final_test1/summary.csv

Порівняння моделей:
  Model  Avg F1 Score  Avg Processing Time (s)  Cost per Request ($)  Total Dataset Cost ($)
  GPT-4o      0,65           5,86           0,003075           0,15
  Gemini 2.0   0,64           4,66           0,000123           0,01
  GPT-4.1     0,60           5,85           0,002460           0,12

Експеримент завершено успішно!

```

Рисунок 2.10 – Результат тестування моделей

Далі отримані дані були нормалізовані (вища точність, швидка обробка та нижча вартість вважалися кращими) і помножені на вагові коефіцієнти критеріїв (0,5 – для точності, 0,3 – для часу та 0,2 – для вартості) для обчислення найкращої альтернативи.

Також сформовано загальну статистику для всіх моделей, де було зображено детально усі характеристики тестування (рис. 2.11).

```

Загальна статистика по моделях:
      elapsed_time          precision recall f1_score \
      mean  std  min  max  mean  mean  mean
model
GPT-4.1      5,85  1,51  4,43  11,96  0,60  0,61  0,60
GPT-4o      5,86  0,64  4,81  7,67  0,66  0,64  0,65
Gemini 2.0   4,66  0,21  4,32  5,61  0,66  0,64  0,64

      food_classification_accuracy request_cost
      mean  mean
model
GPT-4.1      0,96  0,002460
GPT-4o      0,98  0,003075
Gemini 2.0   0,96  0,000123

Порівняння вартості:
      Model Avg F1 Score Avg Processing Time (s) Cost per Request ($) \
1      GPT-4o      0,65      5,86      0,003075
0      Gemini 2.0   0,64      4,66      0,000123
2      GPT-4.1      0,60      5,85      0,002460

