ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Інформаційна система підбору рецептів на основі мультимодальних алгоритмів інтелектуального аналізу даних»

на здобуття освітнього ступеня бакалавра
зі спеціальності 126 Інформаційні системи та технології
освітньо-професійної програми Інформаційні системи та технології

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ _____
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. ІСД-42
Глєб КАЛІНІЧЕНКО
Ім'я, ПРІЗВИЩЕ

Керівник:
PhD Дмитро КОЗЛОВ,
Ім'я, ПРІЗВИЩЕ

Рецензент:
науковий ступінь, вчене звання
Ім'я, ПРІЗВИЩЕ

Київ 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційних систем та технологій
Ступінь вищої освіти бакалавр
Спеціальність 126 Інформаційні системи та технології
Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ
Завідувач кафедрою ІСТ
_____Каміла СТОРЧАК
«____» _____2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Калініченко Глєб Олегович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Інформаційна система підбору рецептів на основі мультимодальних алгоритмів інтелектуального аналізу даних

керівник кваліфікаційної роботи Дмитро КОЗЛОВ, PhD,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій
від «20» 02 2026р. № 51

2. Строк подання кваліфікаційної роботи «01» 06 2026р.

3. Вихідні дані до кваліфікаційної роботи: технічна документація до Gemini API, література з мультимодальних моделей штучного інтелекту, промпт-інжинірингу, розпізнавання об'єктів, рекомендаційних систем та веброзробки.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
   1. Аналіз сучасних підходів до застосування мультимодальних моделей у системах підбору рецептів.
   2. Проєктування інформаційної системи та алгоритмів розпізнавання інгредієнтів.
   3. Реалізація вебзастосунку, тестування та оцінка точності роботи алгоритмів.

5. Перелік ілюстративного матеріалу: *презентація*

6. Дата видачі завдання «20» 02 2026р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області та постановка задачі дослідження.	12.03.2026	
2	Огляд мультимодальних моделей і сервісів підбору рецептів.	21.03.2026	
3	Вибір технологій та проєктування архітектури системи.	23.03.2026	
4	Розробка алгоритмів обробки зображень і промпт-інжинірингу.	14.04.2026	
5	Програмна реалізація вебзастосунку та інтеграція Gemini API.	11.05.2026	
6	Розробка демонстраційних матеріалів.	15.05.2026	
7	Попередній захист дипломної роботи.	19.05.2026	
8	Подання роботи в деканат.	01.06.2026	

Здобувач(ка) вищої освіти _____ Глєб КАЛІНІЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник
кваліфікаційної роботи _____ Дмитро КОЗЛОВ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 64 стор., 29 рис., 4 табл., 31 джерел.

Мета роботи – підвищення точності ідентифікації харчових продуктів за зображенням шляхом використання мультимодальних моделей штучного інтелекту в вебзастосунку підбору рецептів.

Об'єкт дослідження – системи автоматичного підбору рецептів на основі аналізу інгредієнтів.

Предмет дослідження – метод підвищення точності ідентифікації харчових продуктів із використанням мультимодальних моделей штучного інтелекту.

Короткий зміст роботи: У роботі використано методи системного аналізу для вивчення предметної області, методи промпт-інжинірингу для взаємодії з мультимодальними моделями, методи об'єктно-орієнтованого проектування для створення архітектури системи та експериментальні методи для оцінки точності розпізнавання. У ході виконання роботи було спроектовано та реалізовано вебзастосунок для підбору рецептів. Вдосконалено метод ідентифікації продуктів через впровадження механізмів ітераційного промпт-інжинірингу та декомпозиції візуальної сцени. Експериментально підтверджено, що використання верифікації та розподілу зображення за зонами дозволяє суттєво підвищити точність розпізнавання інгредієнтів у складних некооперативних середовищах порівняно зі стандартними підходами.

КЛЮЧОВІ СЛОВА: МУЛЬТИМОДАЛЬНІ МОДЕЛІ, ШТУЧНИЙ ІНТЕЛЕКТ, GEMINI API, РОЗПІЗНАННЯ ОБ'ЄКТІВ, ПІДБІР РЕЦЕПТІВ, ВЕБЗАСТОСУНОК, ПРОМПТ-ІНЖИНІРИНГ

ABSTRACT

Textual part of the qualification work for obtaining a bachelor's degree: 64 pages, 29 figs., 4 tables, 31 sources.

The purpose of the work is to improve the accuracy of food identification from images by using multimodal artificial intelligence models in a web application for recipe selection.

The object of research is automated recipe recommendation systems based on ingredient analysis.

The subject of research is a method for improving the accuracy of food identification using multimodal artificial intelligence models.

Summary of the work: The study employed system analysis methods to examine the subject area, prompt engineering methods for interaction with multimodal models, object-oriented design methods for developing the system architecture, and experimental methods for evaluating recognition accuracy. As a result of the work, a web application for recipe selection was designed and implemented. The method of food identification was improved through the introduction of iterative prompt engineering mechanisms and visual scene decomposition. Experimental results confirmed that the use of verification and image segmentation into zones significantly improves the accuracy of ingredient recognition in complex non-cooperative environments compared with standard approaches.

KEYWORDS: MULTIMODAL MODELS, ARTIFICIAL INTELLIGENCE, GEMINI API, OBJECT RECOGNITION, RECIPE RECOMMENDATION, WEB APPLICATION, PROMPT ENGINEERING.

ЗМІСТ

ВСТУП

Швидкий розвиток мультимодальних моделей штучного інтелекту відкриває нові можливості для автоматизації побутових процесів, зокрема у сфері харчування. Проблема точного розпізнавання вмісту холодильника в неструктурованих умовах залишається критичною для систем підбору рецептів, що зумовлює необхідність розробки нових підходів до ідентифікації об’єктів.

Мета роботи полягає у підвищенні точності ідентифікації харчових продуктів за зображенням шляхом використання мультимодальних моделей штучного інтелекту в вебзастосунку підбору рецептів.

Для досягнення поставленої мети було визначено такі завдання:

1. Проаналізувати сучасні підходи до використання інтелектуальних алгоритмів у системах ідентифікації продуктів.
2. Дослідити можливості мультимодальних моделей (зокрема Gemini) для аналізу візуальних даних.
3. Розробити метод ітераційного промпт-інжинірингу та декомпозиції сцени для покращення результатів розпізнавання.
4. Спроектувати та реалізувати архітектуру вебзастосунку для підбору рецептів.
5. Провести експериментальне дослідження ефективності запропонованого методу на основі створеного датасету.

Об’єктом дослідження є процеси автоматичного підбору рецептів на основі аналізу інгредієнтів.

Предметом дослідження є методи та алгоритми підвищення точності ідентифікації харчових продуктів із використанням мультимодальних моделей штучного інтелекту.

Наукова новизна одержаних результатів полягає у вдосконаленні методу взаємодії з мультимодальними моделями через механізм верифікації та декомпозиції зображення, що дозволяє зменшити кількість помилко ідентифікацій у порівнянні зі стандартними методами одноразового сканування.

Практичне значення роботи. Розроблено діючий прототип вебзастосунку, який дозволяє користувачам отримувати персоналізовані рекомендації щодо страв

на основі реального вмісту їхніх холодильників. Запропонований підхід до обробки зображень може бути використаний в інших системах комп’ютерного зору, що працюють із великим скупченням дрібних об’єктів.

Апробація результатів. Основні положення та результати дослідження пройшли апробацію на науковх заходах:

1. III ВСЕУКРАЇНСЬКА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ «ТЕХНОЛОГІЧНІ ГОРИЗОНТИ: ДОСЛІДЖЕННЯ ТА ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ТЕХНОЛОГІЧНОГО ПРОГРЕСУ УКРАЇНИ І СВІТУ» (Київ, ДУІКТ, 18 листопада 2025 р.) Тези доповіді на тему «Інтерактивні веб-сайти як інструмент технологічних інновацій в освіті» опублікований у збірнику матеріалів конференції (стор. 303)

2. II ВСЕУКРАЇНСЬКА НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ ТЕХНОЛОГІЧНІ ГОРИЗОНТИ: ДОСЛІДЖЕННЯ ТА ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ТЕХНОЛОГІЧНОГО ПРОГРЕСУ УКРАЇНИ І СВІТУ (Київ, ДУІКТ, 18 листопада 2024 р.) Тези доповіді на тему «Перспективи розвитку квантових обчислень і їх вплив на світові технології» опублікований у збірнику матеріалів конференції (стор. 289)

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ЗАСТОСУВАННЯ МУЛЬТИМОДАЛЬНИХ МОДЕЛЕЙ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ

1.1 Актуальність використання інтелектуальних алгоритмів у веб-застосунках підбору рецептів

В еру, коли інтернет інтегрований в кожну частину нашого життя не дивовижно, що прогрес не буде тільки на цьому зупинятись. За останні 4 роки штучний інтелект став однією з найпривілегійованіших галузей для вивчення та праці. Також не дивовижно, що ми почали використовувати штучний інтелект в побуті.

За останніми даними 2025 року 78% компаній використовують штучний інтелект як мінімум в одному бізнес-процесі. Так само не дивовижно, що приблизно 66% людей використовують його постійно для вирішення питань. Приблизно 1.8 мільярдів людей в світі використовують інструменти з застосуванням штучного інтелекту хоча б один раз [1].

Саме тому штучний інтелект інтегрується в наш стиль життя, включаючи харчування. Для людей є постійна проблема в розумінні, що саме вони бажають їсти на сніданок, обід та вечерю, проблема не тільки в знаходженні рецептів, але й їх інтеграції з дієтичними обмеженнями людини. На жаль, даних по дієтичним обмеженням для людей усіх країн немає, але можливо розглянути загальні тенденції харчування людей в тих країнах, які провели ці дослідження.

В США алергія на харчові продукти вражає 6 мільйонів дітей до 18 років. Тільки в США реєструється приблизно від 150 до 200 смертей на рік від харчової алергічної анафілаксії [2]. Схожі дані не тільки в Америці, але й в інших країнах світу. В Англії проаналізували дані за період з 1998 по 2018 рік, щоб зрозуміти, як система охорони здоров'я діагностує та лікує харчові алергії. Дані показали що рівень ймовірних харчових алергій зріс з 75,8 випадків на 100 000 осіб на рік у 2008 році до 159 500 у 2018 році [3].

На рис. 1.1 показано, як змінювалися показники захворюваності та поширеності харчової алергії в Англії у період з 2008 по 2018 роки. Загалом графіки наочно демонструють кілька важливих тенденцій. Лівий графік, який відображає частоту появи нових випадків (incidence rate), свідчить про помітне зростання харчової алергії. Лінії, що відповідають «можливій» та «ймовірний» алергії, піднімаються досить стрімко.

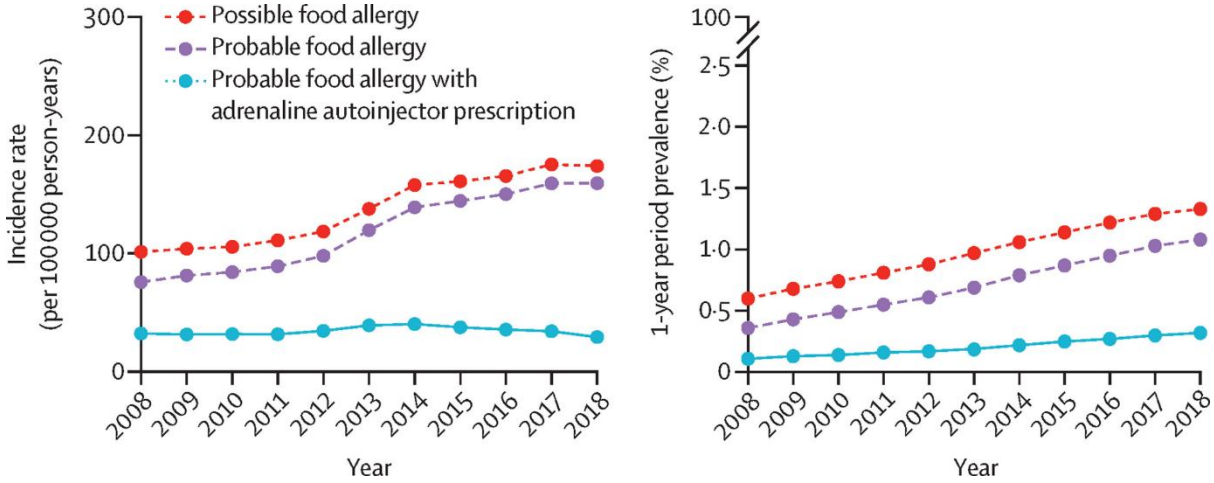

Рис. 1.1 Показники захворюваності та поширеності харчової алергії в Англії [3]

Особливо показовим є те, що рівень ймовірної харчової алергії майже подвоївся: якщо у 2008 році він становив близько 75 випадків на 100 000 людино-років, то у 2018 році перевищив 150. Водночас кількість тяжких випадків, які потребують призначення автоін’єкторів адреналіну, залишається відносно стабільною протягом усього періоду. Це може означати, що, хоча людей з харчовою алергією стає більше, частка пацієнтів із найважчими реакціями суттєво не зросла.

Правий графік відображає річну поширеність (1-year period prevalence), тобто загальну частку населення, яка мала діагноз харчової алергії протягом року. Тут чітко видно поступове, майже лінійне зростання. Поширеність ймовірної алергії збільшується приблизно з 0,4% до понад 1,0%. Ці дані вказують на зростаюче навантаження на систему охорони здоров’я, адже кількість

діагностованих випадків харчової алергії стабільно зростає з року в рік, що узгоджується з наведеними статистичними даними [3].

В Німеччині був проведений аналіз аналітики при пошуку симптомів та ліків від харчових алергій. Починаючи з вересня 2022 до жовтня 2024 було всього 3 649 390 запитів на ці теми. Згідно з графіком 1.2, який показує, як змінювався інтерес до ліків від харчових алергій у Німеччині, можна побачити чітку картину вподобань користувачів і помітну сезонність.

Упродовж усього періоду дослідження найбільше людей цікавилися питаннями, пов’язаними з алергією на горіхи. Саме ця категорія стабільно лідирує за кількістю пошукових запитів і вирізняється найрізкішими коливаннями. Найвищий інтерес спостерігався взимку 2024 року, коли кількість запитів перевищила 80 тисяч. На другому місці за популярністю перебувають запити, пов’язані з гістаміном. Інтерес до цієї теми зростав поступово: якщо на початку періоду йшлося про 25-30 тисяч запитів, то навесні 2024 року їхня кількість уже перевищила 40 тисяч.

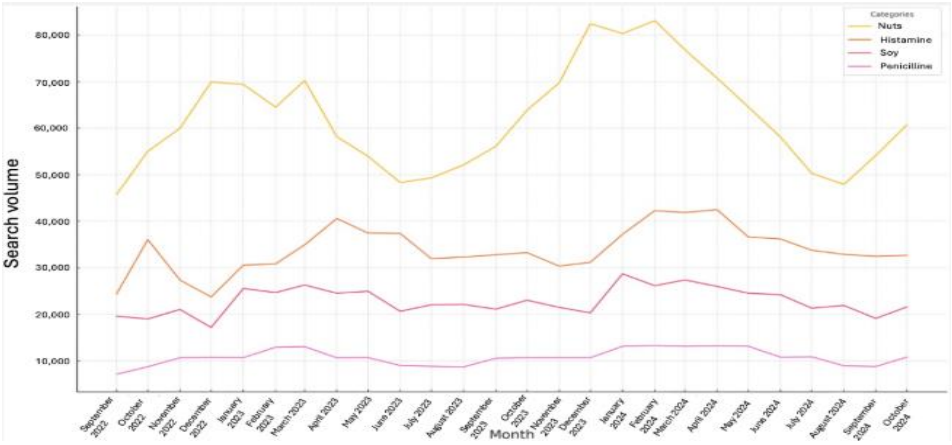

Рис. 1.2 Кількість пошуків алергій в Німеччині [4]

Третю позицію займає алергія на сою – тут показники були відносно стабільними, з невеликим зростанням на початку 2024 року. Найменше уваги користувачі приділяли темі пеніциліну: рівень пошуку залишався низьким і майже не змінювався, тримаючись близько 10 тисяч запитів. Якщо подивитися на всі

криві загалом, стає очевидною спільна тенденція: взимку та на початку весни інтерес до ліків від харчових алергій зростає, а влітку – помітно знижується.

Це означає, що пошук інформації щодо лікування алергій має циклічний характер. Попри те, що загальна кількість запитів за весь період перевищує 3,6 мільйона, найбільше навантаження на інформаційні ресурси та аптеки припадає саме на холодні місяці, коли змінюється харчування та знижується стійкість імунної системи [4].

Спираючись на дані, можна зробити висновок, що наявність базових знань у сфері харчування є важливою вимогою для збереження свого здоров'я і здоров'я близьких людей. Саме тому, більшість користувачів використовують веб-застосунки з інтегрованими алгоритмами штучного інтелекту для персоналізованого підбору харчування та рецептів. Такий підхід є особливо актуальним, оскільки не всі мають фізичний чи фінансовий доступ до консультацій професійного дієтолога.

Водночас, використання штучного інтелекту в харчових рекомендаційних системах супроводжується великою кількістю технічних і методологічних проблем. Фреймворки на основі розпізнавання зображень часто залежать від великих, специфічних для певної галузі наборів даних. Для побудови своєї бази знань вони вимагають значного збору даних для забезпечення прийнятної результативності роботи.