      Total Dataset Cost ($)
1      0,15
0      0,01
2      0,12

Фінальне порівняння моделей (з ваговими коефіцієнтами):
Модель | Точність (F1) | Час (с) | Вартість ($) | Загальна оцінка
-----|-----|-----|-----|-----
Gemini 2.0 | 0,64 | 4,66 | 0,000123 | 0,99
GPT-4.1 | 0,60 | 5,85 | 0,002460 | 0,51
GPT-4o | 0,65 | 5,86 | 0,003075 | 0,50
Збережено підсумкову статистику в results\final_test1\summary.csv

```

Рисунок 2.11 – Загальна статистика тестування

Побудовано діаграми (рис. 2.12 – рис. 2.16) для наглядної демонстрації результатів тестування кожної моделі.

Отже, із загальної статистики, зображеної на рисунку 2.11, можна зробити висновки, що модель GPT-4o найбільш точно розпізнає продукти харчування (точність становить 0,65), однак, при цьому потребує найбільше часу на обробку одного запиту (5,86 секунди) і має найвищу вартість серед розглянутих моделей (0,0031 долара США за запит).

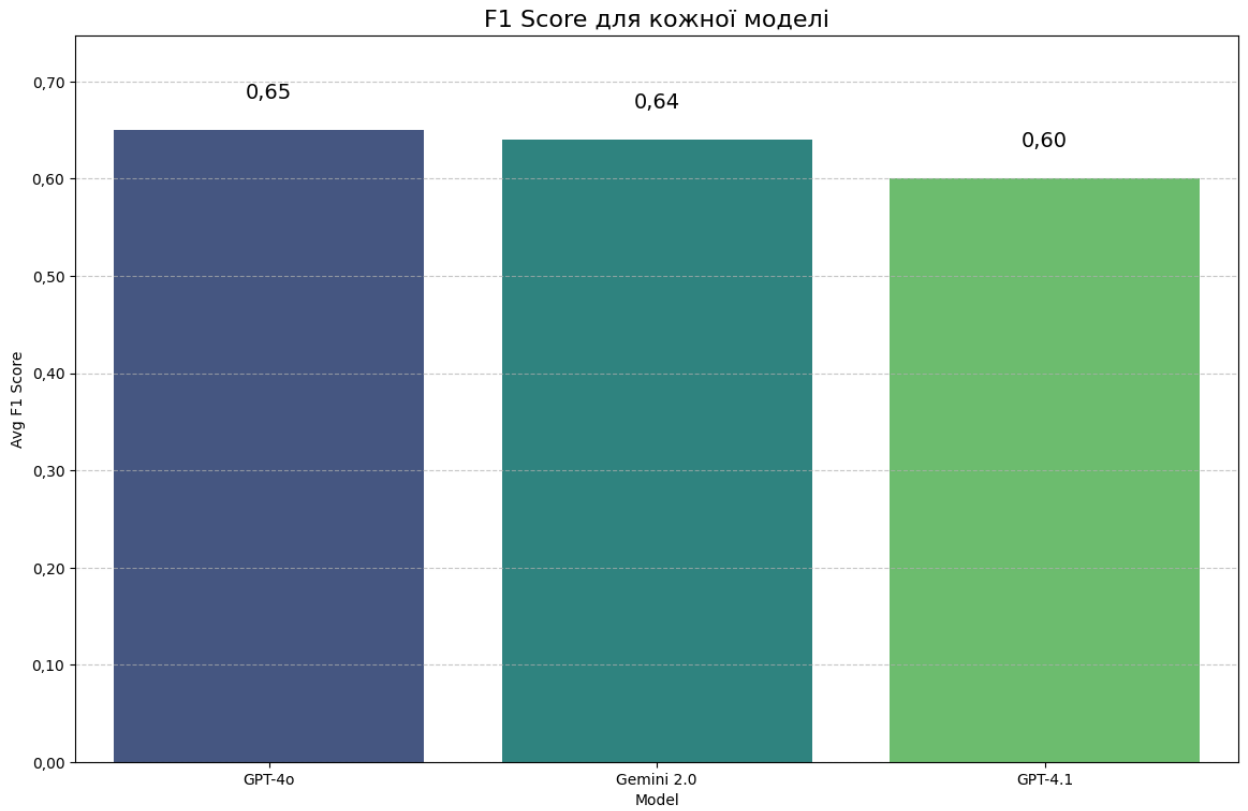


Рисунок 2.12 – Середня міра F1 для кожної моделі

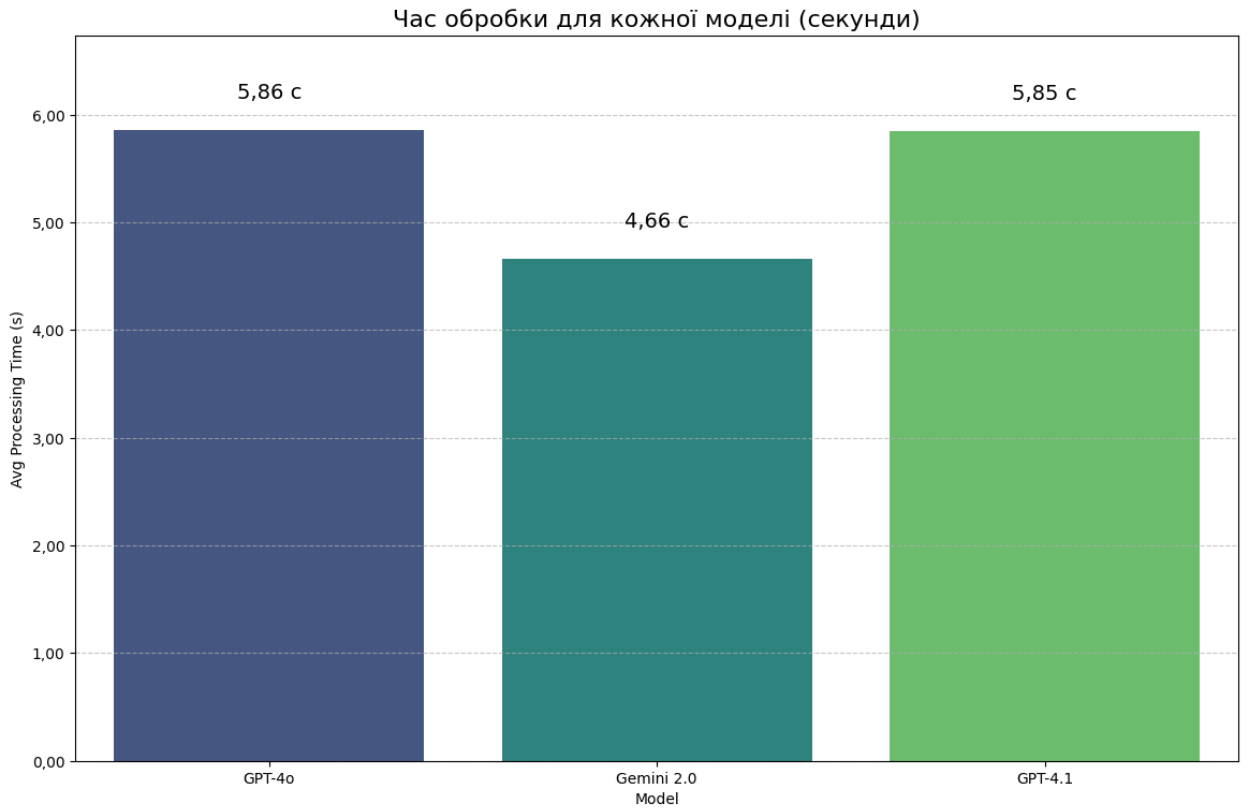


Рисунок 2.13 – Середній час обробки для кожної моделі

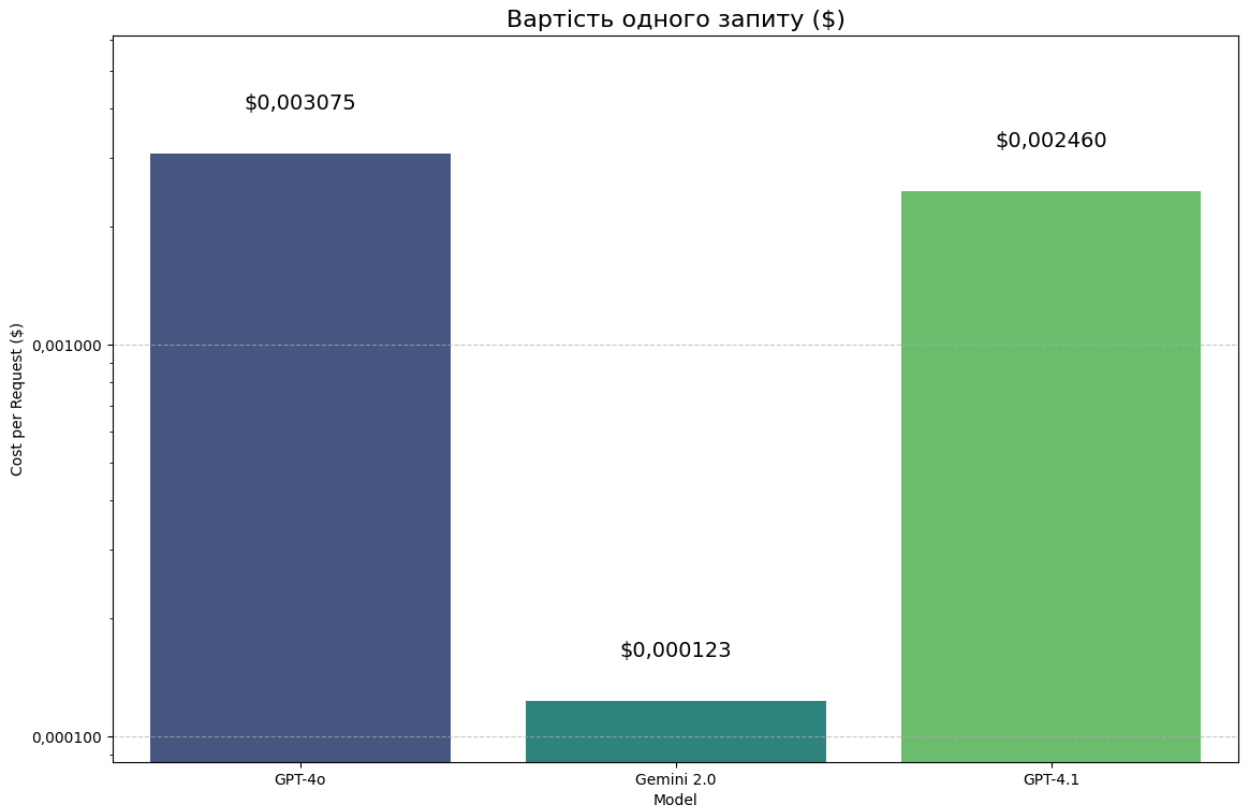


Рисунок 2.14 – Вартість одного запиту для кожної моделі

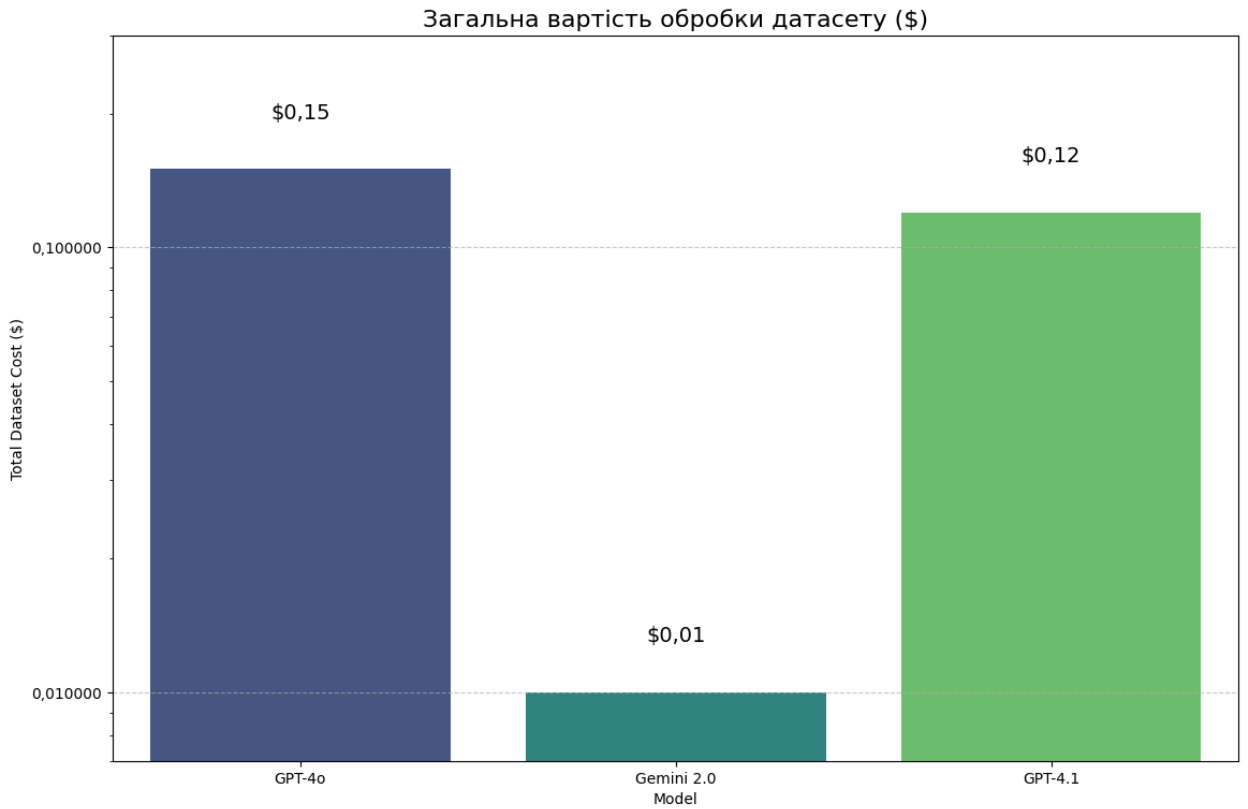
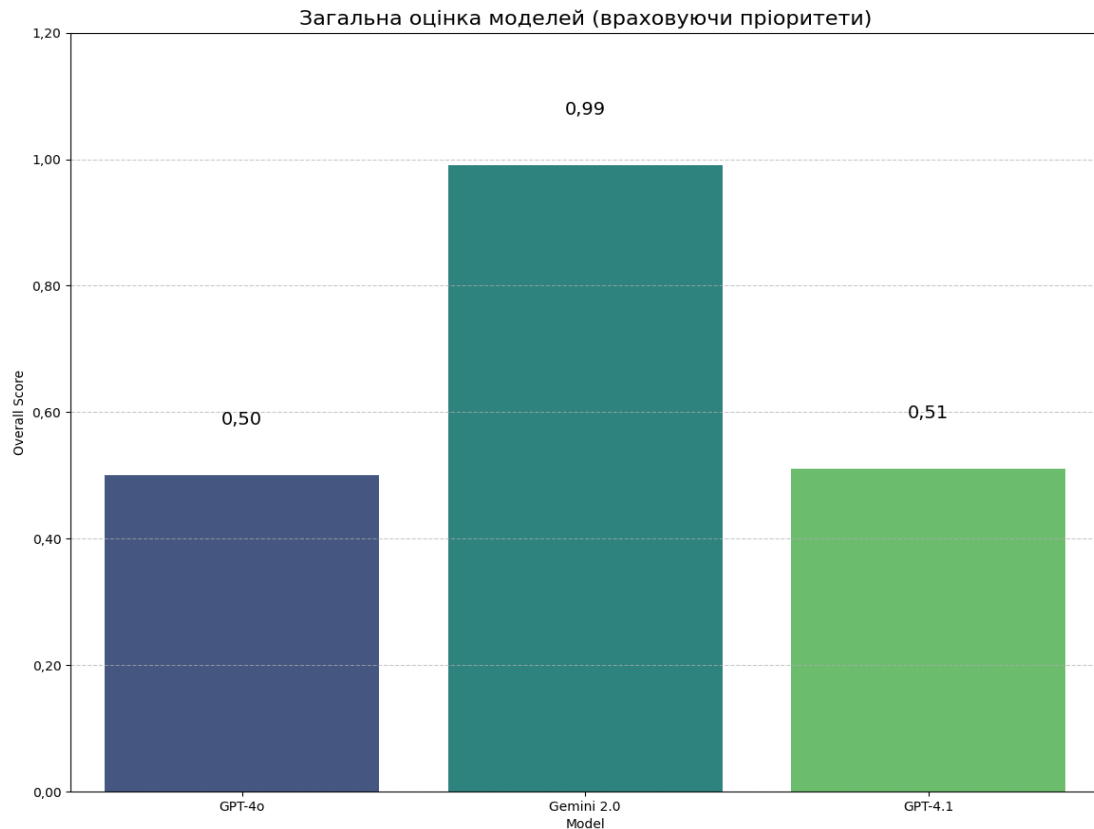


Рисунок 2.15 – Вартість обробки всього набору даних



Вагові коефіцієнти: Точність=0,5, Час=0,3, Вартість=0,2

Рисунок 2.16 – Загальна оцінка моделей

Модель GPT-4.1, незважаючи на те, що є найновішою серед представлених, демонструє найнижчу точність розпізнавання (0,6), а час обробки одного запиту менше лише на 0,01 за GPT-4o (5,85 секунди). Вартість трохи нижча за попередню модель (0,0025 долара США за запит).

У свою чергу, модель Gemini 2.0 Flash забезпечує точність розпізнавання, трохи нижчу за GPT-4o (0,64), однак, обробка одного запиту здійснюється швидше (4,66 секунди), а вартість є найнижчою серед аналізованих моделей – лише 0,0001 долара США за запит.

Порівняльний аналіз показує, що точність Gemini 2.0 Flash є лише на 1,5 % нижчою порівняно з GPT-4o (0,64 проти 0,65), проте за вартістю вона є дешевшою у 31 раз ( $0,0031/0,0001 = 31$ ). Крім того, модель Gemini 2.0 Flash має додаткову перевагу: до 1 мільйона токенів на день надається безкоштовне користування моделлю, що еквівалентно приблизно 8000-8500 запитам із зображенням та текстом, залежно від обсягу.

Також, окрім тестування моделей, було проведено багатокритеріальне порівняння альтернатив, щоб обрати найкращу. Альтернативами виступали моделі Gemini 2.0 Flash, GPT-4o та GPT-4.1, а критеріями для них були «точність», «час обробки» та «вартість одного запиту».

Критеріальний опис альтернатив (табл. 2.1, табл. 2.2) зроблено на основі даних тестування моделей.

Таблиця 2.1 – Критеріальний опис альтернатив

Альтернативи	Точність	Час обробки	Вартість запиту
Gemini 2.0 Flash	середня	швидко	дешево
GPT-4o	висока	довго	дорого
GPT-4.1	низька	середньо	середня

Таблиця 2.2 – Критеріальний опис альтернатив відповідно до шкал

Альтернативи	Точність	Час обробки	Вартість запиту
Gemini 2.0 Flash	2	1	1
GPT-4o	1	3	3
GPT-4.1	3	2	2

Множина альтернатив першої опорної ситуації: {211, 311, 121, 131, 112, 113}. Ці 6 векторних оцінок було порівняно між собою та складено відповідні покрокові матриці попарних порівнянь (рис. 2.17, рис. 2.18).

	211	311	121	131	112	113
211	2	1	0	0	0	0
311	3	2	0	0	0	0
121	0	0	2	1	0	0
131	0	0	3	2	0	0
112	0	0	0	0	2	1
113	0	0	0	0	3	2

Рисунок 2.17 – Початкова матриця попарних порівнянь

	211	311	121	131	112	113
211	2	1	3	1	3	1
311	3	2	3	2	3	2
121	1	1	2	1	1	1
131	3	3	3	2	3	3
112	1	1	3	1	2	1
113	3	3	3	1	3	2

Рисунок 2.18 – Результати попарних порівнянь

Далі було впорядковано множину альтернатив першої опорної ситуації {111, 121, 112, 211, 113, 311, 131} та, використовуючи дану єдину порядкову шкалу, заповнено таблицю 2.3.

Таблиця 2.3 – Векторні оцінки альтернативних моделей

Моделі	Векторна оцінка	Векторна оцінка за єдиною порядковою шкалою	Векторна оцінка за зростанням рангів
Gemini 2.0 Flash	211	411	114
GPT-4o	133	175	157
GPT-4.1	322	623	236

Отже, після проведення багатокритеріального порівняння альтернатив, обрано кращу модель для розпізнавання продуктів – Gemini 2.0 Flash, що підтвердило результат тестування.

Для генерації рецепта з отриманих продуктів харчування після розпізнавання фотографії було обрано модель o4-mini від OpenAI. Ця легка версія GPT-4o має змогу ефективно обробляти перелік інгредієнтів разом із рівнем складності та генерувати зв'язний текст. Під час формування рецепта модель послідовно аналізує список продуктів, оцінює їхню сумісність і логічність поєднання, після чого формує назву страви, деталізований список інгредієнтів і покрокові інструкції.

Завдяки вбудованому механізму логічного узгодження кроків, o4-mini здатна створювати послідовні й зрозумілі рецепти.

Модель також підтримує генерацію у визначеному форматі, наприклад JSON, що важливо для автоматизованих вебзастосунків, які потребують структурованих відповідей без додаткової обробки.

Ці властивості роблять o4-mini ефективним інструментом для завершального етапу системи «список інгредієнтів → рецепт».

У результаті роботи користувач отримує згенерований рецепт українською мовою із урахуванням обраної складності. Рецепт включає в себе назву, складність, перелік потрібних інгредієнтів (розпізнані продукти та базові спеції/олія), а також покроковий опис приготування із приблизним часом на приготування. Якщо ж рецепт створити неможливо – система покаже відповідне повідомлення.

У випадках коли рецепт не можливий, але інгредієнти сумісні – модель запропонує альтернативний варіант (наприклад, якщо на фото всі продукти фрукти, то система запропонує зробити фруктову нарізку тощо).

### **3 ЕТАПИ РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКУ ДЛЯ РОЗПІЗНАВАННЯ ПРОДУКТІВ ХАРЧУВАННЯ ТА ПОДАЛЬШОЇ ГЕНЕРАЦІЇ РЕЦЕПТІВ ЗА ДОПОМОГОЮ МУЛЬТИМОДАЛЬНИХ МОДЕЛЕЙ**

3.1 Вибір інструментальних засобів розроблення вебзастосунку для розпізнавання продуктів харчування та подальшої генерації рецептів за допомогою мультимодальних моделей

Одним із ключових етапів створення вебзастосунку є вибір інструментальних засобів, які забезпечують ефективну реалізацію усіх необхідних функціональних можливостей. Після аналізу сучасних технологій було обрано такі компоненти: мову програмування Python, середовище розробки Visual Studio Code, бібліотеку Gradio для створення інтерактивного інтерфейсу, мультимодальну модель Gemini 2.0 Flash від Google для розпізнавання продуктів харчування на зображеннях, а також мовну модель з мультимодальними можливостями o4-mini від OpenAI для генерації кулінарних рецептів на основі текстового опису інгредієнтів.

Основною мовою розробки було обрано Python – високорівневу мову програмування, яка є стандартом у галузі штучного інтелекту, машинного навчання та обробки зображень. Її популярність обумовлена наявністю розвиненої екосистеми бібліотек і програмних платформ, зручним синтаксисом, а також можливістю швидкої розробки прототипів і інтеграції окремих компонентів програмного забезпечення.

Розроблення застосунку здійснювалося у середовищі Visual Studio Code – зручному багатоплатформному редакторі коду, який підтримує розширення функціональності через додаткові модулі. Завдяки розширенням для роботи з Python, вбудованому командному рядку та засобам для роботи з системою керування версіями Git – середовище забезпечило ефективні умови для написання, тестування та налагодження програмного коду.

Для побудови інтерактивного інтерфейсу застосовано бібліотеку Gradio. Вона дозволяє легко створити вебзастосунок з можливістю завантаження зображення користувачем, перегляду результатів розпізнавання та генерації рецепта. Бібліотека добре поєднується з кодом на Python і дає змогу швидко розгортати демонстраційні версії проєктів навіть на локальному сервері.

Для розпізнавання продуктів на зображеннях було використано модель Gemini 2.0 Flash, яка є мультимодальною та підтримує високоточну обробку вхідних зображень. Вона дозволяє визначати типи продуктів харчування навіть за умов складного фону або поганого освітлення. Модель інтегровано у вебзастосунок шляхом надсилання запиту до попередньо навченої моделі через зовнішній інтерфейс.

Для генерації кулінарних рецептів на основі розпізнаних інгредієнтів використовується o4-mini – компактна мовна модель, що забезпечує швидке формування відповіді без потреби у високих обчислювальних ресурсах. Вона є вдалою альтернативою великим моделям (як-от GPT-4o) у випадках, коли важливо забезпечити автономну роботу та оптимізувати продуктивність. Модель генерує рецепти з покроковим описом, пристосовані до наданого списку продуктів.

Вебзастосунок працює як цілісна система: зображення продуктів надсилається до модуля розпізнавання на основі Gemini 2.0 Flash, після чого список інгредієнтів передається на вхід до o4-mini, яка генерує відповідний рецепт.

Увесь процес автоматизовано, а взаємодія з користувачем здійснюється через інтерфейс, реалізований засобами Gradio.

### 3.2 Етапи програмної реалізації розпізнавання продуктів харчування та подальшої генерації рецептів

Розроблення вебзастосунку для розпізнавання продуктів харчування та подальшої генерації рецептів включає в себе такі етапи:

- розробка частини для розпізнавання продуктів на фотографіях на основі моделі Gemini 2.0 Flash;
- розробка частини для генерації рецепта на основі моделі o4-mini;
- об'єднання цих частин та створення зручного інтерфейсу для користувача на основі бібліотеки Gradio.

Для реалізації частини, що відповідає за розпізнавання продуктів харчування на зображеннях, було створено окремий програмний модуль `ingredient_recognition.py`. Його основна мета – забезпечити взаємодію з мультимодальною моделлю Gemini 2.0 Flash, яка здатна обробляти як текст, так і зображення, та повертати текстову відповідь на запит. Модуль написаний мовою Python з використанням бібліотеки `google.generativeai`, яка забезпечує доступ до API моделі Gemini (рис. 3.1).