Ця залежність від величезних наборів даних створювала проблеми з точки зору масштабованості, різноманітності та адаптивності до нових сценаріїв. Інша проблема виникає при використанні тестових методів, користувачеві може бракувати чіткості щодо бажаних рецептів. Для вирішення таких проблем пропонують використання гібридних систем. Поєднання таких систем як візуально-мовні моделі (надалі в тексті VLMs, від англійського Vision-Language Models) та Генерація з підкріпленням інформаційним пошуком (надалі в тексті RAG, від англійського Retrieval-Augmented Generation) представляють собою ключові досягнення в цій галузі.

Завдяки VLM-моделям системи навчилися "бачити" фото інгредієнтів і розуміти запити користувачів. Проте, їм усе ще важко давати точні поради в складних ситуаціях — наприклад, коли треба суворо врахувати алергії чи специфічні медичні обмеження, де помилка неприпустима

Для цього використовують RAG фреймворки, як додатковий шар, що забезпечує можливість, щоб рекомендовані рецепти не тільки відповідали очікуванням користувачів, але й підтримували узгодженість з доступними інгредієнтами, дієтичними обмеженнями та кулінарними навичками. В кінцевому підсумку, таким чином заповнюється діра між вхідними даними користувача та вихідними даними моделі [5].

З огляду на зазначене, інтеграція штучного інтелекту є суттєвою складовою алгоритмів веб-застосунків для підбору рецептів, оскільки вона забезпечує врахування контекстних даних і індивідуальних обмежень користувачів.

1.2 Огляд популярних мультимодальних моделей для розпізнавання об’єктів

Розглянемо декілька прикладів мультимодальних моделей для розпізнавання об’єктів, які сьогодні найчастіше застосовуються при інтеграції в веб-застосунки. Але спочатку варто розглянути як мультимодальні моделі дійшли до цієї точки в розвитку, а також проблеми, з якими вони можуть стикатися.

Поява великих мовних моделей або LLMs стала неймовірним змінюючим періодом для штучного інтелекту чи, навіть, головною зміною всієї галузі. Велика кількість академіків та дослідників були активно залучені в розвиток LLMs. Але виникла велика проблема в таких моделях, вони обмежені обробкою лише одного типу даних. Обробка декількох модальних даних одночасно є важливою частиною інноваційного прогресу для подальших відкриттів у сфері штучного інтелекту [6].

Природний інтелект – це форма інтелекту, яка виникає в біологічних системах в результаті еволюції та втілена в нейронні процеси. Він включає в себе здатність організмів – від простих форм життя, таких як бактерії, до людей –

відчувати та інтерпретувати своє оточення, обробляти інформацію, адаптуватися шляхом навчання, приймати рішення та проявляти цілеспрямовану поведінку [7].

Для того, щоб штучний інтелект міг емулювати людські когнітивні здібності, він повинен повністю повторити мозкову мультимодальну обробку даних. Фундаментом для більшості таких систем є архітектура Transformer показаний на рис. 1.3, яка завдяки механізму самоуваги дозволяє моделі визначати контекстуальні зв'язки між елементами даних.

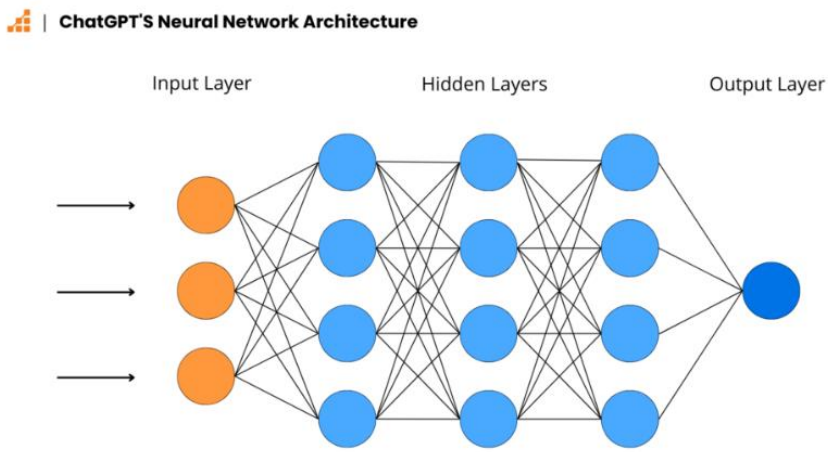

Рис. 1.3 Базова архітектура моделі Transformer: механізми енкодера, декодера та самоуваги [8]

У відповідь на такі обмеження штучного інтелекту, дослідники стали піонерами у створенні новітнього класу нейронних моделей, відомих як візуально-мовні моделі (надалі в тексті VLMs, від англійського Vision-Language Models). Принципову архітектуру цих систем, що демонструє ключові компоненти та методи їх інтеграції, наведено на рис. 1.4. Як видно зі схеми, ефективність VLM залежить від стратегії поєднання візуального енкодера та мовної моделі через шари злиття, де певні параметри залишаються "замороженими" для збереження знань, а інші – навчаються для адаптації до нових даних.

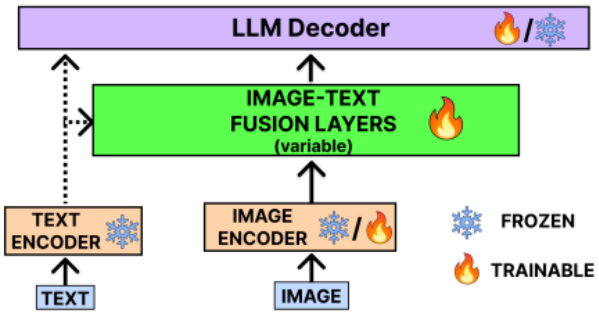

Рис. 1.4 Схема поєднання візуального енкодера та мовної моделі через шари злиття [6]

Візуально-мовні моделі розроблені для виконання таких завдань, як відповіді на візуальні запити, створення підписів до зображень, мультимодальне узагальнення, створення інфографіки, мультимодальний пошук та інше. Їхня досконала взаємодія візуальних і мовних модальностей виводить ці системи на передову технологічного розвитку, надаючи їм змогу з винятковою точністю працювати зі складним поєднанням зображень і тексту. Зокрема, не всі моделі мають VLMs. Для прикладу моделі перетворення тексту в зображення такі як Midjourney та DALL-E не мають компонента генерації мови, що підкреслюють різноманітність мультимодального ландшафту штучного інтелекту [6]. Дані моделі мають також велику кількість проблем. Одна з домінуючих проблем в більшості моделей це галюцинації.

У генеративному моделюванні галюцинація ШІ – це результат, створений моделлю, який виходить за межі інформації, підтвердженої вхідними або навчальними даними, та призводить до появи не правдоподібного, суперечливого або неіснуючого контенту. У контексті великих мовних моделей галюцинації означають згенерований контент, що є беззмістовним або неузгодженим із надійним джерелом і проявляється як порушення фактичної достовірності або невідповідність вихідним даним [9].

Актуальність цієї проблеми підтверджується стрімким зростанням кількості наукових публікацій, присвячених галюцинації у мультимодальних моделях. За

даними моніторингу Google Scholar (це показано на рис. 1.5), кількість релевантних робіт демонструє експоненційне зростання у період з 2021 по 2025 роки.

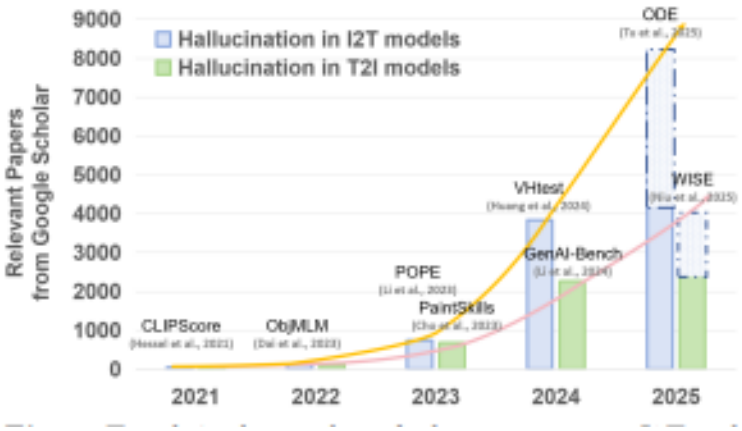

Рис. 1.5 Кількість релевантних робіт опубліковано на Google Scholar [10]

Особливо помітним є сплеск досліджень у сфері моделей «зображення-у-текст» (в таблиці написано I2T, від англійського Image to Text), де у 2025 році кількість публікацій майже досягла 9000, перевищуючи обсяг досліджень галюцинацій у моделях «текст-у-зображення» (в таблиці написано T2I, від англійського Text to Image) [10].

Причин для такого феномену як галюцинації може бути безліч. Більшість мультимодальних моделей потребують велику кількість даних. Мала кількість даних може потенціально призвести до проблематичного міжмодального узгодження, що призводить до галюцинацій. Для тренування якісних моделей кількість даних які вони потребують постійно збільшується. Не завжди можливо гарантувати високу якість даних, а низькоякісні дані підвищують кількість галюцинацій моделі.

Навчання таких моделей складається з двох етапів: попереднє навчання з використанням шумних пар веб-зображень і текстів, що може зашкодити міжмодальному вирівнюванню, та налаштування інструкцій, де інструкції, згенеровані GPT-4, можуть спричинити додатковий шум. LLaVA-1.5 покращує

результати завдяки додаванню даних QA, анотованих людьми. При навчанні схожих моделей за інструкціями, можливе виникнення проблем, бо інструкцій, поки що, недостатньо.

Більшість таких даних – це «позитивні інструкції», тобто приклади розмов, які правильно описують зображення. Негативні інструкції зустрічаються рідко. Через нестачу різноманітних прикладів моделі часто відповідають «Так» на все, навіть якщо правильна відповідь – «Ні». Це призводить до галюцинацій, коли модель дає неточні відповіді. Рівень деталізації текстових описів у навчальних даних для мультимодальних моделей поки що залишається відкритим питанням. У навчальних даних, таких як LAION, тексти, зазвичай, описують лише головні об’єкти на зображенні, тоді як у даних для навчання за інструкціями, наприклад LLaVA-150k, описи значно детальніші й створені GPT-4 на основі об’єктів, розпізнаних моделями зору.

Деякі дослідження вказують, що навіть у таких деталізованих даних зазвичай відсутні описи розташування об’єктів, їхніх властивостей або другорядних об’єктів, що призводить до неповного поєднання тексту і зображення. Інші дослідження стверджують, що занадто деталізований опис може перевантажувати модель, через що вона може видавати деталі, яких на зображенні немає. Саме тому оптимальний рівень деталізації не був точно визначений [11].

Після розуміння причин головних проблем, можна розглянути та порівняти нові моделі. Для порівняння обрано найвідоміші та найпрогресивніші моделі станом на 2025 рік: Google Gemini 2.5, Claude 4, GPT-4.5 (Orion) та DeepSeek v3.1. Детальний функціонал та технічні особливості кожної цієї моделі наведено в таблиці 1.1.

Табл. 1.1

Порівняння різних моделей штучного інтелекту [12]

Модель	Контекстне вікно	Мультимодальна підтримка	Відкриті ваги	Ключові інновації	Сфери застосування	Обмеження
Google Gemini 2.5	100K+ токенів	Текст, фото, аудіо, відео	Ні	Режим Deep Think, Project Astra, нативний синтез аудіо	Мультимодальні асистенти, генерація коду, освіта	Закриті ваги, доступ лише через API
Claude 4	64K токенів	Лише текст	Ні	Система Extended Thinking, файлова система пам'яті	Юридичний та фінансовий аналіз, корпоративний ШІ	Обмеження текстом, закриті ваги
GPT-4.5 (Orion)	30K токенів	Текст, фото, аудіо	Ні	Інтеграція плагінів, мультимодальне введення	ШІ загального призначення, програмування, навчання	Мале контекстне вікно, відсутність відкритих ваг
DeepSeek V3.1	1 млн токенів	Текст, код, фото	Так	Архітектура Mixture-of-Experts, відкриті ваги	Дослідження, робота з великим контекстом	Високі вимоги до апаратного забезпечення

Проаналізувавши дані таблиці, можливо виділити декілька ключових тенденцій у розвитку новітніх моделей. За обсягом контенту лідером є DeepSeek v3.1 з обсягом у 1 мільйон токенів, це в свою чергу робить її найбільш придатною для обробки великої кількості інформації.

Натомість GPT-4.5 має найменше вікно, приблизно у 30 тисяч токенів, що обмежує його використання у роботі з великими джерелами. Модель Gemini 2.5 демонструє найбільш універсальний підхід, з можливістю обробляти не тільки текст та фото але й аудіо та відео, завдяки архітектурі Project Astra. Основна особливість DeepSeek v3.1, яка відрізняє його від інших моделей це наявність відкритих ваг. Завдяки цьому можливо локальне розгортання моделі для забезпечення конфіденційності даних. Всі інші конкуренти залишаються

закритими пропрієтарними системами. Також треба розглядати саме спеціалізацію кожної моделі.

Claude 4 фокусується на глибинному логічному мисленні (Extended Thinking Framework) для бізнес-аналітики, тоді як GPT-4.5 використовує екосистему плагінів для кінцевого користувача. Навіть зараз для цих моделей є деякі обмеження. Їх головне питання полягає в доступності.

Такі моделі як DeepSeek v3.1 випускаються з відкритими вагами, що в свою чергу, сприяє прозорості та розвитку спільноти дослідників, більшість провідних систем, такі як – Gemini 2.5, Claude 4, GPT-4.5, залишаються закритими. Доступ до них можливо отримати лише через API-підписки, що звужує можливість незалежної оцінки. По-друге, спостерігається нерівномірність у розвитку мультимодальних можливостей. Такі моделі як Gemini 2.5 пропонують інтегроване мультимодальне мислення, коли багато інших, як Claude 4 залишаються переважно текст-орієнтованими. Для використання мультимодальних можливостей вони потребують застосування зовнішніх інструментів [12].

Таким чином, незважаючи на стрімкий розвиток штучного інтелекту, розробники прикладних програм стикаються з вибором: використання комерційних, потужних моделей через API або гнучкі рішення з відкритим програмним забезпечення. Це безпосередньо впливає на архітектуру застосунку та доступність споживчих продуктів. Для розуміння як саме ці моделі трансформують індустрію, доцільно перейти до аналізу ринку вже існуючих рішень.

1.3 Порівняння існуючих сервісів підбору рецептів

Для аналізу вже існуючих сервісів, важливо переглянути методи які вже використовуються в більшості застосунків та визначитися, що таке системи рекомендацій рецептів. Для початку треба зрозуміти, що таке рекомендаційна система. Рекомендаційна система – це метод фільтрації інформації, яка

рекомендує предмети, в яких користувач може бути зацікавлений базуючись на його вподобаннях. З постійним зростанням даних в інтернеті, рекомендаційні системи показали себе невід'ємною частиною у вирішувані проблеми інформаційного перевантаження. Тема рекомендації рецептів дуже багатогранна, для оптимальної класифікації наявних систем рекомендацій рецептів, було виділено п'ять категорій: методи на основі контенту, методи на основі спільного фільтрування, гібридні методи, методи на основі графіків методи рекомендацій, орієнтовані на харчування та здоров'я.

Для початку розглянемо методи, засновані на контенті. Їх основна ідея полягає в рекомендації схожого контенту користувачеві, на основі особливостей історії його поведінки та спожитого контенту.

Розглянемо методи, засновані на основі спільного фільтрування. Вони поділені на два основні підходи: один на основі пам'яті, інший на основі моделі. Підхід на основі пам'яті фокусується на прямих стосунках з даними. Одна версія, на основі користувача, знаходить людей з схожим смаком та радить речі, які сподобались їм. Інша версія, на основі предметів, переглядає речі, які сподобались користувачу в минулому для того, щоб знайти схожі предмети. Для порівняння, фільтрування на основі моделі переглядає не тільки поверхневі збіги. Натомість, воно використовує історію для тренування математичної моделі, щоб знаходити шаблони взаємодії між користувачами та продуктами. Це дозволяє системам передбачати, наскільки користувачеві сподобається новий продукт, вираховуючи глибокі взаємозв'язки в даних.

Гібридний метод використовує різні типи рекомендаційних алгоритмів задля уникнення обмежень індивідуальних рекомендаційних алгоритмів. На практиці, гібридний метод рекомендації може комбінувати різні техніки, базуючись на специфічних проблемах. Основна ідея цього методу полягає саме в ефективному додаванні різних рекомендаційних технік та регулюванні ваги між ними. Згідно з останніми опитуваннями, частота використання гібридних методів є відносно низькою, можливо через те, що приріст обчислювальної складності

виявляється не співмірним із перевагами, які дає поєднання цих методів, тому гібридні методи не стали оптимальним вибором серед методів рекомендацій.

Використовуючи алгоритми графіків для рекомендації, можна полегшити проблеми розрідженості даних і холодного старту в традиційних алгоритмах. Такі методи використовують зовнішню інформацію для вивчення схожостей між предметами, таким чином покращуючи точність рекомендацій результатів та різноманітність рекомендованого контенту. Методи рекомендацій на основі графіків переважно покладаються на три техніки: вбудовування графіків, видобування зразків шляхів та агрегація інформації високого порядку. Метод вбудовування графіків перетворює складні зв'язки між користувачами, рецептами та інгредієнтами у прості цифрові коди. Надалі метод агрегації інформації високого порядку збирає дані не лише про прямі вподобання користувача, але й вивчає характеристики всіх “сусідніх” об'єктів у мережі. Метод аналізу шляхів вивчає конкретні маршрути та логічні зв'язки між різними вузлами в базі даних продуктів. Він дозволяє одночасно враховувати смакові переваги та медичні вимоги до дієти, поєднуючи інформацію з різних графів для створення найбільш збалансованих рекомендацій.

Підходи до здорового харчування можуть бути поділенні на дві категорії: рекомендації рецептів на основі потреб здоров'я ті, що базуються на загальних медичних знаннях.

Рекомендації рецептів на основі потреб здоров'я повинні розглядати такі фактори, як здоров'я користувача та його потреби в поживних речовинах. Наприклад, у роботі Яо У та співавторів [13] описано алгоритм, що описує систему, яка рекомендує рецепти шляхом автоматичного вилучення інформації про поживні речовини, щоб допомогти користувачам досягти збалансованого харчування. Пропонується використовувати систему рекомендацій щодо харчування, яка враховує уподобання користувачів та медичні призначення, надаючи, таким чином, персоналізовані здорові рецепти. Рекомендації щодо рецептів, засновані на знаннях про здоров'я, впроваджують зовнішні знання, такі як правила здоров'я та поради експертів у систему рекомендацій [13].

На ринку існують різноманітні застосунки, які використовуються для підбору рецептів. В таблиці 1.2 проведено порівняння трьох основних лідерів.

Табл. 1.2

Порівняння наявних сервісів

Критерій порівняння	SuperCook	Yummly	Tasty
Основна концепція	Агрегатор рецептів за наявними інгредієнтами	Персоналізована система рекомендацій на основі ШІ	Платформа візуального контенту та відео-рецептів
Джерело рецептів	Зовнішні кулінарні сайти	Власна база + інтегровані партнери	Власний ексклюзивний контент
Система фільтрації	Пошук за інгредієнтами	Глибока: дієти, алергії, час, рівень складності, техніка	Базова: типи страв, інгредієнти, дієтичні обмеження
Тип контенту	Текст та посилання на оригінальні ресурси	Текст, професійні фото, відео-інструкції	Короткі динамічні відео
Персоналізація	Мінімальна	Висока	Середня
Інтеграція	Веб-платформа та мобільний додаток	"Розумна" кухня, планування покупок, HealthKit	Соціальні мережі, мобільний додаток, електронна комерція
Монетизація	Безкоштовно	Freemium	Рекламна модель та спонсорські рецепти

Доцільно аналізувати вже наявні сервіси підбору рецептів. Одним з таких прикладів є застосунок SuperCook, на зображенні 1.6 показаний його інтерфейс. Цей сервіс реалізує модель посередника, його основною метою є ефективне використання наявних надлишкових інгредієнтів, оскільки система дозволяє знаходити рецепти на основі вже доступних продуктів. При аналізі сервісу було виведено те, що платформа сама не створює власний кулінарний контент, а виступає посередником між користувачем та списком зовнішніх кулінарних сайтів. Даний підхід забезпечує доступ до значного обсягу рецептів, але водночас призводить до певної нестабільної якості інформації [14].

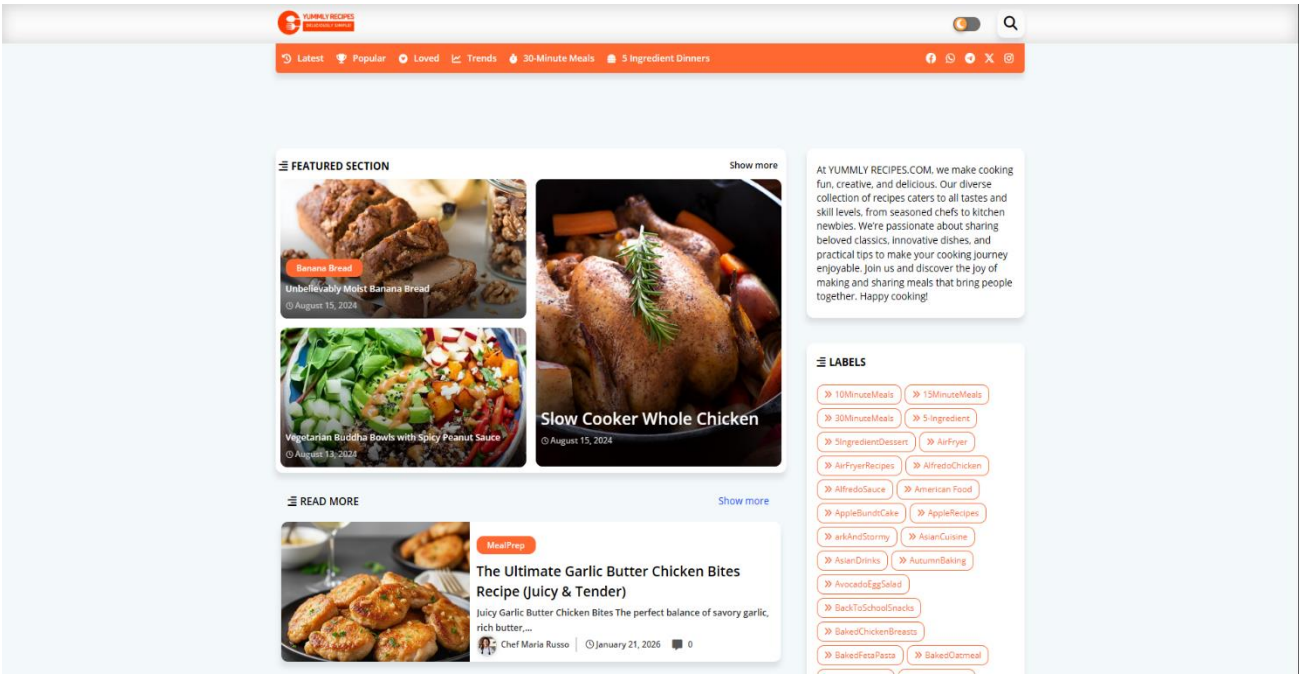

Рис. 1.6 Інтерфейс сайту SuperCook [14]

Інший підхід реалізований у сервісі Yummly. Він характеризується більш високим рівнем технологічної складності та персоналізації. Рис. 1.7 показує інтерфейс сервісу. На відміну від SuperCook, дана система активно використовує алгоритм машинного навчання для аналізу кулінарних уподобань користувача та інших даних користувача для створення особливого профілю для потреб кожного. Фактично сервіс адаптується до потреб користувача та виступає як персональний кулінарний асистент [15].

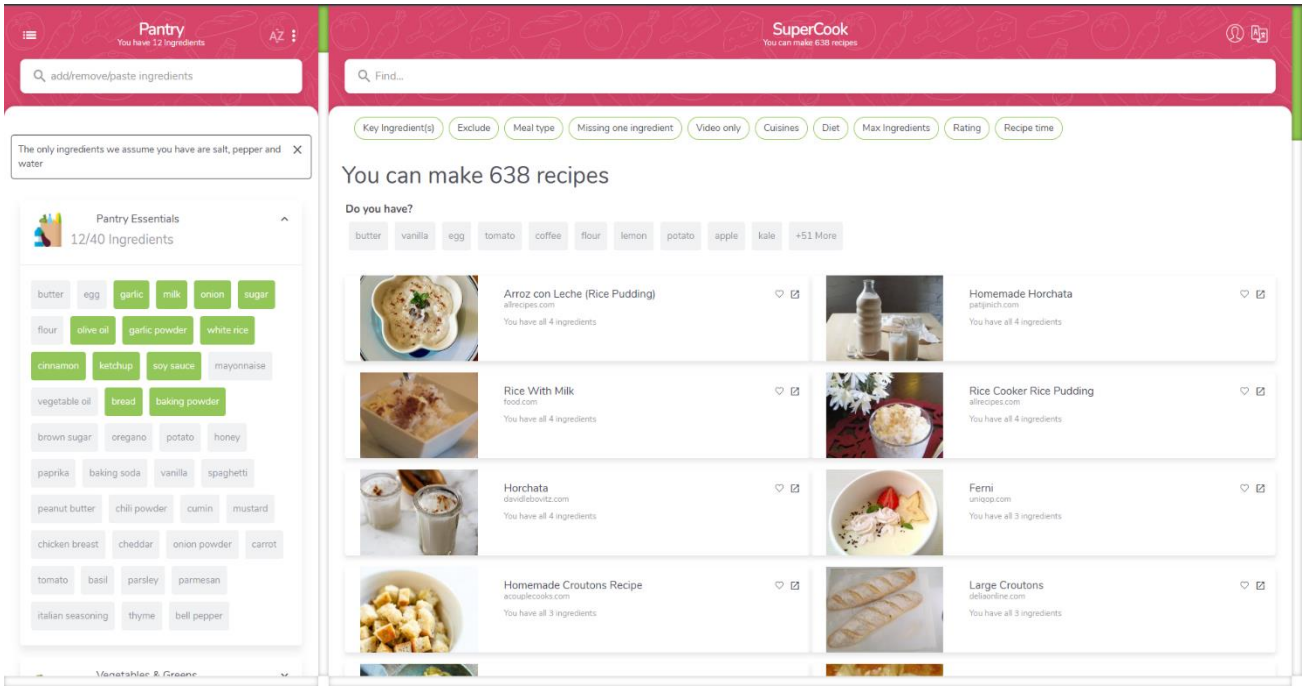

Рис. 1.7 Інтерфейс сайту Yummly [15]

В окремій категорії виступають сервіси, які роблять акцент на мультимедійній подачі кулінарного контенту. Прикладом цього є Tasty. Цей сервіс приділяє увагу візуальному представленню рецептів та зручності користувацького інтерфейсу, показаний на рис. 1.8. Основна частина сервісу - це короткі відеоінструкції, що демонструють процес приготування страв. Такий формат дозволяє спростити процес приготування страв, що робить платформу привабливою для користувачів [16].

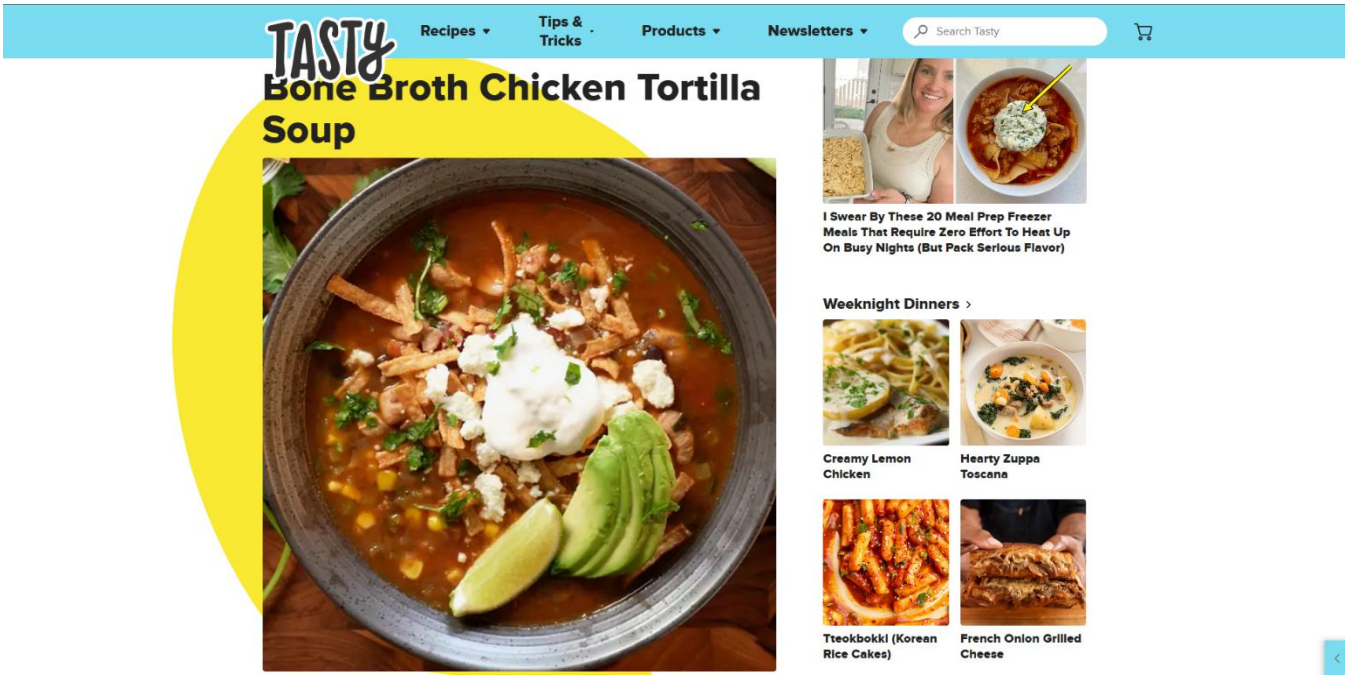

Рис. 1.8 Інтерфейс сайту Tasty [16]

Таким чином, аналіз існуючих сервісів підбору рецептів демонструє різноманітність підходів до організації кулінарних рекомендацій – від агрегаторів рецептів до персоналізованих систем та мультимедійних платформ. Попри широкий функціонал, більшість із них потребує ручного введення інгредієнтів та не використовує повною мірою можливості сучасних мультимодальних моделей для автоматичного аналізу продуктів. У зв’язку з цим, актуальним є дослідження та розробка веб-застосунків, які поєднують алгоритми комп’ютерного зору та великі мовні моделі для розпізнавання інгредієнтів і формування рекомендацій щодо приготування страв. Саме такий підхід реалізовано у, запропонованому в межах даної роботи, застосунку.

1.4 Постановка проблематики та завдань підвищення точності ідентифікації продуктів

Під час аналізу практичних рішень у сфері підбору рецептів було встановлено, що зростання використання штучного інтелекту в повсякденному житті та веб-застосунках. пов’язане з підвищенням вимог до точності, надійності та персоналізації [1]. Зі збільшенням кількості таких систем актуальність проблеми врахування індивідуальних харчових обмежень стає все більш актуальною. Алергії, поширеність яких зростає з кожним наступним роком, створюють додаткові ризики для здоров'я користувачів [2]. Це обумовлює необхідність підвищення якості обробки даних та коректності формування рекомендацій.

Під час проведення аналізу побудови мультимодальних моделей було виявлено, що сучасні візуально-мовні моделі мають можливість обробки текстових та візуальних даних. Однак, їх застосування супроводжується проблемами та обмеженнями [6]. Одна з ключових проблем є галюцинації, коли модель генерує недостовірну або некоректну інформацію, базуючись на упередженості. Це може призводити до помилок під час аналізу фотографій для визначення інгредієнтів та формування рекомендацій [9], [11]. Якість роботи залежить, значною мірою, від обсягу даних та їх якості, які були використані при тренуванні моделі. Це ускладнює їх адаптацію до реальних умов, при яких вони можуть зустрітися з невідомими продуктами [5].

Аналіз існуючих сервісів показав, більшість з них використовують або обмежені підходи до персоналізації, або покладаються на ручне введення інгредієнтів. Таким чином, ефективність та бажання користуватися таким застосунком значним чином знижується, бо підвищується шанс помилок [13]. У лідерів цієї індустрії відсутня повноцінна інтеграція мультимодальних алгоритмів для автоматичного розпізнавання інгредієнтів. Це обмежує ефективність схожих рішень та не дозволяє реалізувати весь потенціал сучасних моделей штучного інтелекту [14]–[16].

Узагальнення результатів аналізу показало, що існує наявність комплексу взаємопов’язаних проблем: недостатня точність при ідентифікації продуктів за зображеннями, схильність моделей до надання некоректних результатів, залежність від якості навчальних даних та обмеження функціональних можливостей існуючих систем підбору рецептів.

У зв'язку з цим, доцільно розробити інформаційну систему для підбору рецептів на основі мультимодальних алгоритмів інтелектуального аналізу даних. Така система має гарантувати автоматичну ідентифікацію продуктів за фотографіями та зменшення впливу галюцинацій моделей шляхом інтеграції механізмів перевірки достовірності результатів та формування доречних рекомендацій. Ця система повинна бути адаптована до різних сценаріїв використання та можливості інтеграції у веб-середовище з урахуванням вимог до масштабованості та доступності.

Відповідно до визначеної проблематики, у межах даної роботи поставлено такі основні завдання:

1. Розробити механізм мультимодальної обробки даних для автоматичної ідентифікації харчових продуктів на зображеннях, що дозволить мінімізувати необхідність ручного введення інгредієнтів користувачем.
2. Реалізувати алгоритм дворівневої верифікації результатів, спрямований на зменшення впливу галюцинацій моделей та підвищення точності формування фінального списку інгредієнтів.
3. Спроектувати та впровадити інтелектуальну систему підбору рецептів, яка на основі верифікованих даних забезпечує формування персоналізованих рекомендацій у середовищі веб-застосунку.