```
ingredient_recognition.py > IngredientRecognizer
1 from google.generativeai import GenerativeModel
2 import google.generativeai as genai
```

Рисунок 3.1 – Бібліотеки для доступу до API Gemini

Після завантаження усі потрібних бібліотек та створення класу `RecognitionResponse`, де описано очікуваний формат відповіді від моделі, було описано головний клас `IngredientRecognizer`. У конструкторі цього класу ініціалізується ключ доступу до API моделі `gemini-2.0-flash` та створюється її екземпляр для подальшої роботи. Основним методом класу є `recognize_from_image_bytes`, де спочатку зображення кодується у формат Base64, а потім формується запит до моделі, що включає оброблене зображення та текстову інструкцію.

## Лістинг 3.1 Реалізація описаних класів:

```

class RecognitionResponse(BaseModel):
    ingredients: List[str]

class IngredientRecognizer:
    def __init__(self, api_key=None):
        self.api_key = api_key or os.getenv("GEMINI_API_KEY")
        if not self.api_key:
            raise ValueError("Gemini API key is required. Provide it directly or
set GEMINI_API_KEY environment variable.")
        genai.configure(api_key=self.api_key)
        self.model = GenerativeModel(model_name="gemini-2.0-flash")
    async def recognize_from_image_bytes(self, image_bytes: bytes) ->
List[str]:
        try:
            # Encode the image as base64
            base64_image = base64.b64encode(image_bytes).decode("utf-8")

```

Особливу увагу приділено побудові тексту запиту, адже це відіграє ключову роль у етапі розпізнавання продуктів. Запит було спеціально розроблено з урахуванням специфіки ідентифікації сирих або упакованих продуктів харчування на фотографії. У ньому чітко зазначено, що модель виступає в ролі експерта з розпізнавання продуктів. Описано правила класифікації та найменування інгредієнтів залежно від їх типу. Наприклад, якщо продукт є штучним (яйця, овочі), необхідно точно вказати кількість видимих одиниць. Для сипких, рідких або пакованих продуктів кількість не зазначається. Окремо описано правила найменування м'ясних і молочних продуктів з прикладами типів, щоб при подальшій генерації рецепта не було плутанини.

Додатково у запиті було зазначено, чого не слід робити: ігнорувати написи на упаковках, вигадувати відсутні продукти, використовувати некоректні фрази на кшталт «пачка масла». Також надано приклади правильних і неправильних відповідей, на які модель буде орієнтуватися. Таке детальне формулювання дозволяє уникати помилок генерації, знижує ймовірність «галюцинацій» моделі та забезпечує стабільно правильний формат відповіді.

Лістинг 3.2 Текстова частина запиту до моделі розпізнавання:

```
prompt = """"Ти експерт з розпізнавання продуктів харчування на фотографіях.
```

```
ЗАВДАННЯ: Ідентифікувати всі продукти на фото і вказати точну кількість згідно правил.
```

```
ПРАВИЛА ВИЗНАЧЕННЯ КІЛЬКОСТІ:
```

```
1) ШТУЧНІ ПРОДУКТИ:
```

```
- ЯЙЦЯ: дуже уважно рахуй точну кількість видимих яєць ("4 шт яйця")
```

```
- ФРУКТИ/ОВОЧІ: вказуй кількість цілих одиниць ("3 шт яблука", "2 шт помідори")
```

```
2) ПРОДУКТИ БЕЗ КІЛЬКОСТІ:
```

```
- УПАКОВАНІ ПРОДУКТИ: пиши без "шт" і без "пачка" ("масло", "твердий сир", "шинка")
```

```
- РІДИНИ: пиши без кількості ("молоко", "олія", "кефір")
```

```
- СИПКІ ПРОДУКТИ: пиши без кількості ("борошно", "цукор", "рис")
```

```
КОНКРЕТНІ НАЗВИ ПРОДУКТІВ:
```

```
1) СИРИ - розрізняй типи:
```

```
- "кисломолочний сир" - для білого м'якого творогу
```

```
- "твердий сир" - для жовтих твердих сирів
```

```
- "м'який сир" - для бринзи, фети, адигейського
```

2) М'ЯСО - вказуй конкретний вид:

- "свинина", "яловичина", "курятина", "індичка"

- Якщо можливо, вказуй частину: "куряче філе", "свиняча вирізка"

3) МОЛОЧНІ ПРОДУКТИ - розрізняй:

- "сметана", "йогурт", "кефір", "масло"

ІНСТРУКЦІЇ ТА ЗАБОРОНИ:

1) ІГНОРУЙ написи на упаковках - орієнтуйся на візуальні ознаки

2) НЕ ВИГАДУЙ продукти, яких не бачиш на фото

3) ЗАБОРОНЕНІ ФРАЗИ: "пачка масла", "1 шт сир", "плитка шоколаду", "пакет цукру"

4) Пиши відповідь УКРАЇНСЬКОЮ мовою

ВІДПОВІДАЙ у форматі списку інгредієнтів, розділеного комами.

ПРАВИЛЬНІ ПРИКЛАДИ:

1) "5 шт яєць, масло, твердий сир, цибуля"

2) "2 шт яблука, банан, молоко, сметана"

3) "3 шт помідори, огірок, шинка, пшеничний хліб"

ПОМИЛКИ ТА ВИПРАВЛЕННЯ:

1) Неправильно: "1 шт масло" / Правильно: "масло"

2) Неправильно: "пачка сиру" / Правильно: "твердий сир" або "кисломолочний сир"

3) Неправильно: "3 шт яблука, 1 пакет цукру" / Правильно: "3 шт яблука, цукор"

Перевір свою відповідь перед відправкою."""

Після отримання відповіді від моделі, текст обробляється: зайві пробіли видаляються, а список інгредієнтів формується через розділення комами або переходами на новий рядок. У разі виникнення помилок під час розпізнавання (наприклад, через неправильний формат зображення або відсутність ключа доступу) реалізовано журналювання за допомогою стандартного модуля logging.

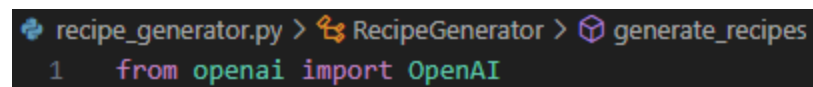
## Лістинг 3.3 Обробка відповіді моделі та обробка помилок:

```

response = self.model.generate_content(
    contents=[
        {
            "role": "user",
            "parts": [
                {"text": prompt},
                {"inline_data": {
                    "mime_type": "image/jpeg",
                    "data": base64_image}}]}],
    generation_config={"temperature": 0.0}
)
ingredients_text = response.text.strip()
if "," in ingredients_text:
    ingredients = [item.strip().lower() for item in
ingredients_text.split(",")]
else:
    ingredients = [item.strip().lower() for item in
ingredients_text.split("\n") if item.strip()]
    if not ingredients:
        ingredients = [ingredients_text.lower()]
    ingredients = [ing[:-1] if ing.endswith('.') else ing for ing in
ingredients]
    return ingredients
except Exception as e:
    logger.error(f"Error recognizing ingredients: {str(e)}")
    raise Exception(f"Error recognizing ingredients with Gemini:
{str(e)}")

```

Другим важливим етапом є розробка частини, що відповідає за генерацію кулінарного рецепта на основі розпізнаних інгредієнтів. Для її реалізації було створено окремий програмний модуль `recipe_generation.py`. Його основне завдання полягає у створенні рецептів на основі списку розпізнаних інгредієнтів, отриманого попереднім етапом, з урахуванням кулінарних правил сумісності та базової логіки. Даний модуль реалізований за допомогою бібліотеки OpenAI, що забезпечує доступ до API мовної моделі GPT серії o4-mini (рис. 3.2).



```
recipe_generator.py > RecipeGenerator > generate_recipes
1 from openai import OpenAI
```

Рисунок 3.2 – Бібліотека для роботи моделі генерації рецепта

У модулі визначено основний клас `RecipeGenerator`, який інкапсулює логіку генерації рецептів. У його конструкторі реалізована підтримка передачі готового клієнта OpenAI, а якщо клієнт не передається – модуль самостійно ініціалізує його на основі ключа зі змінної середовища. У разі його відсутності викликається виняток, який одразу припиняє виконання з інформативним повідомленням.

Лістинг 3.4 Конструктор класу `RecipeGenerator`:

```
class RecipeGenerator:
    def __init__(self, openai_client=None):
        if openai_client:
            self.client = openai_client
        else:
            api_key = os.getenv("OPENAI_API_KEY")
            if not api_key:
                raise ValueError("OpenAI API key is required. Please set
OPENAI_API_KEY environment variable.")
```

```
self.client = OpenAI(api_key=api_key)
```

Ключова функція модуля – метод `generate_recipes`, що відповідає за побудову запиту до моделі. Даний метод приймає два аргументи: список розпізнаних інгредієнтів та рівень складності рецепта (легкий, середній, складний). Далі формується запит, який містить чіткі інструкції для мовної моделі `o4-mini`. У ньому чітко зазначено, що модель виконує роль досвідченого шеф-кухаря, а також правила щодо базових продуктів (сіль, перець, вода тощо), логіки комбінування інгредієнтів та прикладів несумісних поєднань.

Також вказуються вимоги до формату: відповідь має бути у форматі JSON-об'єкта з такими ключами, як назва рецепта, інгредієнти, інструкція, складність, кількість порцій, час приготування, поради щодо подачі, а також список невикористаних інгредієнтів. У випадку неможливості сформувавши рецепт – модель має повернути спеціальну ознаку «`compatible`»: `false`, щоб система повідомила користувача про неможливість створення рецепта.

Лістинг 3.5 Основна частина методу `generate_recipes`:

```
async def generate_recipes(self, ingredients: List[str], difficulty: str = None)
-> Dict[str, Any]:
    try:
        difficulty_guides = {
            "легкий": "прості рецепти, 3-5 кроків, до 30 хвилин, базові
техніки приготування",
            "середній": "середня складність, 5-7 кроків, до 60 хвилин,
різноманітніші техніки",
            "складний": "складні техніки, 7+ кроків, понад 60 хвилин,
комплексні методи приготування"}
        diff_guide = difficulty_guides.get(difficulty.lower(),
difficulty_guides["середній"])
```

```
prompt = f"""
```

Ти досвідчений шеф-кухар. Створи смачний рецепт зі списку інгредієнтів: {' '.join(ingredients)}.

**ЗАВДАННЯ:** Створити смачну страву, використовуючи логічну підмножину доступних інгредієнтів.

**ОБОВ'ЯЗКОВЕ ПРАВИЛО:** Використовуй **ТІЛЬКИ** інгредієнти зі списку + базові продукти (сіль, перець, цукор, олія, вода).

**ПРАВИЛА ПРИГОТУВАННЯ:**

- 1) Продукти, що потребують термічної обробки (м'ясо, риба, яйця, картопля, буряк) - **ЗАВЖДИ** готуй термічно
- 2) Використовуй лише логічно сумісні інгредієнти - не обов'язково всі
- 3) Для продуктів з вказаною кількістю (напр. "10 шт яєць") - бери лише необхідну кількість (2-3 шт)
- 4) Для продуктів без вказаної кількості - додавай реалістичні пропорції

**ЗАБОРОНЕНІ КОМБІНАЦІЇ:**

- 1) Молоко + риба або молоко + огірки(свіжі або консервовані)
- 2) Будь-які експериментальні поєднання, що суперечать кулінарній логіці

**СКЛАДНІСТЬ РЕЦЕПТА** - {difficulty}: {diff\_guide}

**ВИМОГИ ДО ФОРМАТУ:**

- 1) Вказуй реалістичний час приготування для кожного етапу
- 2) Описуй детально способи обробки інгредієнтів
- 3) Обов'язково включи невикористані інгредієнти до списку "unused\_ingredients"

**ФОРМАТ ВІДПОВІДІ (JSON):**

```
{ {
  "compatible": true,
  "recipes": [
    { {
      "name": "Назва рецепта",
```

```

"ingredients": ["2 шт яйця", "300 г картоплі", "1 шт цибуля", "2 ст.л.
олії", "..."],
"instructions": "1. Перший крок.\n2. Другий крок.\n3. Третій крок.",
"total_time": час у хвилинах,
"servings": кількість порцій,
"difficulty": "{difficulty}",
"servicing_suggestions": "Порада щодо подачі.",
"unused_ingredients": ["помідори", "шинка"]
}}
],
"message": "Короткий коментар чому деякі інгредієнти не
використані."
}}
Якщо створення рецепта неможливе через несумісність усіх
інгредієнтів, відповідай СТРОГО у такому форматі:
{{
"compatible": false,
"recipes": [],
"message": "З цих інгредієнтів неможливо створити смачну страву.
Спробуйте завантажити фото з іншими продуктами."
}}""

```

У випадку виникнення виключень або помилок у відповіді модуль здійснює протоколювання за допомогою стандартного журналювання, щоб забезпечити відстеження проблем на етапі генерації. Усі критичні ситуації обробляються належним чином, без порушення роботи всього застосунку.

Лістинг 3.6 Обробка відповіді моделі та обробка помилок генерації:

```

model = "o4-mini"
completion_params = {

```

```

"model": model,
"messages": [
    {
        "role": "user",
        "content": prompt,
    },
],
"response_format": {"type": "json_object"}
}
completion =
self.client.chat.completions.create(**completion_params)
text = completion.choices[0].message.content.strip()
try:
    recipe_data = json.loads(text)
    return recipe_data
except json.JSONDecodeError as e:
    logger.error(f"Error parsing recipe JSON: {e}")
    return {
        "message": f"На жаль, не вдалося створити рецепт з цих
інгредієнтів: {str(e)}",
        "recipes": []
    }
except Exception as e:
    logger.error(f"Error generating recipes: {str(e)}")
    return {
        "message": f"На жаль, не вдалося створити рецепт з цих
інгредієнтів: {str(e)}",
        "recipes": []
    }

```

Останній етап програмної реалізації полягає в інтеграції модулів розпізнавання інгредієнтів і генерації рецептів у єдиний інтерфейс, розроблений за допомогою бібліотеки Gradio. На цьому етапі було реалізовано повноцінний контролер, який пов'язує попередньо створені частини застосунку в узгоджену логіку роботи. Перед тим як зображення потрапляє до модуля розпізнавання інгредієнтів, воно проходить кілька ключових перевірок для забезпечення коректності вхідних даних. Це дозволяє запобігти помилкам, підвищити стабільність роботи системи та зменшити навантаження на моделі.

Спочатку перевіряється тип файлу: на вхід система приймає лише формат JPEG (з розширенням .jpg або .jpeg), оскільки саме такий формат у фотографій. Якщо зображення має інший формат, користувач отримає відповідне повідомлення.

Далі система перевіряє роздільну здатність завантаженого зображення: воно має бути не менше 720×1280 або 1280×720 пікселів, щоб забезпечити достатню чіткість для розпізнавання продуктів. Якщо зображення буде меншої роздільної здатності, система повідомить.

Також, аби уникнути перевантаження моделі під час передачі зображення у байтах, бібліотека Gradio автоматично стискає великі за розміром файли. Також додатково зроблено перевірку у коді, яка перевіряє розмір файлу.

Лістинг 3.7 Функція для перевірки зображення:

```
def validate_image(image_path):  
    if image_path is None:  
        return False, "Будь ласка, завантажте зображення продуктів.", None  
    img_type = imghdr.what(image_path)  
    if img_type not in ("jpeg", "jpg"):  
        return False, "Зображення має бути у форматі JPEG (.jpg, .jpeg).",  
None
```

```

file_size = os.path.getsize(image_path)
if file_size > 5 * 1024 * 1024:
    return False, f"Розмір зображення ({file_size/1024/1024:.2f} MB)
перевищує 5 MB.", None

img = PIL.Image.open(image_path)
width, height = img.size
if not ((width >= 720 and height >= 1280) or (width >= 1280 and height
>= 720)):
    return False, f"Розмір зображення ({width}x{height}) менший за
720x1280 або 1280x720.", None

return True, "Зображення відповідає вимогам.", img

```

Також була зроблена перевірка на завантаження зображення та чи була обрана складність. Лише після проходження всіх перевірок зображення конвертується у масив байтів та передається до модуля `IngredientRecognizer`, який використовує модель `Gemini` для виявлення харчових продуктів на фото. Результат розпізнавання передається до модуля `RecipeGenerator`, який звертається до моделі `OpenAI` для побудови рецепта з урахуванням рівня складності та сумісності інгредієнтів.

У разі успішної генерації, користувач отримує назву рецепта, перелік використаних продуктів, покрокові інструкції приготування, інформацію про порції та час приготування, а також додаткові рекомендації щодо подачі. У випадках, коли інгредієнти несумісні та не дозволяють створити рецепт, користувачеві виводиться відповідне повідомлення.

Лістинг 3.8 Перевірки та вивід згенерованого рецепта користувачу:

```

async def recipe_generation(image_path: str, difficulty: str) -> tuple[str, str,
str]:
    try:
        if not difficulty:

```

```

    return "# Помилка", "", "Будь ласка, виберіть складність рецепт."
is_valid, message, image = validate_image(image_path)
if not is_valid:
    return "# Помилка", "", message
buffered = io.BytesIO()
image.save(buffered, format="JPEG", quality=95)
image_bytes = buffered.getvalue()
# Step 1: Recognize ingredients with Gemini
logger.info("Розпізнавання інгредієнтів...")
ingredients = await
ingredient_recognizer.recognize_from_image_bytes(image_bytes)
# Перевірка чи розпізнані продукти
if not ingredients or len(ingredients) == 0:
    return "# Не знайдено продуктів", "", "На зображенні не вдалося
розпізнати жодних продуктів харчування. Будь ласка, завантажте інше фото з
чітко видимими продуктами."
    ingredients_text = "## Розпізнані інгредієнти:\n" + ",
".join(ingredients)
    logger.info(f"Розпізнані інгредієнти: {ingredients}")
# Step 2: Generate recipe with o4-mini using difficulty level
difficulty_map = {
    "Легкий": "легкий",
    "Середній": "середній",
    "Важкий": "складний"
}
logger.info(f"Генерація рецепта для інгредієнтів зі складністю
{difficulty}...")
# Генеруємо рецепт з перевіркою сумісності інгредієнтів

```

```

modified_recipe_data = await
recipe_generator.generate_recipes(ingredients, difficulty_map.get(difficulty,
"середній"))
    if modified_recipe_data.get("compatible") == False:
        # Якщо інгредієнти несумісні, виводимо просте повідомлення
        message = modified_recipe_data.get("message", "З цих інгредієнтів
неможливо створити смачну страву. Спробуйте завантажити фото з іншими
продуктами.")
        return "# Несумісні інгредієнти", ingredients_text, f"{message}"
    # Format recipe output
    if modified_recipe_data.get("recipes"):
        recipe = modified_recipe_data["recipes"][0]
        recipe_text = f"## {recipe.get('name', 'Рецепт')}\n\n"
        recipe_text += "### Інгредієнти:\n"
        for ingredient in recipe.get('ingredients', []):
            recipe_text += f"* {ingredient}\n"
        recipe_text += "\n### Інструкції:\n"
        instructions = recipe.get('instructions', 'Інструкції відсутні')
        # Переконаємося, що інструкції мають правильне форматування
        if "\\n" in instructions:
            instructions = instructions.replace("\\n", "\n")
        recipe_text += instructions
        recipe_text += "\n\n### Інформація:\n"
        recipe_text += f"* **Загальний час приготування:**
{recipe.get('total_time', 0)} хвилин\n"
        recipe_text += f"* **Кількість порцій:** {recipe.get('servings',
2)}\n"
        # Додаємо поради з подачі, якщо є
        if recipe.get('serving_suggestions'):

```

```

        recipe_text += f"\n###
Подача:\n{recipe.get('serving_suggestions')}
        # Додаємо інформацію про невикористані інгредієнти
        unused_ingredients = recipe.get('unused_ingredients', [])
        if unused_ingredients and len(unused_ingredients) > 0:
            if modified_recipe_data.get("message"):
                recipe_text += f"\n\n###
Примітка:\n{modified_recipe_data.get('message')}
                return "# Рецепт", ingredients_text, recipe_text
            else:
                message = modified_recipe_data.get("message", "На жаль, з цих
інгредієнтів неможливо створити повноцінний рецепт.")
                return "# Результат аналізу", ingredients_text, f"{message}"
        except Exception as e:
            logger.error(f"Помилка в процесі генерації рецепта: {str(e)}")
            return "# Помилка", "", f"Сталася помилка при генерації рецепта:
{str(e)}"

```

Отже, після розробки усіх трьох етапів, було створено вебзастосунок з інтерфейсом Gradio у вигляді блочної структури з двома основними колонками: вхідними даними та результатами генерації рецепта.

У лівій колонці користувач може завантажити зображення, яке містить сирі або паковані продукти. Компонент для завантаження зображення налаштовано максимально спрощено, щоб уникнути конфліктів – використовується тип `filepath` та обмежено джерела лише до завантаження з комп'ютера.

Також додано інструкцію з технічними вимогами до зображень: підтримується лише JPEG-формат, файл повинен мати роздільну здатність не менше ніж  $720 \times 1280$  або  $1280 \times 720$  пікселів і розмір не перевищувати 5 мегабайтів.

Нижче розміщено радіокнопки для вибору бажаного рівня складності рецепта (легкий, середній або важкий), що впливає на логіку генерації. Кнопка запуску ініціює повний цикл обробки: зображення спочатку аналізується моделлю розпізнавання інгредієнтів, після чого отримані продукти передаються до мовної моделі для побудови рецепта. Кнопка очищення скидає всі елементи інтерфейсу до початкового стану – це реалізовано через окрему функцію `clear_outputs`.

У правій колонці розміщено результати роботи системи: список розпізнаних продуктів, назву створеної страви, список інгредієнтів та покрокову інструкцію. У випадках, коли з наданих інгредієнтів не вдалося побудувати жодного рецепта, передбачено повернення спеціального повідомлення про несумісність.

Наприкінці інтерфейсу додано коротку інструкцію для користувача, яка пояснює послідовність дій та логіку формування рецепта.

Лістинг 3.9 Реалізація інтерфейсу за допомогою Gradio:

```
def clear_outputs():
    """Функція для очищення всіх полів інтерфейсу"""
    return None, None, "", "", ""
with gr.Blocks() as demo:
    gr.Markdown("# Генератор рецептів")
    with gr.Row():
        with gr.Column():
            image_input = gr.Image(
                label="Зображення продуктів",
                type="filepath",
                sources=["upload"],
                image_mode="RGB"
            )
            gr.Markdown("""
```