РОЗДІЛ 2. АНАЛІЗ ТА ПІДБІР ТЕХНОЛОГІЧНИХ РІШЕНЬ ТА ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Вибір мультимодальної моделі Gemini та стеку розробки

Для обробки фотографій, надісланих користувачем було використано штучний інтелект Gemini, розроблений компанією Google. Переглянувши інші моделі та їх можливості, була обрана саме ця модель для використання її прогресивних мультимодальних можливостей.

Модель була оцінена командою Gemini Team Google на чотирьох різних можливостях. Перший напрям охоплює розпізнавання об’єктів високого рівня за допомогою завдань з підписанням або у форматі «запитання-відповідь», таких як VQAv2 (Visual Question Answering v2) – відомий тест, де модель відповідає на відкриті питання до реальних фотографій, версія v2 збалансована так, щоб модель не мала змоги вгадати відповідь за статистикою тексту, не дивлячись на фото.

Увагу також було приділено детальній транскрипції та розпізнаванню дрібних елементів за допомогою завдань, таких як TextVQA та DocVQA – перший тест оцінює здатність зчитувати текст з фотографій реального світу, а інший - це тест на розуміння інфографіків з даними, що подані у складному візуальному форматі з текстом, іконками та різним просторовим розміщенням. Ці тести вимагають від моделі розпізнавання дрібних деталей. Також оцінено здатність розуміння діаграм, що вимагає просторового розуміння компонування вхідних даних за допомогою завдань ChartQA та InfographicVQA, обидва тести були розроблені для оцінки розуміння графіків та просторового мислення.

Найскладнішим етапом тестування стала перевірка мультимодального міркування за допомогою завдань, таких як Ai2D (AI2 Diagrams) – це набір даних, який складається з наукових діаграм, модель має розуміти зв’язки між елементами. MathVista – це тест на розв'язання математичних задач, які потребують візуального контексту. MMMU (Massive Multi-discipline Multimodal

Understanding) – це тест який містить 30+ запитань від вищої математики до мистецтвознавства; для відповіді потрібні знання рівня університету.

Процес тестування відбувався за суворою методологією, що базується на принципах Zero-shot prompting – це опрацювання, де модель штучного інтелекту бачить завдання вперше без попередніх тестувань на схожих запитань, модель отримує вказівку надавати короткі відповіді, що відповідають конкретному еталону. Усі числа отримано за допомогою “жадібного вибіркового” (Greedy sampling): методи генерації тексту, коли модель завжди обирає найбільш імовірне наступне слово, що в свою чергу робить результати тестів відтвореними. Важливою технічною особливістю оцінювання Gemini стало повне ігнорування зовнішніх інструментів OCR. Розпізнавання тексту здійснювалося нативно самою архітектурою моделі, що підтверджує її цілісну мультимодальну природу [17].

	Gemini Ultra (pixel only)	Gemini Pro (pixel only)	Gemini Nano 2 (pixel only)	Gemini Nano 1 (pixel only)	GPT-4V	Prior SOTA
MMMU (val) Multi-discipline college-level problems (Yue et al., 2023)	**59.4%** pass@1 **62.4%** Maj1@32	47.9%	32.6%	26.3%	56.8%	56.8% GPT-4V, 0-shot
TextVQA (val) Text reading on natural images (Singh et al., 2019)	**82.3%**	74.6%	65.9%	62.5%	78.0%	79.5% Google PaLI-3, fine-tuned
DocVQA (test) Document understanding (Mathew et al., 2021)	**90.9%**	88.1%	74.3%	72.2%	88.4% (pixel only)	88.4% GPT-4V, 0-shot
ChartQA (test) Chart understanding (Masry et al., 2022)	**80.8%**	74.1%	51.9%	53.6%	78.5% (4-shot CoT)	79.3% Google DePlot, 1-shot PoT (Liu et al., 2023)
InfographicVQA (test) Infographic understanding (Mathew et al., 2022)	**80.3%**	75.2%	54.5%	51.1%	75.1% (pixel only)	75.1% GPT-4V, 0-shot
MathVista (testmini) Mathematical reasoning (Lu et al., 2023)	**53.0%**	45.2%	30.6%	27.3%	49.9%	49.9% GPT-4V, 0-shot
AI2D (test) Science diagrams (Kembhavi et al., 2016)	**79.5%**	73.9%	51.0%	37.9%	78.2%	81.4% Google PaLI-X, fine-tuned
VQAv2 (test-dev) Natural image understanding (Goyal et al., 2017)	**77.8%**	71.2%	67.5%	62.7%	77.2%	86.1% Google PaLI-X, fine-tuned

Рис. 2.1 Порівняння результатів тестування моделей Gemini на мультимодальних бенчмарках [17]

Для розуміння історії мультимодальних принципів та причини використання Gemini доцільно проаналізувати результати її першої флагманської версії – Ultra 1.0. Хоча на момент 2026 року більшість індустрії вже перейшла до

використання наступного покоління, саме результати версії 1.0 заклали стандарт нативного мультимодального тестування без використання OCR.

На рис. 2.1 можливо побачити порівняння моделей з відомою моделью GPT-4V, яка була розроблена компанією OpenAI [17]. На момент виходу Gemini Ultra встановила нові SOTA-показники у 30 з 32 основних галузевих тестів. Зокрема, вона вперше подолала перешкоду у 90% у тестів MMLU та показала високу ефективність у складних візуальних міркуваннях. Використання цієї моделі як базису для порівняння дозволяє простежити динаміку зростання когнітивних здатностей ШІ за останні кілька років [17].

На даний момент в проєкті використовується модель Gemini 2.5 Flash з використанням API Gemini для швидкого підключення та надійного з'єднання. Модель Gemini 2.5 Flash була обрана завдяки її нативній мультимодальній архітектурі без використання OCR, Gemini обробляє візуальні та текстові дані в єдиному нейронному просторі, що забезпечує вищу точність при ідентифікації складних об'єктів. Критично важливою перевагою моделі є здатність приймати та обробляти декілька зображень одночасно. Для цього проєкту це дозволяє користувачу завантажувати фото різних полиць холодильника або окремих груп продуктів, а моделі це дозволяє точніше сформувати перелік наявних інгредієнтів. У порівнянні з конкурентними рішеннями Gemini демонструє значно нижчу вартість використання за мільйон токенів при збереженні високої якості генерації, що робить систему масштабованою та придатною для проектування MVP (Minimum Viable Product). Важливою технічною особливістю є підтримка керованої генерації за допомогою JSON-схем. Це дозволяє жорстко зафіксувати формати відповіді моделі, при впевненості, що модель завжди повертатиме валідний JSON-об'єкт. ака інтеграція з TypeScript-кодом додатку повністю усуває помилки парсингу та необхідність складної пост-обробки неструктурованого тексту [18].

Для розробки проєкту використовувались всі технології показані в таблиці 2.1.

Табл. 2.1

Вибір технологічного стеку розробки

Технологія	Призначення
TypeScript	Забезпечення типізації даних та підвищення стійкості системи до помилок
React	побудова інтерфейсу користувача
Next.js	фреймворк для розробки веб-застосунку
Tailwind CSS	стилізація UI
ShadCN	бібліотека компонентів
Spoonacular API	отримання списку рецептів
Gemini API	аналіз фотографія для виявлення інгредієнтів

Вибір TypeScript як основної мови для розробки застосунку зумовлений необхідністю створення надійного та стабільного програмного коду. На відміну від JavaScript, TypeScript впроваджує строгу типізацію, що надає наступні переваги. По-перше, запобігання помилкам на етапі розробки. Більшість помилок виявляються компілятором до запуску коду. Це важливо при роботі з Gemini API, де структура відповіді є складною. Завдяки системі типів код стає самодокументованим. Таким чином, це полегшує процес рефакторингу та масштабування системи, оскільки розробник завжди бачить очікувану структура даних. TypeScript є стандартом галузі для розробки на Next.js та React. Це забезпечує гарантовану сумісність з сучасними бібліотеками та інструментами розробки, що в свою чергу підвищує продуктивність та якість фінального продукту [19].

Вибір бібліотеки React як основи розробки інтерфейсу обумовлений її інтеграцією з фреймфорком Next.js, що створює ефективне середовище розробки сучасних веб-додатків. Використання React дозволило застосувати компонентний підхід, завдяки якому можливо розділяти інтерфейс на окремі компоненти, що значно спрощує підтримку та можливості розширення. Також глибоке розуміння принципів роботи з станом та “хуками” дозволило реалізувати динамічну взаємодію з користувачем [20].

Для розробки клієнтської частини було використано фреймворк Next.js – це фреймворк на базі React для створення повнофункціональних веб-додатків. При розробці використовуються компоненти React для створення користувацьких інтерфейсів, а Next.js – для реалізації додаткових функцій та оптимізації. Він також автоматично налаштовує інструменти нижнього рівня, такі як пакетизатори та компілятори. Таким чином, можливо зосередитися на розробці продукту та його швидкого випуску. Цей фреймворк один з найпопулярніших на даний момент в світі для написання веб-застосунків з React. На ньому створено такі застосунки як: TikTok, Hulu, DoorDash, United Airlines, HBO Max, Nike, IGN та багато інших. Він був обраний для цього проєкту через сукупність технічних та стратегічних переваг. По-перше, архітектура Next.js забезпечує високу швидкість рендерингу та SEO-оптимізацію. По-друге, використання даного фреймворку дозволило спиратися на існуючий практичний досвід розробника, що суттєво скоротило час на проектування архітектури [21].

Для стилізації застосунку був використаний Tailwind CSS фреймворк, який орієнтований на функціональність та швидкість. З використанням вже готових утилітарних класів, які дозволяють стилізувати блоки в HTML/TSX коді, що прискорює процес розробки інтерфейсу. Tailwind пропонує чітку систему відступів, кольорів та шрифтів які допомагають уникнути неочікуваних помилок у CSS та забезпечує стабільний візуальний стиль додатку. Завдяки можливостям фреймворку у фінальну збірку потрапляють лише ті стилі, які використовуються [22].

Вибір бібліотеки shadcn/ui, яка базується на Radix UI та Tailwind була використана через її високорівневих компонентів інтерфейсу для пришвидшення розробки інтерфейсу завдяки її готовим складним компонентам, які відразу мають професійний вигляд. На відміну від стандартний бібліотек shadcn не встановлюється як залежність, а копіюються в проєкт, що дозволяє гнучко змінювати внутрішню логіку та дизайн компонентів під конкретні потреби [23].

Для підбору рецептів був використаний Spoonacular API. Оскільки він має доступ до тисяч верифікованих рецептів з детальним описом інгредієнтів та кроків приготування. API дозволяє пошук рецептів за набором інгредієнтів, що ідеально доповнює функціонал розпізнавання продуктів з фото з використанням Gemini. Додатковим фактором вибору Spoonacular API стала його доступність до тестування прототипів та їх розробки. Сервіс пропонує гнучку модель безоплатного використання, яка надає достатній ліміт щоденних запитів для проведення повного циклу розробки та валідації системи. Це дозволяє реалізувати повноцінний функціонал пошуку рецептів та аналізу нутрієнтів без додаткових фінансових витрат на етапі створення MVP (Minimum Viable Product) [24].

2.2 Розробка математичної моделі узгодженого прийняття рішень на основі динамічної оцінки довіри

У межах розробленої інформаційної системи підбору рецептів процес ідентифікації інгредієнтів та формування рекомендацій для користувача можливо описати у вигляді математичної моделі узгодженого прийняття рішень на основі динамічної оцінки довіри користувачем. Ця модель надає можливість враховувати невизначеність результатів мультимодального аналізу та забезпечує підвищення точності розпізнавання продуктів на зображеннях методом інтеграції механізму додаткової верифікації. Загальну структуру наведено на рис. 2.2, де відображено основні етапи обробки даних та прийняття рішення.

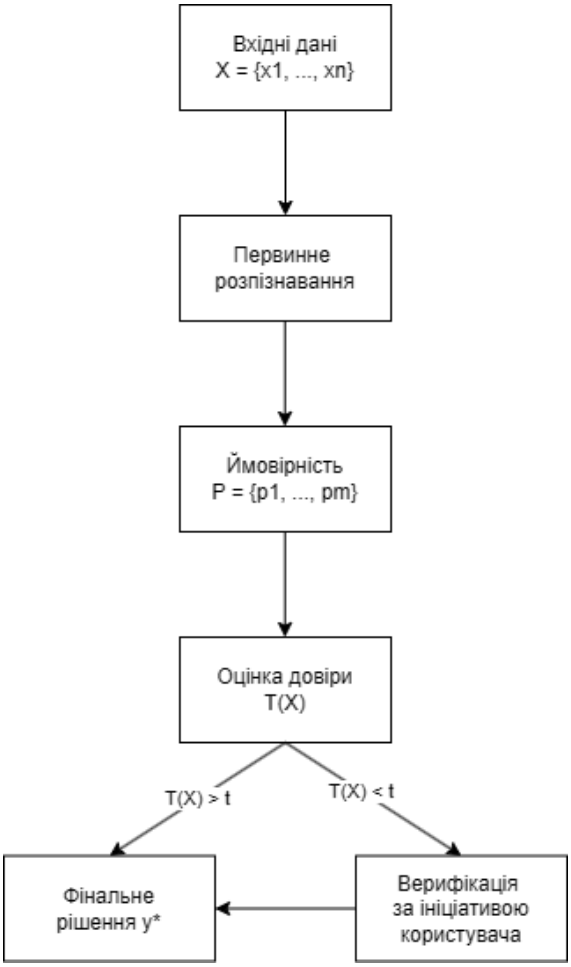

Рис. 2.2 Структурна схема адаптивної математичної моделі

Вхідні дані системи подаються у вигляді множини ознак

$$X = \{x_1, x_2, \dots, x_n\}, \#(2.1)$$

де X – це вектор ознак, вилучених із завантаженої фотографії. Кожен елемент x_i відповідає певним характеристикам об’єкта. Наприклад, піксельним даним, колірним гістограмам або текстурі окремих об'єктів. На першому етапі система не розуміє, що точно зображено. Але вже готує структурований масив даних для нейронної мережі.

Далі при первинному розпізнаванні вхідний вектор обробляється мультимодальною моделлю, яка зіставляє зрозумілі ознаки з відомими класами продуктів. При аналізі алгоритм може брати до уваги контур об'єкта, етикетку та

її форму, намагаючись ідентифікувати кожен інгредієнт у кадрі. В процесі виконується семантична сегментація, під час якої модель відокремлює різні продукти на основі контексту та візуальної схожості. Результатом є набір сирих даних, для розуміння яких потрібно провести статичну інтерпретацію. Цей блок можна вважати головним, оскільки саме в ньому пікселі на фотографіях перетворюються на зрозумілі для комп'ютера категорії.

На основі множини модель формує розподіл ймовірностей для кожного об'єкта:

$$P = \{p_1, p_2, \dots, p_m\} \#(2.2)$$

Для кожного розпізнаного продукта модель генерує значення pj, яке показує ступінь впевненості алгоритму належності об’єкта до відповідного класу. Сума ймовірностей для одного об'єкта зазвичай дорівнює одиниці, що тим самим дозволяє оцінити ступінь впевненості моделі зі схожими позиціями. Вектор P є критичним, оскільки саме завдяки йому система може оцінити власну точність ще до того, як надасть результат користувачу. Вхідними даними для першого етапу класифікації виступає вектор характеристик X, мультимодальна модель виконує класифікацію об’єктів, формуючи ймовірнісний розподіл їх належності до заданої множини класів інгредієнтів. Математично цей процес описується за допомогою апостеріорної ймовірності:

$$p_j = P(y = j \mid X), \qquad j = 1, \dots, m, \#(2.3)$$

де: y – цільова змінна, що позначає клас інгредієнта, X – вектор вхідних даних, p_j – ступінь впевненості моделі у тому, що об’єкт належить саме до класу j, m – загальна кількість категорій продуктів, доступних у базі даних системи.

Отримані значення p_j стають основою для розрахунку інтегральної оцінки довіри $T(X)$. У класичних підходах рішення приймається за правилом максимальної ймовірності

$$\hat{y} = arg\ max_j p_j \#(2.4)$$

Правило передбачає вибір класу з найбільшим значенням ймовірності. Такий підхід не враховує рівень невизначеності результату, що може призвести до помилок у випадках, коли ймовірності є нечіткими або неоднозначними.

З метою подолання зазначеного вище обмеження у запропонованій моделі використовується функція довіри $T(X)$:

$$T(X) = \frac{1}{N} \Sigma_{i=1}^{N} c_i, \#(2.5)$$

де c_i – рівень впевненості для i-го інгредієнта, а N – кількість виявлених об’єктів. Вона аналізує, чи достатній рівень впевненості моделі для автоматичного прийняття рішення без ризику помилки. Якщо це значення переважає поріг, то система вважає результат достатньо надійним і переходить до фіналу. У випадку якщо виникає неоднозначність, наприклад, якщо модель не може класифікувати два схожі продукти, система ініціює верифікацію. Такий підхід дозволяє уникнути некоректних рекомендацій умовлених хибним розпізнаванням.

Також функція довіри може бути подана через нормалізовану ентропію розподілу:

$$T(X) = 1 - \frac{-\Sigma_{j=1}^{m} p_j log p_j}{log m}, \#(2.6)$$

що дозволяє врахувати ступінь невизначеності результату розпізнавання. У такому випадку низькі значення функції відповідають високому рівню

невизначеності. Ключовим елементом моделі є введення порового значення довіри:

$$T(X) < \tau, \#(2.7)$$

яке визначає необхідність додаткової перевірки результату. Зазвичай перевірка виконується автоматично в класичних системах, у запропонованій моделі механізм верифікації активується за ініціативою користувача. Це забезпечує високу точність фінального результату, оскільки система може перевірити сама себе. Також такий підхід підвищує рівень довіри до системи, оскільки користувач відчуває контроль над процесом.

Сам процес активації верифікації можна описати змінною

$$v = \begin{cases} 1, \text{якщо користувач ініціює перевірку,} \\ 0, \text{інакше.} \end{cases} \#(2.8)$$

У випадку $v = 1$ система переходить до етапу повторного аналізу, під час якого формується більш точний результат. Загальна структура узгодження рішеньміж етапами наведена на рис. 2.3.

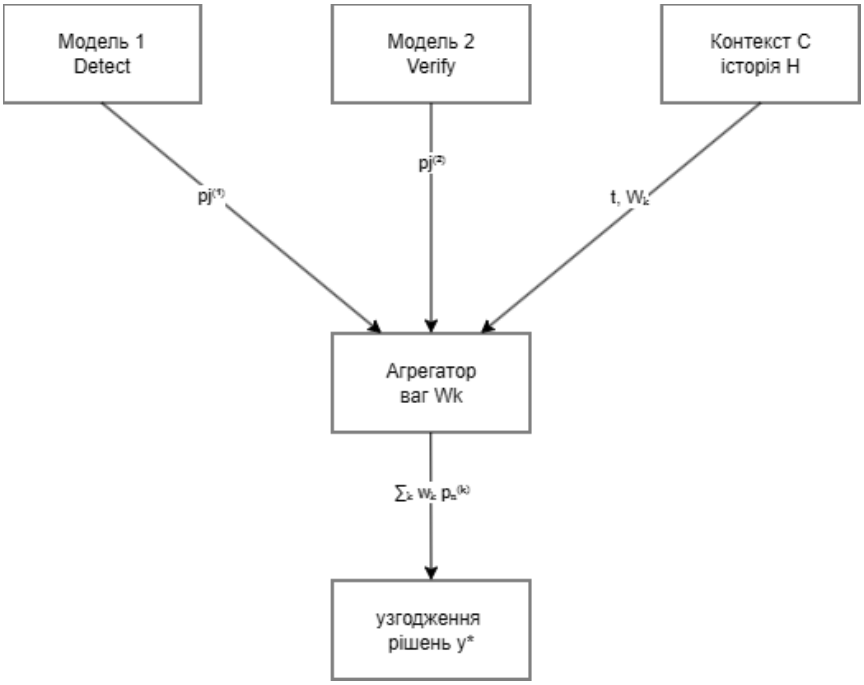

Рис. 2.3 Графік узгодження рішень між підмоделями

“Модель 1: Detect” і “Модель 2: Verify” представляють різні ітерації аналізу зображення. Detect – це первинний автоматичний прохід моделі, результатом якого буде вектор ймовірностей $p_j^{(1)}$. Функція Verify активується у випадку низької впевненості та генерує другий набір даних $p_j^{(2)}$. Завдяки розділу на дві частини система має два незалежні погляди на складний об'єкт, що підвищує стійкість до помилок.

Компонент “Контекст C та історія H” забезпечує інтеграцію зовнішніх параметрів, які впливають на точність верифікації. Контекст C може включати зовнішні умови, тоді як історія H – це попередній досвід моделі при взаємодії з користувачем. На основі цих даних обчислюється порогове значення t та визначаються вагові коефіцієнти w_k, які відображають ступінь довіри системи до автоматизованого розпізнавання порівняно з експертною верифікацією. Функція забезпечує адаптивність системи, запобігаючи її залежності від виключно статичних алгоритмічних підходів.

Агрегатор ваг виконує ключову роль, в ньому відбувається синтез накопичених даних. Він оперує ваговими коефіцієнтами $w1$ та $w2$, де сума ваг

зазвичай нормалізується на одиниці. Він збалансовує внесок кожного етапу обробки, щоб мінімізувати ймовірність некоректного розпізнавання. При здійсненні верифікації користувачем агрегатор надає пріоритет уточненим даним. Це дозволяє математично обґрунтувати ефективність залучення інтелектуального ресурсу людини в ситуаціях високої невизначеності.

Перед останнім блоком виконується математична операція - зважена сума:

$$\Sigma\, w_k p_j^{(k)} \#(2.9)$$

Вона є двигуном агрегатора, при якій кожна ймовірність j з усіх етапів множиться на відповідні ваги та додається. У результаті формується єдиний вектор, який враховує всі доступні інформативні ознаки щодо типу продукту. Цей підхід дозволяє згладити випадкові помилки різних етапів. При помилці однієї моделі друга виправить результат та сумарне значення вплине на формування максимально достовірного результату. Операція зваженого підсумовування перетворює різні здогадки на єдину, математично обґрунтовану оцінку.

Процес прийняття фінального рішення формалізується за допомогою виразу:

$$\hat{Y} = \{^{Y^{(1)},\ v=2}_{\Phi(Y^{(1)},Y^{(2)}),v=1}, \#(2.10)$$

де $Y^{(1)}$ – результат первинного розпізнавання, $Y^{(2)}$ – результат верифікації та Φ – функція узгодження. Узагальнено, цю формулу можливо представити у вигляді зваженої агрегації:

$$\widehat{y_j^*} = \arg max_j \sum_{k=1}^{K} w_k p_j^{(k)}, \#(2.11)$$

де w_k – вагові коефіцієнти довіри до результатів окремих етапів обробки, а $p_j^{(k)}$ – оцінка ймовірності для j класу на кожному етапі k. Виходячи з цього, фінальний компонент відповідає за вибір найбільш імовірного класу продукта за принципом $argmax$. Система знаходить інгредієнт, у якого підсумковий вектор набрав найбільшу зважену суму, і приймає його як істину. Рішення y^* є результатом дії між штучним інтелектом та користувачем. На цьому етапі невизначеність вважається усуненою, а отриманий список продуктів передається до логіки підбору рецептів.

Таким чином, запроваджена модель реалізує багаторівневу обробку даних, що включає первинний аналіз, оцінку довіри та механізм повторної верифікації.

Аналіз розробленої архітектури та математичного апарату дозволяє зробити висновок про ефективність гібридного підходу до розпізнавання інгредієнтів. Сформована модель забезпечує інтеграцію результатів мультимодального аналізу з урахуванням контекстних параметрів та історичних даних. Особливістю запропонованого рішення є формалізація процедури переходу від автоматизованої обробки до експертної верифікації. Замість статичного прийняття рішення система базується на динамічному обчисленні функції довіри та адаптивних порогових значень. Використання зазначеної методики створює умови для багаторівневого уточнення результатів, де залучення користувача виступає математично обґрунтованим етапом мінімізації невизначеності. Запропонована система дозволяє досягти вищої точності класифікації у складних сценаріях, зберігаючи високу швидкість обробки даних

2.3. Архітектура інтеграції ШІ-інтерфейсів у веб-систему

Насамперед, користувач взаємодіє з системою через фронтенд для завантаження фотографій для обробки. Для завантаження використовується модальне вікно – елемент інтерфейсу, який переводить пристрій у режим прийому даних, що були введені в діалогове вікно. Такий підхід до завантаження даних забезпечує зручність, покращує UX та мінімізує можливі помилки користувача.

Взаємодія між застосунком та штучним інтелектом виконується через API. API – це конкретний запит, що надсилається клієнтом на віддалений сервер, який в свою чергу містить спеціалізовану службу або набір даних. Ці взаємодії виконуються тільки за використання суворих протоколів, щоб гарантувати розуміння між сервером та клієнтом. При ініціюванні виклику, клієнт об'єднує в один пакет конкретну цифрову адресу та набір інструкцій. Ці інструкції містять метод, який повідомляє серверу, чого саме користувач бажає отримати: отримання сформованої відповіді або надіслати нові дані для обробки.

Основною частиною цього процесу є використання заголовків та дані. Заголовки виконують роль інформації та містять секретний ключ API для підтвердження особи, забезпечення безпеки та опис формату даних, на які користувач дав запит. Дані – це зміст повідомлення, яке майже завжди має формат JSON [25].

JSON (JavaScript Object Notation) – це компактний текстовий формат обміну даними. Він побудований на двох універсальних структурах даних. Перший це набір пар «ім'я-значення» та другий – впорядкований список значень. Ця простота є його найсильнішою стороною в порівнянні з старішими стандартами, такими як: XML або SOAP. Оскільки така структура побудови JSON файлів безпосередньо відповідає об'єктам, словникам та масивам, які зустрічаються майже в кожній сучасній мові програмування, то процес перекладу, який необхідний для переміщення даних з серверу до клієнта, відбувається майже миттєво. Формат JSON вважається стандартним вибором практично для всіх RESTful API та сучасних мікросервісів, адже він забезпечує баланс між зручністю для людей та

ефективністю для машин. Розробники мають можливість швидко виправляти помилки в файлі, машини, в свою чергу, аналізують його, використовуючи менше ресурсів процесора, ніж потрібно при більш складних документоорієнтованих форматах. Зменшуючи обчислювальні витрати на обмін даними, JSON забезпечує безперебійну інтерактивність у реальному часі [26].

Після отримання даних у вигляді пакета, сервер перевіряє ідентичність користувача та обробляє його запит. Після цього він надсилає відповідь з кодом статусу, щоб повідомити застосунок про успішність операції. Система API при використанні штучного інтелекту дозволяє простому веб-застосунку мати велику потужність таких моделей як Gemini, перетворюючи виклики API з простого інструменту для роботи з даними на важливе з'єднання, яке дозволяє програмному забезпеченню мислити та діяти в режимі реального часу.

Для того щоб система була масштабованою та функціональні обов'язки були чітко розподілені, ця розробка працює на основі трьохрівневої архітектури рис. 2.4.

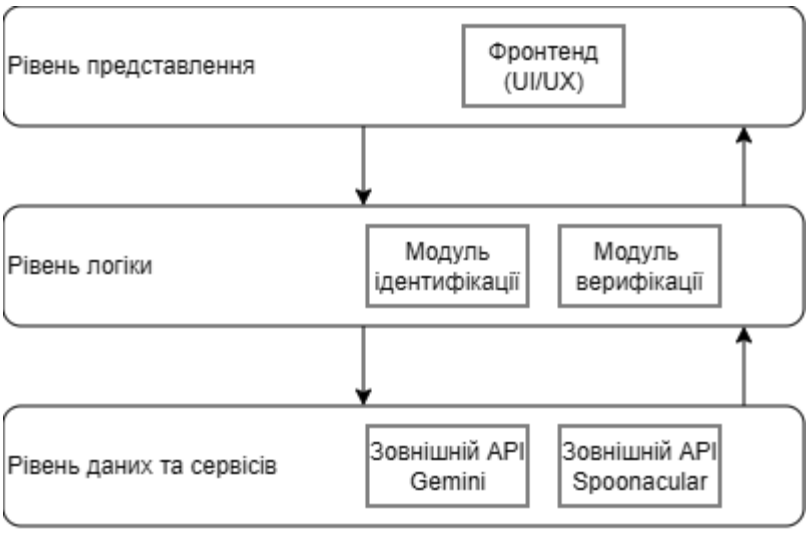

Рис. 2.4. Схема трьохрівневої архітектури інтелектуальної системи підбору рецептів

Дана архітектурна модель включає три рівні:

1. Рівень представлення (Presentation Tier) реалізовано як веб-інтерфейс, тобто фронтенд: через нього користувач взаємодіє із системою та бачить результати.

2. Рівень логіки (Logic Tier) – це основна частина системи, де працює авторський підхід до обробки даних. На цьому рівні присутній модуль ідентифікації, який спочатку розпізнає інгредієнти, а також модуль верифікації – по суті, “розумний аудитор”, що допомагає зменшити ймовірність галюцинацій моделі та вважається головним носієм наукової новизни рішення.

3. Рівень даних і сервісів (Data & Services Tier) відповідає за підключення до зовнішніх хмарних ШІ-сервісів та баз рецептів. Обмін даних відбувається через захищені API-шлюзи.

Для кращого розуміння інтеграції штучного інтелекту до цієї інформаційної системи, потрібно переглянути рис. 2.5, на якому зображено діаграму послідовності, яка візуалізує процес розпізнавання та верифікації інгредієнтів за допомогою мультимодальних моделей ШІ.

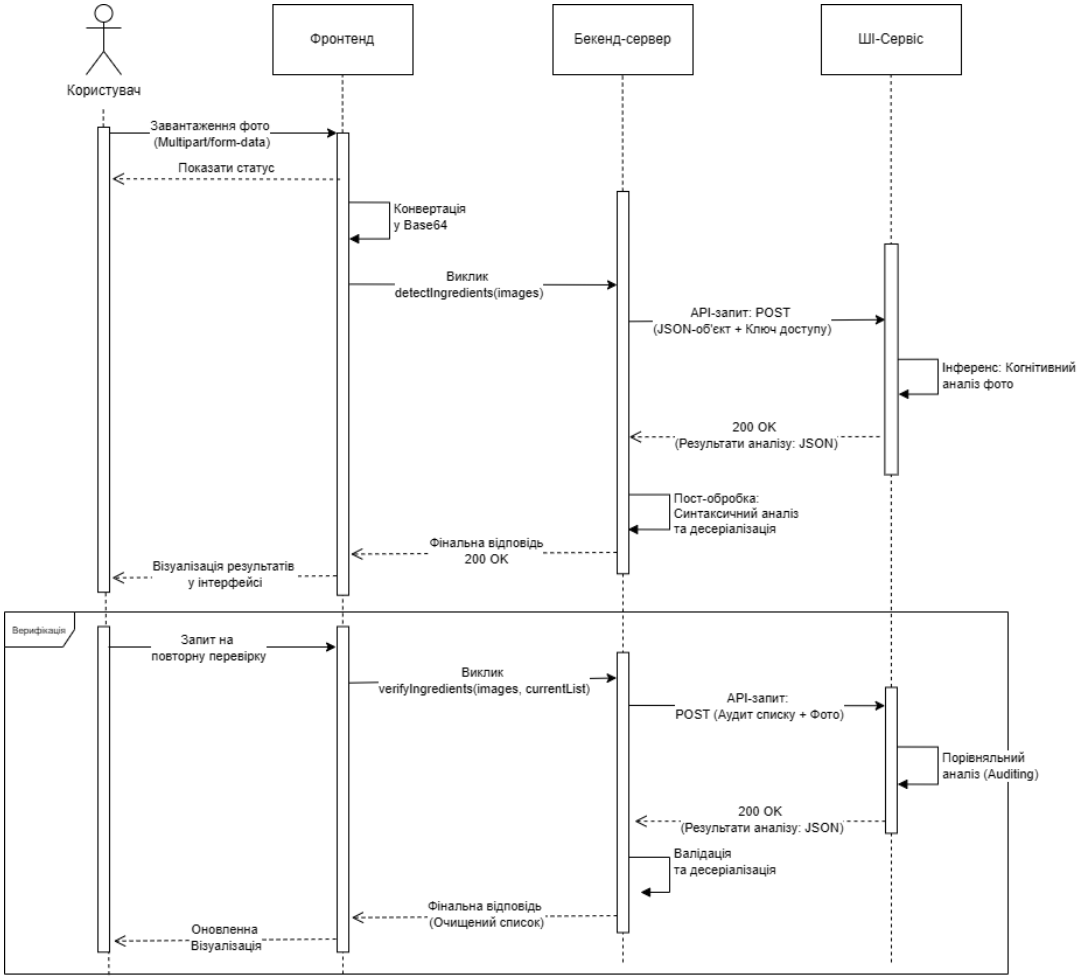

Рис. 2.5 Діаграма послідовності процесу розпізнавання та верифікації інгредієнтів за допомогою мультимодальних моделей ШІ

Дана діаграма базується на принципах асинхронної взаємодії між компонентами вебсистеми та зовнішнім сервісом. Процес починається на рівні фронтенд-інтерфейсу, де клієнт виконує попередню підготовку фотографій шляхом перетворення їх на формат base64 для подальшої передачі. Використання технологій серверних дій дозволяє інкапсулювати бізнес-логіку та забезпечити безпечне зберігання ідентифікаторів доступу на стороні бекенд-сервера. Під час формування запиту до API, система використовує декларативні схеми відповідей, що гарантує структурну цілісність відповіді у форматі JSON.

Ключовою частиною архітектури є впровадження циклу верифікації для мінімізації ймовірності виникнення галюцинацій. На цьому етапі модель змінює рольову установку на скептичного аудитора. Сервер передає сервісу не лише

вихідні зображення, а й сформований раніше список даних для аналізу. Такий підхід забезпечує дворівневу перевірку достовірності об’єктів. Після завершення аналізу та десеріалізації, список повертається до застосунку для фінальної візуалізації. Використання сучасних стандартів обміну даними та мультимодальних алгоритмів аналізу дозволяє досягти високої точності розпізнавання при збереженні низької затримки в роботі інтерфейсу.

У межах архітектури штучний інтелект виконує роль центрального інтелектуального компонента. Він буде забезпечувати обробку вхідних даних. На першому етапі здійснюється аналіз наданої інформації, отриманої від користувача з наміром ідентифікації об'єктів на зображенні. Результатом аналізу є сформований список інгредієнтів із відповідним рівнем впевненості.