```

**Вимоги до зображень:**
- Формат JPEG (.jpg, .jpeg)
- Розмір не менше 720×1280 або 1280×720 пікселів
- Розмір файлу не більше 5 МВ
""")
difficulty = gr.Radio(
    ["Легкий", "Середній", "Важкий"],
    label="Складність рецепту"
)
submit_button = gr.Button("Розпізнати інгредієнти та згенерувати
рецепт", variant="primary")
clear_button = gr.Button("Очистити все")
with gr.Column():
    title_output = gr.Markdown("# Рецепт")
    ingredients_output = gr.Markdown()
    recipe_output = gr.Markdown()
submit_button.click(
    fn=recipe_generation,
    inputs=[image_input, difficulty],
    outputs=[title_output, ingredients_output, recipe_output]
)
clear_button.click(
    fn=clear_outputs,
    inputs=[],
    outputs=[image_input, difficulty, title_output, ingredients_output,
recipe_output]
)
gr.Markdown("""
### Як користуватися:
1. Завантажте фотографію наявних продуктів (JPEG формат)

```

2. Виберіть бажаний рівень складності рецепту
3. Натисніть кнопку для розпізнавання інгредієнтів та генерації рецепту
4. Отримайте список розпізнаних інгредієнтів та рецепт (якщо можливо його створити)

### Примітка:

Система автоматично підбирає найкращу комбінацію інгредієнтів для створення смачної страви.<br>


Не всі розпізнані інгредієнти будуть використані у рецепті - програма вибере логічне поєднання продуктів.

""")

Результат роботи лістингу 3.9 продемонстровано на рисунках 3.3, 3.4.

## Генератор рецептів

Зображення продуктів
x



Перетягніть зображення сюди  
- або -  
Натисніть щоб завантажити

**Вимоги до зображень:**

- Формат JPEG (.jpg, .jpeg)
- Розмір не менше 720x1280 або 1280x720 пікселів
- Розмір файлу не більше 5 MB

Складність рецепту

Легкий
 Середній
 Важкий

Розпізнати інгредієнти та згенерувати рецепт



Очистити все

**Як користуватися:**

1. Завантажте фотографію наявних продуктів (JPEG формат)
2. Виберіть бажаний рівень складності рецепту
3. Натисніть кнопку для розпізнавання інгредієнтів та генерації рецепту
4. Отримайте список розпізнаних інгредієнтів та рецепт (якщо можливо його створити)

**Примітка:**

Система автоматично підбирає найкращу комбінацію інгредієнтів для створення смачної страви.  
Не всі розпізнані інгредієнти будуть використані у рецепті - програма вибере логічне поєднання продуктів.

Use via API  · Зроблено на основі gradio 

### Рецепт

Рисунок 3.3 – Інтерфейс створеного вебзастосунку

**Генератор рецептів**

Зображення продуктів

**Рецепт**

**Розпізнані інгредієнти:**  
2 шт яйця, молоко, томатна паста, грудинка

**Томатно-молочний омлет з грудинкою**

**Інгредієнти:**

- 2 шт яйця
- 50 мл молока
- 2 ст.л. томатної пасты
- 150 г грудинки
- 1 ст.л. олії
- 50 мл води
- сіль за смаком
- перець за смаком
- щіпка цукру

**Інструкції:**

1. Підготуйте інгредієнти (5 хв).
  - Грудинку наріжте смужками або кубиками (150 г).
2. Розігрійте сковороду з 1 ст.л. олії на середньому вогні, викладіть грудинку та обсмажуйте до золотистої скоринки, приблизно по 5 хв. з кожного боку (10 хв).
3. Додайте до грудинки томатну пасту (2 ст.л.), 50 мл води, щіпка цукру, сіль і перець. Ретельно перемішайте та тушуйте на середньому вогні ще 5 хвилин (5 хв).
4. У чистій мисці збийте 2 яйця з 50 мл молока, сіллю та перцем до однорідності (3 хв).
5. Зменшіть вогонь до мінімуму, рівномірно вилийте яєчно-молочну суміш на грудинку з томатною основою. Накрийте кришкою та готуйте 7–8 хвилин, поки омлет не загусне (8 хв).
6. Зніміть сковороду з вогню, дайте омлету «дозріти» під кришкою ще 2 хвилини (2 хв).

**Інформація:**

- Загальний час приготування: 33 хвилин
- Кількість порцій: 2

**Подача:**  
Подавайте гарячим, можна поприкрасити свіжою зеленню або дробкою червоного перцю.

Вимоги до зображень:

- Формат JPEG (.jpg, .jpeg)
- Розмір не менше 720×1280 або 1280×720 пікселів
- Розмір файлу не більше 5 МВ

Складність рецепту

Легкий  Середній  Важкий

**Розпізнати інгредієнти та згенерувати рецепт**

Очистити все

Рисунок 3.4 – Приклад роботи вебзастосунку

### 3.3 Тестування розробленого вебзастосунку та аналіз результатів

З метою перевірки надійності та працездатності розробленого вебзастосунку було проведено серію ручних функціональних тестувань. Особливу увагу приділено критичним етапам взаємодії користувача із системою, зокрема обробці вхідного зображення, перевірці його відповідності технічним вимогам, а також поведінці інтерфейсу у випадку не обраної складності (рис. 3.5 – рис. 3.8).

Також було проведено тестування згенерованих рецептів на відповідність базовим кулінарним принципам та загальній логіці приготування страв. Оцінювалось, наскільки правильно поєднуються інгредієнти та чи дотримано логіку етапів приготування. Це важливо, щоб переконатися, що система не буде пропонувати сумнівні комбінації, як-от молоко + огірки (рис. 3.9).

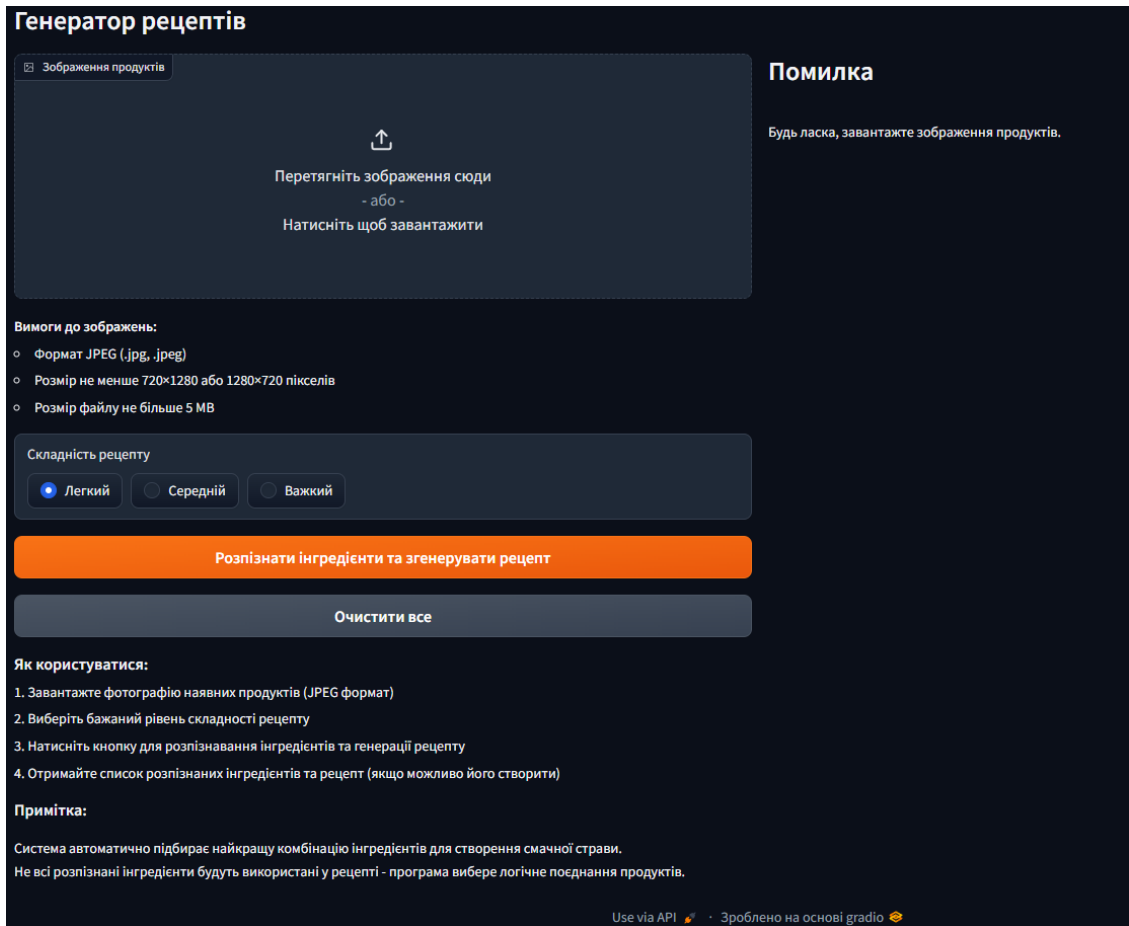


Рисунок 3.5 – Повідомлення, коли не обрано зображення

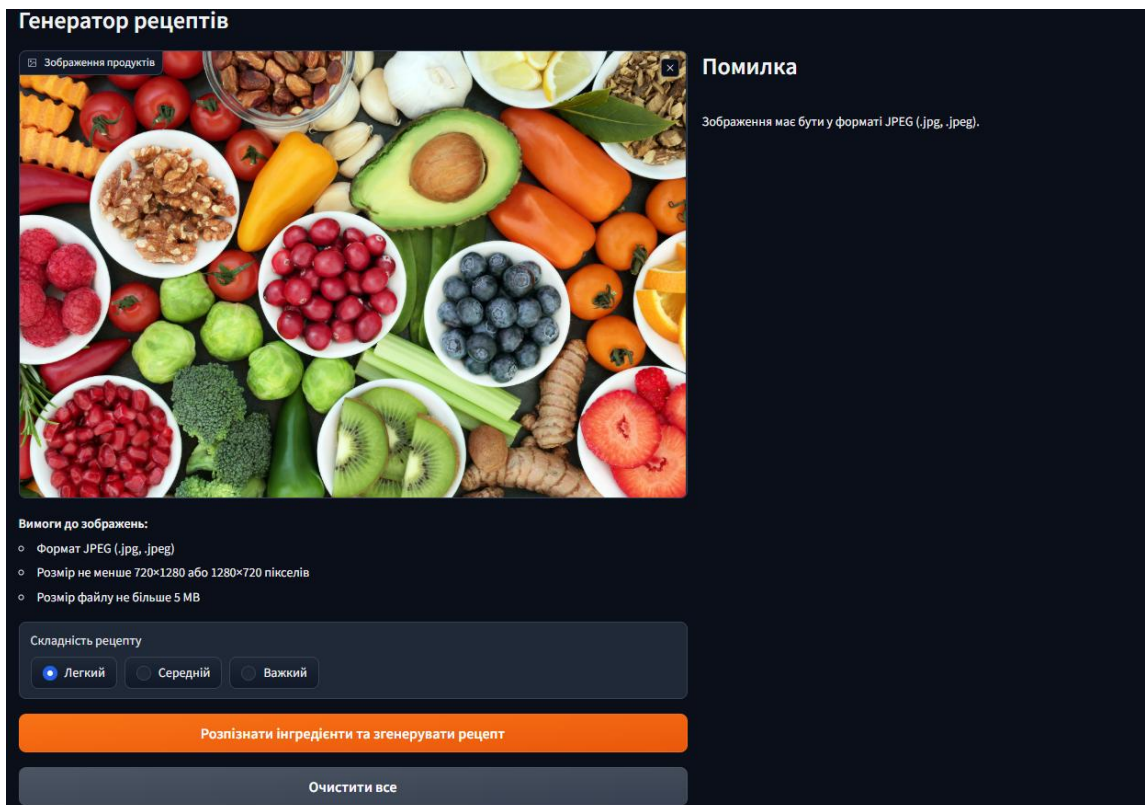


Рисунок 3.6 – Повідомлення, коли обрано інший формат

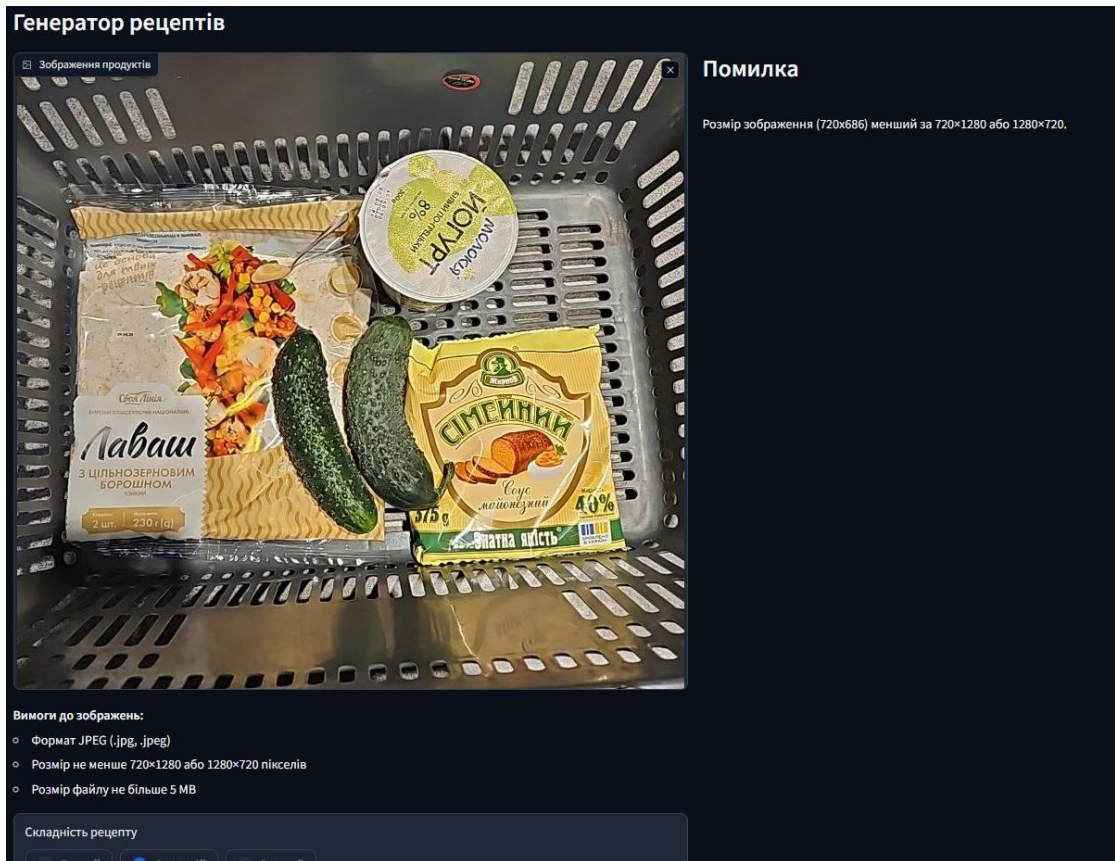


Рисунок 3.7 – Повідомлення, коли обрано зображення з малою роздільною здатністю

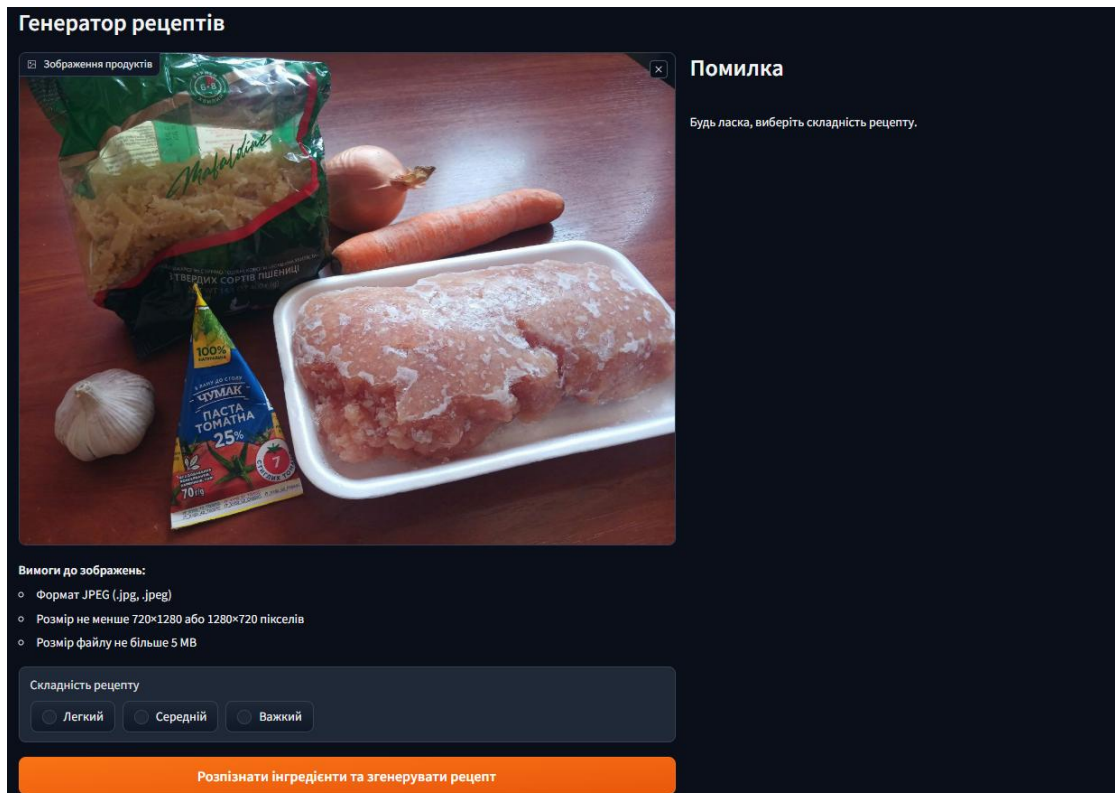


Рисунок 3.8 – Повідомлення, коли не обрано складність

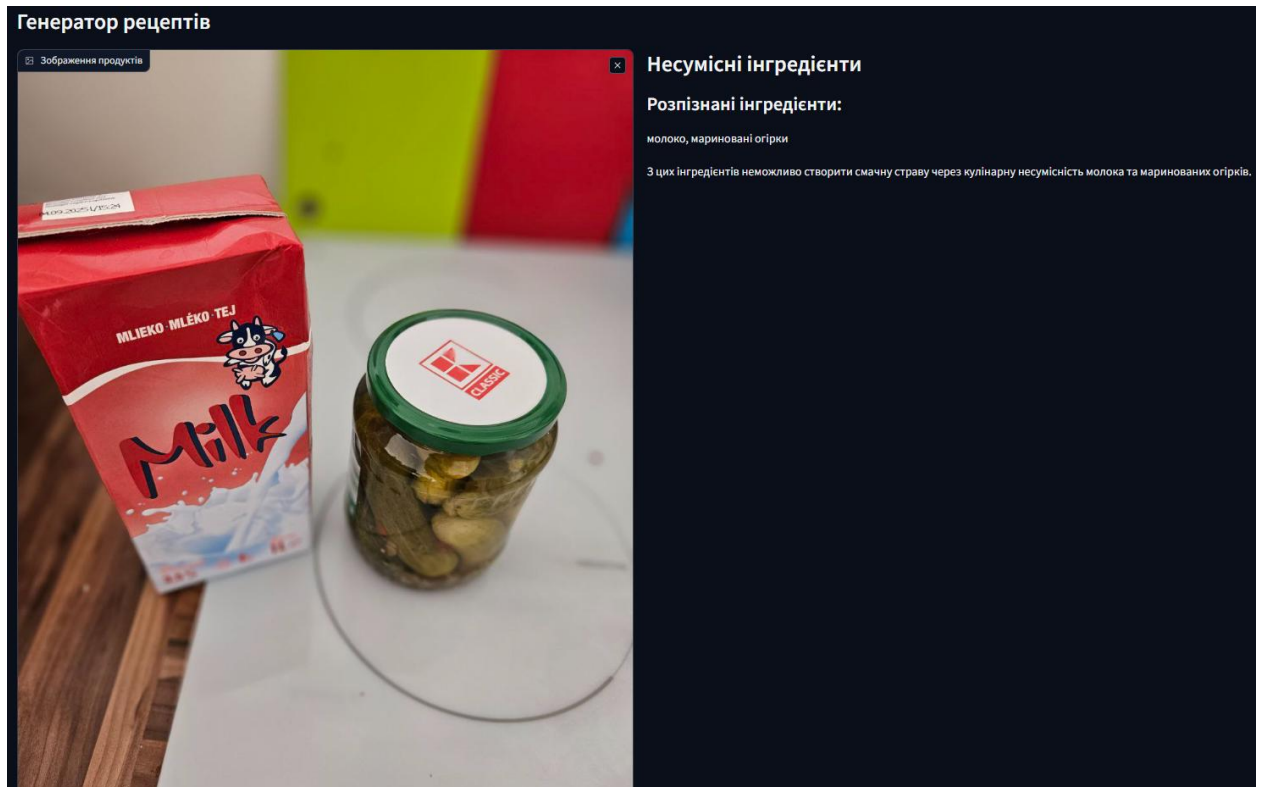


Рисунок 3.9 – Повідомлення, коли неможливо створити рецепт через несумісність інгредієнтів

Результати тестування дозволили переконатися, що інтерфейс коректно реагує на ситуації, як-от завантаження зображення у невірному форматі, файл занадто малого розміру або відсутність обраного рівня складності рецепта. А також, що система не лише технічно справна, а й генерує змістовні та практичні рецепти.


Окремо було проведено тестування на розпізнавання в поганих умовах (недостатнє освітлення та розмитий фокус), щоб оцінити важливість якості вхідного зображення (рис. 3.10, рис. 3.11).

На рисунку 3.9 видно, що модель через тінь розпізнає чотири крабові палички замість трьох, а на рисунку 3.10 – шинка замість сальтисону через розмитий фокус зображення.

Оскільки розпізнавання є ключовим етапом, то далі наступний етап генерації працює гірше, бо модель для рецепта використовує хибні продукти або неправильну їхню кількість. Саме тому так важлива якість завантажених зображень.

Генератор рецептів

Зображення продуктів



Рецепт

**Розпізнані інгредієнти:**

4 шт крабові палички, 1 шт яйце, 1 шт перець, гірчиця, твердий сир, рамен

**Рамен з крабовими паличками, яйцем і перцем**

**Інгредієнти:**

- 4 шт крабові палички
- 1 шт яйце
- 1 шт болгарський перець
- 1 ч.л. гірчиці
- 50 г твердого сиру
- 1 упаковка рамен
- 2 ст.л. рослинної олії
- 500 мл води
- сіль за смаком
- чорний перець мелений за смаком

**Інструкції:**


1. Налийте 500 мл води в каструлю, доведіть до кипіння (5 хв).
2. Відкрийте упаковку рамен, опустіть ложку в киплячу воду, варіть 4-5 хвилини до готовності. Злийте більшість бульйону, залишивши ~2 ст. л. (загалом 5 хв).
3. Поки вариться рамен, очистіть перець від насіння, наріжте тонкою соломкою (2 хв), а крабові палички – на скибочки завтовшки 0,5 см (2 хв).
4. У невеликій мисці збийте яйце з дрібною солю та перцю (1 хв).
5. На середньому вогні розігрійте велику сковороду з 2 ст.л. олії. Влийте збите яйце та швидко зробіть м'який омлет, постійно помішуючи лопаткою (2-3 хв). Перекладіть його на тарілку.
6. У ту ж сковороду покладіть порізані перець і крабові палички, обсмажуйте 3-4 хв, щоб перець став м'яким, а палички – теплими.
7. Поверніть у сковороду омлет, додайте рамен і залишений бульйон, ретельно перемішайте усе разом (2 хв).
8. Додайте 1 ч.л. гірчиці, ще раз перемішайте. Складіть суміш у каструлю чи сотейник, посипте зверху тертим сиром, накрийте кришкою і потримайте на мінімальному вогні 2 хв до розплавлення сиру.
9. Подавайте гарячим, приправивши за потреби сіллю й перцем.

**Інформація:**

Рисунок 3.10 – Фотографія з поганим освітленням

Генератор рецептів

Зображення продуктів



Рецепт

**Розпізнані інгредієнти:**

2 шт помідори, 4 шт огірки, капуста, шинка, твердий сир, майонез

**Теплий овочевий салат з шинкою та сиром**

**Інгредієнти:**

- 200 г капусти
- 2 шт помідори
- 4 шт огірки
- 150 г шинки
- 100 г твердого сиру
- 3 ст.л. майонезу
- 1 ст.л. рослинної олії
- щіпка цукру
- сіль за смаком
- перець за смаком

**Інструкції:**

1. Підготовка капусти (5 хв): тонко нашаткуйте капусту. Посипте її щіпкою цукру, сіллю й полийте ½ ст.л. олії. Ретельно помніть руками, щоб капуста трохи розм'якшлася.
2. Обсмажування шинки (7 хв): наріжте шинку смужками завтовшки 0,5-1 см. Розігрійте на середньому вогні ½ ст.л. олії й обсмажте шинку до легкої рум'яної скоринки.
3. Нарізка овочів (5 хв): помийте й наріжте помідори кубиками, огірки – тонкими півкільцями.
4. Збір салату (3 хв): у великій мисці змішайте капусту з помідорами та огірками. Додайте обсмажену шинку.
5. Заправка (2 хв): заправте майонезом, посоліть і поперчіть за смаком. Добре перемішайте.
6. Фінальний штрих (2 хв): натріть зверху салату твердий сир, щоб він трохи підтанув від тепла.
7. Подача (1 хв): дайте страві постояти 2-3 хвилини під кришкою, потім розкладіть по тарілках.

**Інформація:**

- Загальний час приготування: 35 хвилин
- Кількість порцій: 3

**Подача:**

Вимоги до зображень:

- Формат JPEG (.jpg, .jpeg)
- Розмір не менше 720x1280 або 1280x720 пікселів
- Розмір файлу не більше 5 МВ

Складність рецепту

Легкий  Середній  Важкий

Розпізнати інгредієнти та згенерувати рецепт

Очистити все

Рисунок 3.11 – Фотографія з розмитим фокусом