Отримані дані передаються на наступний етап обробки, на якому використовується механізм верифікації результатів. На цьому етапі штучний інтелект повторно аналізує вхідне зображення та список інгредієнтів, наданий на попередньому етапі. Це дозволяє виявити неточності або помилки при первинному розпізнаванні. Цей підхід реалізує принцип подвійної перевірки та спрямований на зменшення впливу галюцинацій на відповіді.

Отже, штучний інтелект функціонує як інструментарій узгодження та прийняття рішень. Використання багаторівневої обробки даних дозволяє підвищити точність ідентифікації інгредієнтів, а також забезпечити стабільність роботи системи в умовах неповних або неоднозначних вхідних даних.

2.4. Проєктування алгоритмів обробки візуальних даних

Метою розробленого алгоритму є автоматизація обробки зображень для ідентифікації інгредієнтів та формування структурованого списку інгредієнтів для подальшого пошуку страв. Застосований підхід забезпечує високу зручність використання та спрощує взаємодію користувача з вебзастосунком.

Під час роботи алгоритм здійснює аналіз наданого користувачем зображення із використанням мультимодальної моделі штучного інтелекту,

здатної одночасно обробляти текстову та візуальну інформацію. Результатом аналізу є перелік інгредієнтів із відповідними оцінками впевненості. Загальну послідовність роботи алгоритму обробки візуальних даних наведено на рис. 2.5, де зображено основі етапи перетворення вхідного зображення у фінальний результат.

Описаний алгоритм визначає послідовність операцій в інтелектуальній системі розпізнавання об’єктів із застосуванням механізмів динамічної оцінки довіри. Ініціація процесу починається з функцією “початок”. Наступним етапом є завантаження зображення, що виступає в ролі даних для подальшого аналізу. Надалі фотографія перетворюється в формат Base64, що дозволяє передавати складні структури піксельних даних у вигляді стандартизованих текстових рядків для подальшої обробки. Після отримання даних, мультимодальною моделлю здійснюється вилучення візуальних ознак кожного об'єкта, які потім трансформуються у вектор ймовірностей, де кожен елемент відповідає певному класу інгредієнта, а його значення відображає впевненість моделі в ідентифікації конкретного продукту.

Нис. 2.5 блок-схема обробки візуальних даних

Важливим етапом схеми є обчислення рівня довіри, який порівнює максимальну отриману ймовірність із динамічно заданим порогом чутливості. Якщо зображення має високий рівень шуму або інгредієнти перекривають один

одного, модель може видати низький рівень впевненості, після якого користувач може ініціювати перехід до підсистеми верифікації. На наступному етапі, після проходження верифікації, процес завершується механізмом агрегації. Фінальний висновок формується шляхом синтезу двох прогнозів. Кожному джерелу присвоюється ваговий коефіцієнт, що дозволяє пріоритезувати вторинний аналіз фотографії над первинним у спірних даних.

Запропонована методика забезпечує формування верифікованого списку продуктів. У підсумку, алгоритм не просто автоматизує розпізнавання, а створює надійну систему зі зворотнім зв’язком, де точність підбору рецептів безпосередньо корелюється з мінімізацією невизначеності на етапі ідентифікації об’єктів.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ РОЗРОБЛЕНОГО АЛГОРИТМУ

3.1. Розробка програмного забезпечення та лістинг коду

Програмна реалізація інформаційної системи виконана у вигляді вебзастосунку, який забезпечує інтерактивну взаємодію користувача з алгоритмом обробки візуальних даних та їх повторне опрацювання. Основною метою реалізації є інтеграція мультимодальної моделі штучного інтелекту до середовища для автоматизованого розпізнавання інгредієнтів та створення списків для подальшого пошуку рецептів.

Архітектура системи базується на основі мережевої взаємодії між клієнтським інтерфейсом та серверною частиною. Фронтенд відповідає за взаємодію з користувачем, бекенд реалізує обробку даних та взаємодію із зовнішніми сервісами штучного інтелекту. Застосування клієнт-серверної архітектури сприяє підвищенню масштабованості системи, забезпечує чіткий розподіл функціональних обов'язків між її компонентами та дозволяє оптимізувати використання обчислювальних ресурсів.

У серверній частині застосунку реалізовано механізм формалізації відповіді моделі штучного інтелекту, що дозволяє привести результат розпізнавання до попередньо визначеної структури. Зазвичай відповіді, згенеровані моделями, мають неоднорідні вільні текстові формати. Завдяки цьому механізму, показаному на рис. 3.1, відповідь задається у форматі JSON-об'єкта для списку знайдених інгредієнтів.

```
const ingredientSchema: ResponseSchema = {
  type: SchemaType.OBJECT,
  properties: {
    ingredients: {
      type: SchemaType.ARRAY,
      items: {
        type: SchemaType.OBJECT,
        properties: {
          name: {
            type: SchemaType.STRING,
            description: "Lowercase name of the food item"
          },
          confidence: {
            type: SchemaType.NUMBER,
            description: "Certainty score between 0.0 and 1.0"
          }
        },
        required: ["name", "confidence"]
      }
    }
  },
  required: ["ingredients"],
};
```

Рис. 3.1 Механізм формалізації відповіді

Наведений фрагмент коду реалізує механізм стандартизації відповіді моделі, яку мультимодальна модель повинна повернути після завершення аналізу зображення. Відповідь визначається як об'єкт, що містить масив ingredients. Елементами цього масиву є окремі записи про знайдені продукти. Для кожного продукту передбачено два обов'язкові атрибути: name (містить назву інгредієнту) та confidence (характеризує рівень впевненості моделі у коректності розпізнавання). При використанні цієї схеми забезпечується не лише їх стандартизація, а й можливість подальшої обробки. Застосування формалізованої схеми відповіді підвищує програмну стійкість системи до неоднозначних або варіативних текстових відповідей моделі та надає коректну інтеграцію ШІ-компонента у загальну архітектуру вебзастосунку.

Надалі система базується на використанні механізму стандартизації реалізації. На рис. 3.2 наведено процес взаємодії із API мультимодальної моделі штучного інтелекту, який формує запит та отримує структуровану відповідь.

```
export async function detectIngredients(images: { base64: string; type: string }[]) {

  const parts = images.map(img => ({
    inlineData: {
      data: img.base64,
      mimeType: img.type,
    },
  }))

  try {
    const model = genAI.getGenerativeModel({
      model: "gemini-2.5-flash",
      systemInstruction: `You are a professional kitchen inventory assistant. Your task is to identify raw ingredients and food items from images. Be concise and only list items you are sure are present.`,
      generationConfig: {
          responseMimeType: "application/json",
          responseSchema: ingredientSchema
      },
    })
```

Рис. 3.2 Механізм першого запиту до ШІ

Функція detectingredients реалізує ключовий етап обробки зображення: первинне розпізнавання інгредієнтів на основі фото, наданих користувачем. Вхідні дані представленні у вигляді масиву із зображенням, попередньо перетворені у формат base64, а також містить інформацію про їх MIME-тип. Дане представлення дозволяє уніфікувати передачу даних через мережу та забезпечує сумісність із API штучного інтелекту.

На першому етапі виконання функції здійснюється трансформація вхідних даних, які відповідають вимогам Google Generative AI SDK. Для кожної фотографії в масиві формується об'єкт, що містить дані та тип файлу для коректної інтерпретації вхідних даних мультимодальною моделлю, яка повинна відновити зображення з текстового представлення та виконати його аналіз.

Наступним етапом є ініціалізація моделі штучного інтелекту із заданими параметрами. Для цього використовується модель gemini-2.5-flash. Додатково

подаються системні інструкції, які визначають поведінку моделі та обмежують її генерацію лише коректним результатом. Важливим аспектом також є налаштування формату відповіді, де вказується використання JSON формату та застосовується вже визначена схема для забезпечення стандартизації результату.

Після формування запиту виконується звернення до API. Вслід за отриманням результату він десеріалізується у JSON-об'єкт. Це дозволяє перетворити відповідь у структуровані формати для подальшої обробки та передачі до клієнтської частини застосунку. Окрему увагу приділено обробці помилок. Запит здійснюється у конструкції try…catch, що дозволяє перехоплювати виняткові ситуації, пов’язані з мережею, некоректною відповіддю або внутрішніми помилками сервісу. При виникненні таких ситуацій система інформує користувача про неможливість аналізу.

Подальший етап обробки базується на результатах первинного аналізу та використовує раніше отримані дані для уточнення результатів розпізнавання. Механізм повторної верифікації результатів представлено на рис. 3.3.

```
export async function verifyIngredients(images: { base64: string; type: string }[], currentList: string[]) {

  const parts = images.map(img => ({
    inlineData: {
      data: img.base64,
      mimeType: img.type,
    },
  }))

  try{
    const model = genAI.getGenerativeModel({
      model: "gemini-2.5-flash",
      systemInstruction: "You are a skeptical auditor. You verify the work of other AIs to ensure accuracy and prevent hallucinations.",
      generationConfig: { responseMimeType: "application/json", responseSchema: ingredientSchema },
    })
```

Рис. 3.3 Механізм повторної верифікації результатів

Механізм verifyIngredients за своєю структурою є подібним до первинної обробки даних. Ключовою особливістю даного етапу є використання додаткового

вхідного параметра currentList, який представляє собою масив інгредієнтів, отриманих на етапі первинного розпізнавання. Зазначений масив використовується як основа для повторного аналізу, що дозволяє моделі не лише ідентифікувати нові об'єкти, а перевіряє коректність вже існуючого списку. Механізм повторної верифікації результатів спрямований на усунення можливих помилок та зменшення впливу галюцинацій моделі. Основна відмінність між первинним та повторним етапами обробки полягає у формуванні промпту та системних інструкцій. Модель отримує роль “аудитора”, що перевіряє попередній результат, видаляє недостовірні елементи та доповнює список лише у випадку високої впевненості.

Після реалізації серверних механізмів перевірки та верифікації виникає необхідність його інтеграції з клієнтською частиною застосунку, адже користувач повинен мати можливість ініціювати процес аналізу через інтерфейс системи. Для цього у клієнтській частині реалізовано відповідну функцію handleUpload, що показана на рис. 3.4. Вона Забезпечує підготовку вхідних даних та запуск первинного етапу розпізнавання.

```
const handleUpload = async (e: React.ChangeEvent<HTMLInputElement>) => {
    const files = e.target.files
    if (!files) return
    if (files.length > 4) {
      toast.error("Maximum 4 photos allowed")
      return
    }
    setLoading(true)
    try {
      const newImages: { base64: string; type: string }[] = []
      for (const file of Array.from(files)) {
        const reader = new FileReader()

        const base64 = await new Promise<string>((resolve) => {
          reader.onloadend = () => {
            resolve((reader.result as string).split(",")[1])
          }
          reader.readAsDataURL(file)
        })
        newImages.push({
          base64,
          type: file.type,
        })
      }
      setImages(newImages)
      const data = await detectIngredients(newImages)
      setDetected(data.ingredients)
      const avg =
      data.ingredients.reduce(
        (sum: number, i: DetectedIngredient) => sum + i.confidence,
        0
      ) / Math.max(data.ingredients.length, 1)
      setAvgConfidence(avg)
    } catch {
      toast.error("Initial scan failed.")
    } finally {
      setLoading(false)
    }
}
```

Рис. 3.4 Функція підготовки даних та запуску первинної обробки даних

Функція handleUpload реалізує клієнтський етап попередньої обробки візуальних даних та передачі їх на сервер для аналізу. Функція отримує зображення від користувача, йде підготовка до передачі через API та ініціація процесу розпізнавання інгредієнтів. На першому етапі функція здійснює

перевірку вхідних даних для контролю їх кількості, щоб уникнути перевантаженість системи та забезпечити стабільність обробки запитів. У випадку перевищення кількості зображення користувач отримує відповідне повідомлення.

Далі виконується процес підготовки зображень. Зображення перетворюються у формат base64, для цього використовується механізм FileReader. Такий підхід забезпечує можливість передачі бінарних даних через HTTP-запити у стандартизованому форматі. Після перетворення кожного зображення, вони доповнюються інформацією про MIME-тип та зберігаються у масиві. Після цих дій реалізується передача підготовлених даних до модуля штучного інтелекту.

Після отримання відповіді від серверної частини відбувається оновлення стану застосунку для відображення списку розпізнаних інгредієнтів. Додатково обчислюється середній рівень впевненості шляхом агрегування значення confidence для кожного елемента. Цей показник використовується для оцінки достовірності результату та може бути застосований у подальших етапах прийняття рішень. Окрему увагу приділено обробці виняткових ситуацій. Використання конструкції try…catch дозволяє забезпечити коректну реакцію системи на помилки. У разі виникнення помилки користувач отримує повідомлення про невдалу спробу аналізу.

Функція handleUpload реалізує повний цикл підготовки та передачі візуальних даних, а також забезпечує інтеграцію клієнтського інтерфейсу з модулем інтелектуального аналізу.

Для реалізації повторного аналізу зображень механізму у клієнтській частині застосунку використовується функція handleDoubleCheck, зображена на рис. 3.5, яка забезпечує ініціацію та уточнення списку інгредієнтів.

```
const handleDoubleCheck = async () => {
    if (images.length === 0) return
    setIsVerifying(true)
    toast.info("Auditing detection...")
    try {
      const data = await verifyIngredients(images, detected.map(i => i.name))
      setDetected(data.ingredients)
      const avg =
      data.ingredients.reduce(
        (sum: number, i: DetectedIngredient) => sum + i.confidence, 0)
        / Math.max(data.ingredients.length, 1)
      setAvgConfidence(avg)
      toast.success("List refined!")
    } catch {
      toast.error("Verification failed.")
    } finally {
      setIsVerifying(false)
    }
}
```

Рис. 3.5 Механізм ініціації та уточнення списку інгредієнтів

Функція handleDoubleCheck реалізує механізм повторної верифікації результатів розпізнавання інгредієнтів за ініціативою користувача. Вона здійснює повторний виклик серверного модуля verifyIngredients, передаючи початкові зображення та сформований список інгредієнтів для уточнення. Після отримання відповіді відбувається оновлення списку продуктів та повторне обчислення середнього рівня впевненості, що дозволяє підвищити достовірність результатів. У функції також передбачено обробку помилок та індикацію стану виконання операції.

3.2. Оптимізація промпт-інжинірингу для розпізнавання інгредієнтів

Під час розробки системи було встановлено, що якість розпізнавання інгредієнтів значною мірою залежить від структури та змісту запиту до штучного інтелекту. З огляду на це, була проведена оптимізація промпт-інжинірингу для забезпечення точності та надійності роботи системи. На початкових етапах

використання моделі, були використані узагальнені запити, які містили мінімальний набір інструкцій. Приклад схожого запиту показаний на рис. 3.6.

```
const prompt = `Look at the fridge image carefully.Return a JSON object with an
'ingredients' array. Each ingredient must have a 'name' (lowercase) and a 'confidence'
score (0.0 to 1.0)`
```

Рис. 3.6 Базовий неструктурований запит до мультимодальної моделі

Під час експериментальної перевірки базового запиту проявилися помітні обмеження: зокрема генерації неточних або надлишкових результатів. Модель мала можливість додавати інгредієнти та мала більший шанс на галюцинації, що негативно впливає на достовірність результатів.

З метою усунення вищезазначених недоліків було проведено оптимізацію структури: введення чіткої послідовності аналізу зображення для поетапного огляду різних частин холодильника, зокрема верхньої, середньої та нижньої полиць, а також додаткових відсіків. Текст розробленого запиту представлено на рис. 3.7.

```
const prompt = `
      Look at the fridge image carefully.
      Scan the fridge systematically from:
      1. top shelf
      2. middle shelf
      3. bottom shelf
      4. door compartments
      5. drawers
      For each area:
      - identify clearly visible food or drink items
      - use common ingredient names
      - do NOT guess items that are not clearly visible
      After scanning all areas:
      - combine the items into one list
      - remove duplicates
      - format names in lowercase
    Then:
    Return a JSON object with an 'ingredients' array.
    Each ingredient must have a 'name' (lowercase) and a 'confidence' score (0.0 to 1.0).
    Confidence rules:
    - 1.0: Crystal clear, unmistakable.
    - 0.8: Highly likely, but slightly blurry.
    - 0.5: Obstructed view or generic packaging.
    - Below 0.5 = poor visibility`
```

Рис. 3.7 Оптимізований запит для первинного аналізу

Така декомпозиція складного візуального завдання базується на принципах “Chain-of-Thought” [27], де модель змушена проходити через послідовні логічні кроки перед формуванням фінального списку. Завдяки цьому підходу модель більш системно аналізує зображення та зменшує ймовірність пропуску об'єктів. Додатково було введено обмеження щодо поведінки моделі, які забороняють генерацію припущень у випадках недостатньої впевненості. Мультимодальна модель отримала чітку інструкцію не включати до результату об'єкти, які не є чітко видимими. Це дозволяє значно зменшити кількість помилкових ідентифікацій.

Для подальшої мінімізації галюцинацій було впроваджено ітеративний механізм перевірки результатів, відомий як “Chain-of-Verification” [28]. Реалізація цього етапу наведена на рис. 3.8.

```
const prompt = `A previous scan suggested these items are in the fridge:
${currentList.join(", ")}.
    Look at the image again carefully.
    Tasks:
    1. Remove any items that are NOT clearly visible in the image.
    2. Add items ONLY if they are clearly visible.
    3. Do not guess ingredients based on packaging or context.
    4. Return a deduplicated list of ingredient names in lowercase.
    Then:
    Return a JSON object with an 'ingredients' array.
    Each ingredient must have a 'name' (lowercase) and a 'confidence' score (0.0 to 1.0).
    Confidence rules:
    - 1.0: Crystal clear, unmistakable.
    - 0.8: Highly likely, but slightly blurry.
    - 0.5: Obstructed view or generic packaging.
    - Below 0.5 = poor visibility`
```

Рис. 3.8 Промпт для етапу верифікації

На цьому етапі модель виступає в ролі незалежного аудитора, який критично переглядає попередньо сформований список інгредієнтів та зіставляє його з візуальними доказами на зображенні. Використання такої стратегії самокорекції “Self-Refine” [29] дозволяє системі автоматично виправляти помилки розпізнавання без втручання користувача.

Важливим елементом оптимізації також було використання структурованого формату відповіді у вигляді JSON. Як показано у роботі «On the Planning Abilities of Large Language Models : A Critical Investigation» (Valmeekam та ін.) [30], перехід до структурованого виводу дозволяє ефективно вирішувати завдання вилучення даних. Така система перетворює ймовірнісні відповіді моделі у детерміновані об'єкти, придатні для подальшої програмної обробки.

Окрім того, до кожного об'єкта було додано показник впевненості. У дослідженні «Language Models (Mostly) Know What They Know» (Kadavath та ін.) [31] було обґрунтовано, що сучасні мовні моделі здатні до самооцінювання та визначення ймовірності правильності власних відповідей. Це створює додатковий рівень фільтрації даних, де інгредієнти з низьким показником впевненості можуть бути виключені або винесені на додаткове підтвердження користувачем.

Варто зазначити, що базовий запит рис. 3.6 продемонстрував настільки низьку стабільність формату JSON та високу похибку, що його було відхилено ще

на етапі прототипування. Тому для подальшого кількісного тестування й перевірки точності (підрозділ 3.4) за точку відліку взяли вже структурований і оптимізований запит на рис. 3.7, а далі порівнювали його з повною системою верифікації на рис. 3.8.

3.3. Реалізація функціоналу інтелектуального підбору рецептів

Після завершення етапів первинного розпізнавання та їх верифікації система переходить до реалізації інтелектуального пошуку рецептів. Основним принципом цієї фази є автоматичне співставлення наявних інгредієнтів із базою кулінарних рецептів та отримання доречного переліку страв. У програмній реалізації цей процес виконано на основі взаємодії клієнтської та серверної частини вебдодатку.

У клієнтській частині механізм підбору рецептів реалізовано за принципом реактивного керування станом, при якому система автоматично відслідковує зміни у масиві інгредієнтів. Приклад коду можливо побачити на рис. 3.9.

```
useEffect(() => {
    if (!hasLoaded || ingredients.length < 3) return
    const fetchRecipes = async () => {
      let apiUrl = `/api/recipes?ingredients=${ingredients.join(",")}`
      if (streak <= 2) apiUrl += "&maxReadyTime=20$sort=popularity"
      else if (streak <= 6) apiUrl += "&maxReadyTime=40&sort=healthiness"
      else apiUrl += "&sort=random&cuisine=asian,italian,mexican"
      try {
        const response = await fetch(apiUrl)
        const data = await response.json()
        if (Array.isArray(data)) {
          setRecipes(data)
        }
      } catch (err) {
        console.error("Failed to fetch recipes:", err)
      }
    }
    fetchRecipes()
  }, [ingredients, hasLoaded])
```

Рис. 3.9 Механізм підбору рецептів

Для наведеного лістингу коду був використаний хук useEffect, що ініціює запит на отримання рецептів лише після того, як застосунок повністю завантажується та кількість інгредієнтів досягає мінімального достатнього значення. Такий підхід дозволяє уникнути передчасних або надлишкових запитів та оптимізує використання зовнішнього API. Код реалізує клієнтський механізм динамічного формування запиту до підсистеми рекомендацій. Головним параметром є список інгредієнтів, які передаються у вигляді рядка запиту. Додатково система враховує допоміжний показник streak, який використаний для адаптації параметрів пошуку. При низькому показнику streak система формує запити на швидкі та популярні рецепти для полегшення приготування страв новим користувачам. На кожному наступному рівні показника, рецепти стають більш складними для покращення навиків користувача. Завдяки цьому реалізується елемент персоналізації рекомендацій, що дозволяє адаптувати результати пошуку до характеру взаємодії користувача із системою.

Запит, сформований на клієнтському рівні, передається до серверного маршруту, який є проміжним шаром між застосунком та зовнішнім сервісом рецептів. Це дозволяти ізолювати ключ доступу до API та централізувати обробку параметрів запитів. Зокрема це забезпечує безпечну інтеграцію сервісу в застосунок, що показано на рис. 3.10.

```
export async function GET(req: Request) {
    const { searchParams } = new URL(req.url)
    const ingredients = searchParams.get('ingredients')
    if(!ingredients) {
        return NextResponse.json({error: "No ingredients provided"}, {status: 400 })
    }
    const apiKey = process.env.SPOONACULAR_API_KEY
    const res = await fetch(
        `https://api.spoonacular.com/recipes/findByIngredients?
ingredients=${ingredients}&apiKey=${apiKey}`
    )
    const data = await res.json()
    return NextResponse.json(data)
};
```

Рис. 3.10 Функція для інтелектуального підбору рецептів

Серверний маршрут отримує параметр ingredients із рядка та перевіряє його на наявність. Якщо список сформовани коректно, сервер формує звернення до зовнішнього API Spoonacular, передаючи список продуктів та ключ доступу. У протилежному випадку, якщо список пустий, система повертає повідомлення про помилку з відповідним HTTP-статусом. Відповідь десеріалізується з формату JSON і повертається клієнтській частині у стандартизованому вигляді. Серверний модуль виконує функцію посередника між інтерфейсом користувача та базою рецептів, забезпечуючи контрольований обмін даними.

Функціонал інтелектуального підбору рецептів у розробленій системі реалізовано як послідовність взаємопов’язаних етапів:

- аналіз списку інгредієнтів,
- формування параметрів запиту,
- звернення до зовнішнього сервісу рецептів
- отримання набору кулінарних пропозицій.

Запропонований підхід дозволяє автоматизувати процес пошуку рецептів та забезпечити персоналізований характер рекомендацій.

3.4. Тестування та оцінка точності роботи розроблених алгоритмів

У даному розділі проведено оцінку ефективності розроблених алгоритмів, обробки візуальних даних та верифікації результатів. Основною метою тестування є визначення точності розпізнавання продуктів, а також аналіз впливу механізму повторної перевірки на якість отриманих результатів.

Для проведення експериментального дослідження було сформовано тестовий набір даних: зображення холодильників, що містять різну кількість продуктів та відрізнялися умовами зйомки. Для кожного зображення було сформовано еталонний список інгредієнтів, який використовувався для порівняння з результатами роботи системи.

Датасет складається загалом з 40 фотографій. 25 були підготовлені автором з попереднім контролем умов зйомки та набору продуктів, що дозволило

сформувати сценарії різної складності. 15 фотографій відносяться до категорії “низької складності”, де інгредієнти чітко видимі та не перекриваються. 10 фотографій відносяться до категорії “середньої складності”, де навмисно створювались умови для виникнення помилок розпізнавання (перекриття об'єктів, схожі упаковки тощо). Інші 15 фотографій були отримані від сторонніх користувачів, що дозволило наблизити експеримент до реальних умов використання системи. Ці фотографії відносятся до категорії “високої складності”. Деякі з авторів фотографій знаходяться за кордоном. Залучення таких даних дозволяє підтвердити географічну інваріантність системи та її здатність коректно функціонувати поза межами локального ринку.

Кожне зображення було оброблено системою двічі: на етапі первинного розпізнавання та після застосування механізму повторної верифікації. Отримані результати були зафіксовані та використані для подальшого аналізу точності системи та оцінки впливу додаткового етапу перевірки.

Для калібрування системи було сформовано еталонну вибірку “низької складності” зображень, приклад одного з яких наведено на рис. 3.11. Основна характеристика таких даних – мінімальна зашумленість та ідеальні умови для розпізнавання, що дозволяє оцінити базову точність алгоритму.

Рис. 3.11 Приклад вхідного зображення “низької” складності

Наведене зображення характеризується сукупністю ознак, що мінімізують візуальний шум і спрощують роботу мультимодальної моделі. Завдяки внутрішньому освітленню холодильника отримана висока яскравість і чіткість деталей. Відсутні глибокі тіні або сильні відблиски на пакуваннях, що дозволяє моделі безперешкодно аналізувати текстури та колірні гами продуктів. Відсутність перекриття об'єктів, завдяки їх відокремленому розташуванню на полицях, також полегшує процес аналізу зображень. Жоден об'єкт не загороджує інший, що гарантує видимість повного контуру кожного продукту та спрощує завдання семантичної сегментації. Більшість продуктів повернуті лицьовою стороною до камери, щоб забезпечити високу читабельність текстових маркувань та логотипів. Камера також розташована прямо навпроти полиць, що забезпечує ортогональний ракурс. Це мінімізує перспективні спотворення форми інгредієнтів, роблячи їх легко впізнаваними для алгоритмів комп'ютерного зору.

Для ілюстрації процесу тестування на рис. 3.12 наведено приклади вхідних зображень “високої” складності, що використовувались у дослідженні.

Рис. 3.12 Приклад вхідного зображення “високої” складності

Наведений рисунок ілюструє приклад вхідних даних, які класифікуються як об'єкти “високої” складності. Наведене зображення містить значну кількість візуальних чинників, які суттєво ускладнюють процес сегментації та ідентифікації об'єктів.

Основним фактором, який ускладнює процес ідентифікації, є щільність заповнення простору холодильника. Це призводить до часткового або повного перекриття деяких продуктів та створює проблему розпізнавання неповних контурів об'єктів. Таке розташування об’єктів вимагає від алгоритму здатності до контекстуального висновку. Крім того, на фотографії присутня велика кількість інгредієнтів у прозорих пакуваннях, які спричиняють оптичні спотворення.

Додаткову складність створює ракурс зйомки під кутом, що вносить перспективні викривлення геометрії продуктів. Різна орієнтація етикок обмежує можливість прямого зчитування тексту. Слід також зазначити наявність дрібних об'єктів у контейнерах та продуктів у сітчастих упаковках, що створює візуальний патерн, який важко класифікувати без глибокого аналізу сцени. Використання таких зображень у наборі даних дозволяє об’єктивно оцінити стійкість розробленої системи до реальних, неструктурованих умов використання.

Для проведення порівняльного аналізу та оцінки стабільності роботи системи були проаналізовано отримані метрики, приклад наведено в таблиці 3.1. Базуючись на дослідженні наведених вище прикладів (рис. 3.11 та рис. 3.12), було проведено серію тестів, результати яких підтверджують пряму залежність точності системи від різних факторів.

Табл. 3.1

Приклад систематизації отриманих метрик

Складність	Низька	Висока
Source of Truth (Реальні продукти)	Milk, coconut milk, caviar, bread, bacon, cream	Chicken breast , potatoes, onions, eggs, garlic sauce, chili sauce, sweet and sour sauce, ketchup, tomatoes, lemons, avocados, milk, pork loin, Pepsi Zero, mango, black caviar
Результат без перевірки	caviar, bread, coconut milk, cream, bacon, milk	cabbage, soda, yogurt, sliced ham, ketchup, sweet chili sauce, raw chicken, tomatoes, lemons, kiwis, milk, onions, potatoes, eggs
Алгоритм з перевіркою	caviar, coconut milk, cream, bacon, milk, bread	avocados, soda, mango, black caviar, sliced ham, ketchup, sweet and sour sauce, raw chicken, tomatoes, lemons, kiwis, milk, onions, potatoes, eggs, chili sauce
TP База	6	9
FP База	0	6
FN База	0	7
Середня точність База	100%	43%
TP Покращено	6	12

Складність	Низька	Висока
FP Покращено	0	4
FN Покращено	0	4
Середня точність Покращенно	100%	60%

Щоб візуально перевірити отримані метрики й показати, як система працює на практиці, було проаналізовано складне зображення рис. 3.12 прямо у веб-інтерфейсі розробленого застосунку. На рис. 3.13 і рис. 3.14 показано роботу «Fridge Scanner» у двох режимах.

Як видно з рис. 3.13, під час першого сканування система припустилася кількох помилок (галюцинацій) і «знайшла» на фото те, чого там не було: "cabbage" (капуста) та "kiwis" (ківі). При цьому середня впевненість моделі становила 68%.

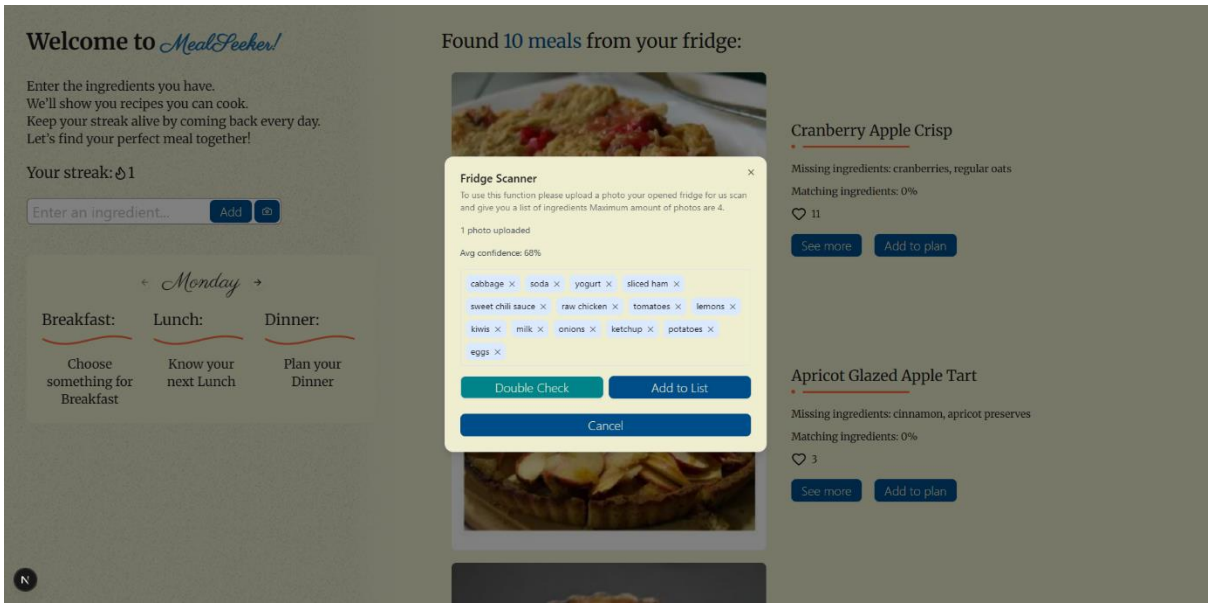

Рис. 3.13 Результат первинного розпізнавання інгредієнтів

Після ввімкнення перевірки "Double Check" показано на рис. 3.14 система ще раз переглянула сцену. У підсумку вона прибрала хибні теги та додала

інгредієнти, які спочатку пропустила, зокрема "black caviar" (чорна ікра) та "mango" (манго). Рівень впевненості моделі піднявся до 87%.

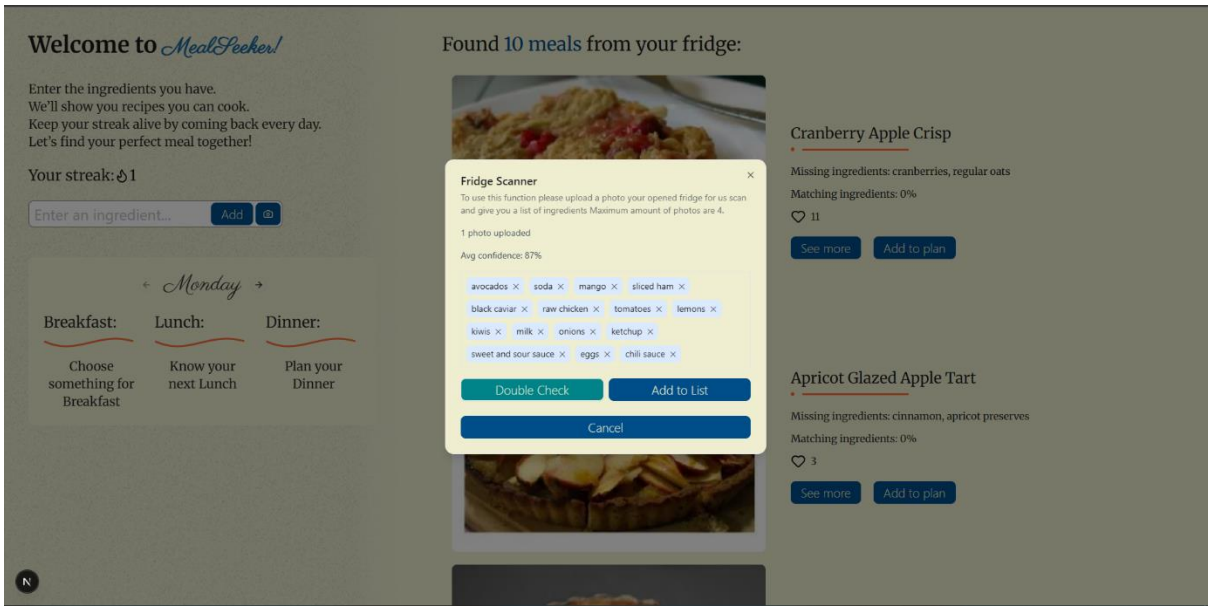

Рис. 3.14 Результат роботи авторського алгоритму верифікації

Це добре узгоджується з даними з табл. 3.1: коли модуль верифікації підключено, стає менше хибнопозитивних спрацьовувань (FP), а фінальний список продуктів для подальшого пошуку рецептів виходить помітно точнішим.»