### 3.4 Перспективи подальшої роботи

У ході реалізації вебзастосунку для розпізнавання сирих або пакованих продуктів харчування з фотографій та подальшої генерації рецептів було успішно впроваджено всі основні функціональні компоненти. Система працює стабільно, забезпечує якісне розпізнавання інгредієнтів за допомогою мультимодальної моделі та генерує рецепти відповідно до заданої складності. Водночас проєкт має потенціал для подальшого вдосконалення та розширення функціональності за допомогою інших технологій [20–34].

Одним із важливих напрямів розвитку є створення мобільної версії застосунку. Це дозволить користувачам завантажувати фото продуктів безпосередньо з камери смартфона, що суттєво підвищить зручність використання системи в реальних умовах.

Перспективним також є вдосконалення логіки генерації рецептів з урахуванням індивідуальних параметрів користувача – обмеження за калорійністю або максимальний час приготування. У цьому випадку модель буде враховувати як набір продуктів на фото, так і вказані обмеження, формуючи рецепт, який водночас є і реалістичним, і релевантним для конкретного користувача.

Не менш важливим завданням у подальшій роботі є створення особистого кабінету користувача з можливістю збереження улюблених рецептів, перегляду історії запитів, формування списку покупок тощо.

Із технічного боку перспективу має вдосконалення алгоритму перевірки вхідних даних – зокрема, автоматичне визначення якості зображення ще до надсилання в модель, а також виявлення некоректного формату, недостатньої чіткості або надмірної кількості об'єктів на фото.

Окремим напрямом розвитку може стати інтеграція спеціалізованої моделі для розпізнавання сирих продуктів, навченої на власному наборі даних. Це дозволить досягти ще більшої точності при класифікації продуктів та забезпечити стабільну якість розпізнавання незалежно від умов зйомки.

Таким чином, подальша робота над вебзастосунком відкриває широкі можливості для розвитку, що дозволить не лише покращити якість продукту, але й адаптувати його до потреб ширшого кола користувачів, у тому числі й комерційного впровадження.

## ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований вебзастосунок для розпізнавання продуктів харчування та подальшої генерації рецептів. Робота охоплює повний цикл створення застосунку – від аналізу аналогічних рішень до реалізації функціоналу із використанням сучасних інструментів.

Розглянуто існуючі вебзастосунки зі схожим функціоналом та проведено аналіз наукових джерел, що стосуються задач розпізнавання харчових продуктів. Також було проаналізовано існуючі методи розпізнавання, особливо ті, які працюють саме з сирими або пакованими продуктами, а не з готовими стравами.

Для реалізації розпізнавання продуктів було використано модель Gemini 2.0 Flash, а для генерації рецептів – o4-mini. У якості інтерфейсу було обрано бібліотеку Gradio, а вся логіка реалізована мовою програмування Python.

Проведено тестування працездатності системи, зокрема перевірку вхідних зображень на відповідність технічним вимогам, а також аналіз логічності згенерованих рецептів.

Розглянуто перспективи покращення системи, які включають у себе створення мобільної версії застосунку, вдосконалення логіки генерації рецептів з урахуванням індивідуальних параметрів користувача, можливість створення особистого кабінету, а також вдосконалення алгоритму перевірки вхідних даних.

Результати кваліфікаційної роботи можуть бути корисними у сфері автоматизації побутових рішень, а також при подальшому створенні інтелектуальних кулінарних помічників. Результати роботи апробовано у вигляді тез доповіді під час Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У ХХІ СТОЛІТТІ» [35].

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Зберігайте, публікуйте та діліться рецептами з кулінарами з усього світу - Cookpad. URL: <https://cookpad.com/ua> (дата звернення 09.04.2025).
2. Підбір рецептів. *ONLINE.UA*. URL: <https://retsepty.online.ua/ukr/podbor-retsepta/> (дата звернення 09.04.2025).
3. MyFridgeFood - Home. URL: <https://www.myfridgefood.com/> (дата звернення 09.04.2025).
4. Supercook: recipe search by ingredients you have at home. *SuperCook - Zero Waste Recipe Generator*. URL: <https://www.supercook.com/#/desktop> (дата звернення 09.04.2025).
5. Abiyev, R., & Adepoju, J. (2024). Automatic Food Recognition Using Deep Convolutional Neural Networks with Self-attention Mechanism. *Human-Centric Intelligent Systems*, 4(1), 171-186.
6. Hannon, B., Kumar, Y., Li, J. J., & Morreale, P. (2024). Chef Dalle: Transforming Cooking with Multi-Model Multimodal AI. *Computers*, 13(7), 156.
7. Lo, F. P. W., Qiu, J., Wang, Z., Chen, J., Xiao, B., Yuan, W., ... & Lo, B. (2024). Dietary Assessment with Multimodal ChatGPT: A Systematic Analysis. *IEEE Journal of Biomedical and Health Informatics*.
8. Şener, E., & Ulu, E. K. (2024). Culinary Innovation: Will the Future of Chefs' Creativity be Shaped by AI Technologies? *Tourism: An International Interdisciplinary Journal*, 72(3), 340-352.
9. Yang, Z., Khatibi, E., Nagesh, N., Abbasian, M., Azimi, I., Jain, R., & Rahmani, A. M. (2024). ChatDiet: Empowering personalized nutrition-oriented food recommender chatbots through an LLM-augmented framework. *Smart Health*, 32, 100465.
10. Dhote, D., Bhoyar, M., Mange, K., Dhote, A., & Keswani, V. (2024). Nutri Flow: Guiding your Nutritional Journey. *Grenze International Journal of Engineering & Technology (GIJET)*, 10.

11. Kansaksiri, P., Panomkhet, P., & Tantisuwichwong, N. (2023). Smart Cuisine: Generative recipe & ChatGPT powered nutrition assistance for sustainable cooking. *Procedia Computer Science*, 225, 2028-2036.

12. Hidaka, A., & Kurita, T. (2017, May). Consecutive dimensionality reduction by canonical correlation analysis for visualization of convolutional neural networks. In *Proceedings of the ISCIE international symposium on stochastic systems theory and its applications* (Vol. 2017, pp. 160-167). The ISCIE Symposium on Stochastic Systems Theory and Its Applications.

13. Muresan, H., & Oltean, M. (2018). Fruit recognition from images using deep learning. *Acta Univ. Sapientiae*, 10(1), 26-42.

14. Khan, Z. A., Waqar, M., Cheema, K. M., Mahmood, A. A. B., Ain, Q., Chaudhary, N. I., ... & Raja, M. A. Z. (2024). EA-CNN: Enhanced attention-CNN with explainable AI for fruit and vegetable classification. *Heliyon*, 10(23), e40820.

15. Guo, C., Ai, W., Hu, S., Du, X., & Chen, N. (2022). Sea surface wind direction retrieval based on convolution neural network and wavelet analysis. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15, 3868-3876.

16. Savani, N., & Lunagaria, M. (2022). Detection of Grocery Items in Store Using Deep Learning in Retail Business. In *ICT Systems and Sustainability: Proceedings of ICT4SD 2022* (pp. 257-266). Singapore: Springer Nature Singapore.

17. Naminas K. Instance segmentation: identify and classify objects precisely. *Label Your Data – Your Trusted Data Annotation Services Provider*. URL: <https://labeleyourdata.com/articles/instance-segmentation> (дата звернення 18.04.2025).

18. Janssen, D. (2024). *Product Recognition in Store Environments: A Deep Learning Approach* (Master's thesis, University of Twente).

19. Zero-shot Image Classification with OpenAI's CLIP | Pinecone. URL: <https://www.pinecone.io/learn/series/image-search/zero-shot-image-classification-clip/> (дата звернення 19.04.2025).

20. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40-48.

21. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.

22. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2024) Improving the effectiveness of image classification structural methods by compressing the description according to the information content criterion, *Computers, Materials & Continua*, vol. 80, no. 2, pp. 3085-3106.

23. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 113-125.

24. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9), pp. 64-72.

25. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.

26. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.

27. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, *International Journal of Academic and Applied Research*, 7(9), pp. 57-70.

28. Gorokhovatskyi V., Chmutov Y., Tvoroshenko I., and Kobylin O. (2025) Reducing computational costs by compressing the structural description in image classification methods, *Advanced Information Systems*, vol. 9, no. 1, pp. 5–12.
29. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, *International Journal of Academic and Applied Research*, 7(11), pp. 134-145.
30. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine*, pp. 25-27.
31. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, vol. 11, pp. 126938-126949.
32. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, *IEEE Access*, vol. 12, pp. 73376-73385.
33. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2025) Image description compression in classification structural methods, *IEEE Access*, vol. 13, pp. 43631-43641.
34. Yakovleva O., Matúšová S., Tvoroshenko I., and Isaiev Y. (2024) Visitor counting based on video stream analysis from surveillance cameras to solve various business problems, *Verejná správa a regionálny rozvoj ekonómia, manažment a marketing*, XX(1), pp. 67-87.
35. Боцвін О.С. (2025) Аналіз особливостей сучасних застосунків для пошуку рецептів за продуктами харчування. *Радіоелектроніка і молодь у XXI столітті: тези доповідей 29-го Міжнародного молодіжного форуму (Харків, 16–19 квітня 2025 р.)*. Харків: ХНУРЕ, 2025. Т. 7. С. 20-22.