Для проведення об'єктивного аналізу ефективності розроблених алгоритмів було застосовано метод порівняльного тестування. Процес оцінки складається з декількох послідовних етапів. Для кожної фотографії тестового датасету було сформовано перелік фактично наявних продуктів шляхом візуального аналізу. Цей набір даних виступає як “джерело істини” (Source of Truth), на основі якого розраховуються всі подальші метрики точності.

На першому етапі вхідні зображення обробляються механізмом аналізу. Отримані результати зіставляються з еталонними даними для визначення показників: True Positive (TP) – кількість коректно ідентифікованих об'єктів; False Positive (FP) – випадки помилкової ідентифікації; False Negative (FN) – кількість пропущених системою об'єктів. Після завершення механізму первинного аналізу проводиться повторна обробка даних із застосуванням механізму верифікації.

Отримані результати аналогічно порівнюються для виявлення динаміки зміни точності. На основі отриманих значень TP, FP та FN для кожної ітерації розраховано середню точність у відсотковому еквіваленті.

Базуючись на розрахованій метриці, було проведено порівняльний аналіз роботи базового алгоритму перевірки та вдосконалення моделі з механізмом верифікації. Результати візуалізовано на рис. 3.15-3.17, що дозволяє наочно оцінити ефективність проведеної оптимізації.

На рис. 3.15 представлено порівняння кількості хибнопозитивних результатів, отриманих на етапі первинного аналізу та після застосування механізму верифікації. Як видно з графіка, використання повторної перевірки дозволяє суттєво зменшити кількість помилкових ідентифікацій (з понад 70 до 22 випадків), що свідчить про ефективність впровадження фільтра впевненості та механізму повторної перевірки. У первинному механізмі спостерігалася тенденція до "надлишкового розпізнавання", коли випадкові відблиски або елементи пакування інтерпретувалися як інгредієнти. Удосконалений алгоритм успішно нівелює цей ефект, що критично важливо для формування коректного списку рецептів.

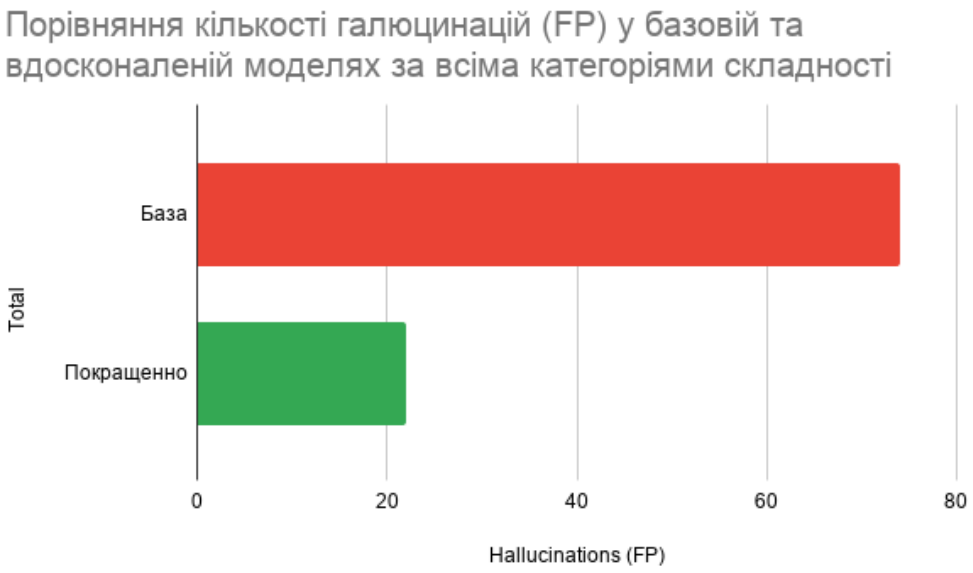

Рис. 3.15 Порівняння кількості хибнопозитивних результатів

На рис. 3.16 наведено порівняння кількості правильних та помилкових результатів для базового та вдосконаленого підходу. Аналіз графіка показує, що після застосування механізму верифікації спостерігається зростання кількості правильних визначень при одночасному зменшенні помилкових. Це свідчить про підвищення загальної точності системи. Механізм верифікації не просто видаляє сумнівні результати, а допомагає моделі краще фокусуватися на деталях, які були пропущені при первинному швидкому скануванні.

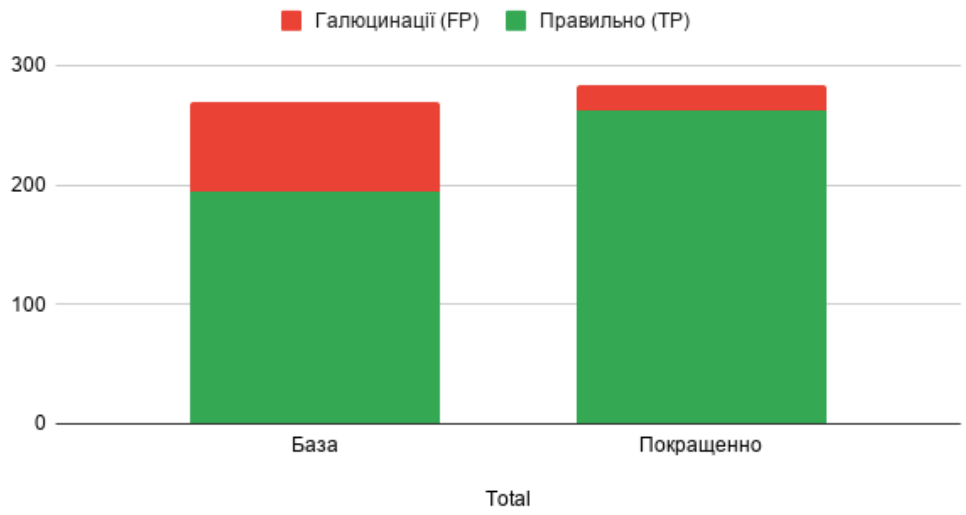

Рис. 3.16 Порівняння кількості правильних та помилкових результатів

На рис. 3.17 представлено залежність точності розпізнавання від складності вхідних даних. Як видно з результатів, у випадку простих зображень система демонструє високу точність навіть без додаткової перевірки. Однак, зі зростанням складності, точність базової моделі суттєво знижується, тоді як використання механізму верифікації дозволяє зберігати стабільно високі показники навіть у складних умовах. Це підтверджує гіпотезу про те, що ітеративний промпт-інжиніринг та декомпозиція візуальної сцени за полицями є ефективним рішенням для подолання обмежень стандартних методів комп'ютерного зору в неструктурованих середовищах.

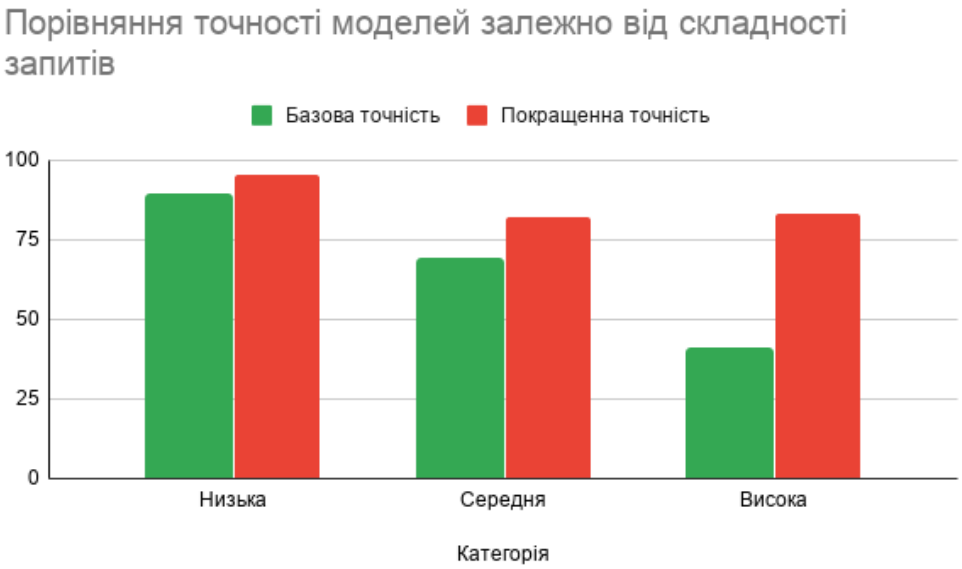

Рис. 3.17 Залежність точності розпізнавання від складних вхідних даних

Отримані експериментальні дані верифікують ефективність запропонованого методу ітераційної обробки. Впровадження механізму аудиту дозволило досягти високої прецизійності розпізнавання, мінімізуючи вплив візуальних шумів та артефактів, що є критично важливим для стабільної роботи алгоритмів у неструктурованих середовищах із високим рівнем складності.

ВИСНОВКИ

У ході виконання дипломної роботи було розв’язано актуальну науково-практичну задачу підвищення точності ідентифікації харчових продуктів у системах автоматичного підбору рецептів. За результатами дослідження зроблено наступні висновки.

Проаналізовано сучасні підходи до розпізнавання об'єктів та встановлено, що традиційні моделі комп'ютерного зору часто мають обмеження в неструктурованих середовищах (наприклад, щільно заповнений холодильник). Це підтвердило доцільність переходу до мультимодальних моделей штучного інтелекту, що здатні обробляти контекстні дані.

Досліджено можливості мультимодальної моделі Gemini, яка поєднує візуальне сприйняття з глибоким розумінням природної мови. Встановлено, що ключовим фактором якості ідентифікації є якість вхідного промпту та спосіб подачі візуальних даних.

Обґрунтовано та вдосконалено метод ітераційного промпт-інжинірингу. Запропонований підхід, що базується на декомпозиції візуальної сцени (розбиття зображення на зони або полиці) та механізмі верифікації, дозволяє моделі фокусуватися на деталях, які ігноруються при загальному скануванні. Це дозволило суттєво знизити відсоток помилкових розпізнавань моделі.

Розроблено інформаційну систему у вигляді вебзастосунку. Архітектура системи забезпечує зручну взаємодію користувача з ШІ-моделлю: від завантаження фото до отримання списку рецептів, адаптованих під наявні інгредієнти.

Проведено експериментальну перевірку розробленого методу на створеному наборі даних. Результати тестування продемонстрували, що використання механізму верифікації та декомпозиції сцени дозволяє підвищити загальну точність ідентифікації продуктів у середньому на 20,6%, особливо в умовах низької освітленості або часткового перекриття об’єктів.

Отримані результати підтверджують гіпотезу про те, що поєднання сучасних мультимодальних моделей із гнучкими стратегіями запитів є ефективним інструментом для побутової автоматизації та персоналізованого підбору харчування.

ПЕРЕЛІК ПОСИЛАНЬ

1. AI Usage Statistics: 2026 Data : [електронний ресурс]. URL: https://explodingtopics.com/blog/ai-usage-statistics (дата звернення: 24.04.2026).
2. Food Allergy Statistics and Trends : [електронний ресурс]. URL: https://worldmetrics.org/food-allergy-statistics/ (дата звернення: 24.04.2026).
3. The Lancet Public Health: Planetary health and food systems : [електронний ресурс]. URL: https://www.thelancet.com/journals/lanpub/article/PIIS2468-2667(24)00163-4/fulltext (дата звернення: 24.04.2026).
4. JMIR Formative Research: AI in Dietary Assessment : [електронний ресурс]. URL: https://formative.jmir.org/2025/1/e75395 (дата звернення: 24.04.2026).
5. MDPI Sensors: Deep Learning for Food Recognition : [електронний ресурс]. URL: https://www.mdpi.com/1424-8220/25/2/449 (дата звернення: 24.04.2026).
6. arXiv: Leave No Patient Behind: Multimodal LLMs : [електронний ресурс]. URL: https://arxiv.org/abs/2404.07214 (дата звернення: 24.04.2026).
7. Springer: Biosemiotics and Artificial Intelligence : [електронний ресурс]. URL: https://link.springer.com/article/10.1007/s12304-020-09388-7 (дата звернення: 24.04.2026).
8. Scalable Path: ChatGPT Architecture Explained : [електронний ресурс]. URL: https://www.scalablepath.com/ai/chatgpt-architecture-explained (дата звернення: 24.04.2026).
9. JMIR: Personalized Nutrition and AI : [електронний ресурс]. URL: https://www.jmir.org/2025/1/e77893 (дата звернення: 24.04.2026).
10. arXiv: Advanced Visual Understanding in LLMs : [електронний ресурс]. URL: https://arxiv.org/abs/2507.19024 (дата звернення: 24.04.2026).
11. ResearchGate: Hallucination of Multimodal Large Language Models : [електронний ресурс]. URL: https://www.researchgate.net/publication/380263724_Hallucination_of_Multimodal_Large_Language_Models_A_Survey (дата звернення: 24.04.2026).
12. ResearchGate: Comparative Analysis of Leading AI Models 2025 : [електронний ресурс]. URL: https://www.researchgate.net/publication/392160200_The_Most_Advanced_AI_Models_of_2025_-Comparative_Analysis_of_Gemini_25_Claude_4_LLaMA_4_GPT-45_DeepSeek_V31_and_Other_Leading_Models (дата звернення: 24.04.2026).
13. ResearchGate: A Review of Recipe Recommendation Methods : [електронний ресурс]. URL: https://www.researchgate.net/publication/391974878_A_Review_of_Recipe_Recommendation_Methods (дата звернення: 24.04.2026).
14. SuperCook: Recipe Search by Ingredients : [електронний ресурс]. URL: https://www.supercook.com/#/desktop (дата звернення: 24.04.2026).
15. Yummly: Personalized Recipe Recommendations : [електронний ресурс]. URL: https://www.yummlyrecipes.com/ (дата звернення: 24.04.2026).

16. Tasty: Easy Recipes and Cooking Videos : [електронний ресурс]. URL: https://tasty.co/ (дата звернення: 24.04.2026).
17. arXiv: LLM-Based Zero-Shot Recognition : [електронний ресурс]. URL: https://arxiv.org/abs/2312.11805 (дата звернення: 24.04.2026).
18. Google AI for Developers: Gemini API Docs : [електронний ресурс]. URL: https://ai.google.dev (дата звернення: 24.04.2026).
19. TypeScript Documentation: Language Features : [електронний ресурс]. URL: https://www.typescriptlang.org/docs/ (дата звернення: 24.04.2026).
20. React Documentation: User Interface Library : [електронний ресурс]. URL: https://react.dev/ (дата звернення: 24.04.2026).
21. Next.js Documentation: The React Framework : [електронний ресурс]. URL: https://nextjs.org/docs (дата звернення: 24.04.2026).
22. Tailwind CSS: Utility-First CSS Framework : [електронний ресурс]. URL: https://tailwindcss.com/docs (дата звернення: 24.04.2026).
23. shadcn/ui: Reusable UI Components : [електронний ресурс]. URL: https://ui.shadcn.com/ (дата звернення: 24.04.2026).
24. Spoonacular API: Food and Recipe Documentation : [електронний ресурс]. URL: https://spoonacular.com/food-api/docs (дата звернення: 24.04.2026).
25. arXiv: Evaluation of Multimodal Models : [електронний ресурс]. URL: https://arxiv.org/pdf/2410.20211 (дата звернення: 24.04.2026).
26. ResearchGate: XML and JSON in Cloud-Based Systems : [електронний ресурс]. URL: https://www.researchgate.net/publication/394418035_XML_and_JSON_in_Cloud-Based_Systems (дата звернення: 24.04.2026).
27. arXiv: Chain-of-Thought Prompting in LLMs : [електронний ресурс]. URL: https://arxiv.org/pdf/2201.11903 (дата звернення: 24.04.2026).
28. arXiv: Chain-of-Verification Reduces Hallucination : [електронний ресурс]. URL: https://arxiv.org/pdf/2309.11495 (дата звернення: 24.04.2026).
29. arXiv: Self-Refine: Iterative Refinement with LLMs : [електронний ресурс]. URL: https://arxiv.org/pdf/2303.17651 (дата звернення: 24.04.2026).
30. arXiv: Tree of Thoughts: Deliberate Problem Solving : [електронний ресурс]. URL: https://arxiv.org/pdf/2305.15771 (дата звернення: 24.04.2026).
31. arXiv: Inner Monologue: Embodied AI Reasoning : [електронний ресурс]. URL: https://arxiv.org/pdf/2207.05221 (дата звернення: 24.04.2026).