Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБЛЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ ANDROID ДЛЯ РОЗПІЗНАВАННЯ ТА ПОЯСНЕННЯ СКЛАДУ ХАРЧОВИХ ПРОДУКТІВ З АВТОМАТИЗОВАНИМ ПЕРЕКЛАДОМ УКРАЇНСЬКОЮ МОВОЮ
(тема)

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-22-2
Соколова А. М.
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Яковлева О. В.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____ Кобилін О. А.
(підпис) (прізвище, ініціали)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти перший (бакалаврський)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Соколовій Аліні Миколаївні
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення мобільного застосунку Android для розпізнавання та пояснення складу харчових продуктів з автоматизованим перекладом українською мовою

затверджена наказом університету від 26 травня 2026 року № 435Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 19 червня 2026 р.
3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, результати аналізу існуючих програмних рішень у предметній області, відкриті бази даних харчових продуктів та добавок (зокрема Open Food Facts), набори відкритих даних з репозиторіїв (Kaggle), офіційна документація до великих мовних моделей та середовищ розробки мобільних і серверних застосунків, мови програмування Python, JavaScript та TypeScript, мультимодальні великі мовні моделі (Gemini, GPT, Claude, DeepSeek, Qwen), фреймворк для мобільної розробки React Native, інструментарій Expo, вебфреймворк FastAPI, хмарний сервіс для проєктування інтерфейсів Figma, інструмент для побудови UML-діаграм та блок-схем draw.io, середовища розробки VS Code, хмарна платформа для розгортання серверної частини Railway.

4. Перелік питань, що потрібно опрацювати в роботі
1. Поточний стан проблеми автоматичного аналізу складу харчових продуктів.
2. Розробка методу для розпізнавання та пояснення складу харчових продуктів з автоматизованим перекладом українською мовою на основі упаковки.
3. Розроблення мобільного застосунку Android для розпізнавання та пояснення складу харчових продуктів з автоматизованим перекладом українською мовою.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) опис проблеми, наявність готових рішень на ринку, методи та підходи для детектування та розпізнавання тексту, постановка задачі, постановка задачі експериментів, огляд наборів даних, створення власного набору даних, розробка критерію оцінювання якості, розробка запиту, відбір моделей, автоматизування оцінювання роботи моделей, оптимізація часу роботи моделей, вибір програмного забезпечення, розробка архітектури застосунку, приклади роботи застосунку.

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	13.04.2026	
2	Аналіз завдання, підбір літератури	14.04.26-16.04.26	
3	Аналіз літератури з проблеми, що вирішується	17.04.26-19.04.26	
4	Аналіз існуючих методів розпізнавання	20.04.26-22.04.26	
5	Формування кроків вирішення проблеми	23.04.26-05.05.26	
6	Програмна реалізація	26.04.26-20.05.26	
7	Аналіз отриманих результатів	21.05.26-04.06.26	
8	Оформлення пояснювальної записки	05.06.26-09.06.26	
9	Перевірка на нормоконтроль	10.06.26-19.06.26	
10	Перевірка на плагіат	11.06.26-30.06.26	
11	Рецензування	20.06.26-30.06.26	
12	Підготовка презентації та доповіді	20.06.26-30.06.26	
13	Занесення роботи в електронний архів	30.06.26-09.07.26	
14	Попередній захист кваліфікаційної роботи	30.06.26-09.07.26	

Дата видачі завдання 13 квітня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Яковлева О. В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 120 с., 28 табл., 34 рис., 7 дод., 91 джерело.

АНАЛІЗ СКЛАДУ, ІНГРЕДІЄНТ, МОБІЛЬНИЙ ЗАСТОСУНОК, МУЛЬТИМОДАЛЬНА ВЕЛИКА МОВНА МОДЕЛЬ, РОЗПІЗНАВАННЯ ТЕКСТУ, ХАРЧОВИЙ ПРОДУКТ, ANDROID, GEMINI 2.5 PRO.

Об’єктом роботи є процес розпізнавання та аналізу складу харчових продуктів на основі зображення упаковки.

Метою роботи є розроблення мобільного застосунку Android для розпізнавання та пояснення складу харчових продуктів з автоматизованим перекладом українською мовою.

Під час роботи використано: MLLMs для розпізнавання та аналізу, підхід LLM-as-a-Judge для оцінювання, методи розробки мобільних застосунків.

Практична цінність мобільного застосунку полягає в автоматизації розпізнавання та пояснення складу харчових продуктів.

Розроблено мобільний застосунок, що забезпечує зчитування та пояснення складу продукту харчування, його переклад українською мовою та додаткову перевірку за персональним запитом за допомогою MLLM.

Область застосування: свідоме споживання та турбота про здоров'я, підтримка збалансованого раціону, інформований вибір продуктів у магазині.

COMPOSITION ANALYSIS, INGREDIENT, MOBILE APPLICATION, MULTIMODAL LARGE LANGUAGE MODEL, TEXT RECOGNITION, FOOD PRODUCT, ANDROID, GEMINI 2.5 PRO.

The objective of this work is the recognition and analysis of food product composition based on packaging images.

The goal of this work is to develop an Android mobile application for the recognition and explanation of food product composition with automated translation into Ukrainian.

The following were used during the work: MLLM for recognition and analysis, the LLM-as-a-Judge approach for evaluation, and mobile application development methods.

The practical value of the mobile application lies in automating the recognition and explanation of food product composition.

A mobile application has been developed that recognizes and explains food product composition, translates it into Ukrainian, and supports additional verification via user requests using MLLM.

Applications: conscious consumption and health awareness, balanced diet support, informed product selection in stores.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

$ – долар США

М – мільйон

с – секунда

ПЗ – програмне забезпечення

Android – операційна система для мобільних пристроїв від Google

API – Application Programming Interface (інтерфейс програмування застосунків)

APK – Android Package Kit (пакет встановлення застосунку для Android)

HTTP – Hypertext Transfer Protocol (протокол передачі гіпертексту)

iOS – операційна система для мобільних пристроїв від Apple

JSON – JavaScript Object Notation (текстовий формат обміну даними)

LLM – Large Language Model (велика мовна модель)

MLLM – Multimodal Large Language Model (мультимодальна велика мовна модель)

MLLMs – Multimodal Large Language Models (мультимодальні великі мовні моделі)

OCR – Optical Character Recognition (оптичне розпізнавання символів)

ВСТУП

Сучасний споживач щодня стикається з великою кількістю упакованих харчових продуктів, рішення про придбання яких часто приймається за кілька секунд. Водночас саме склад продукту суттєво впливає на здоров’я людини в довгостроковій перспективі.

З формальної точки зору проблема інформування споживача є частково вирішеною. У країнах європейського союзу є регламент, який зобов’язує виробників наносити на упаковку детальну інформацію про склад, харчову цінність та алергени. В Україні також існує відповідне законодавство щодо оформлення етикеток. Проте виконання цих норм не гарантує, що пересічний покупець легко розбереться у складі конкретного продукту. Аналіз ринку показує, що значна частина споживачів або взагалі не розуміє дані на етикетках, або інтерпретує їх лише частково. Складність викликають дрібний шрифт, складна хімічна термінологія та великий обсяг тексту одночасно кількома мовами.

Окремою проблемою є мовний бар’єр. За час воєнного стану на полицях українських магазинів з’явилась значна кількість імпортних товарів, етикетки яких часто відсутні українською мовою або подаються у вигляді дрібного наклейкового стікера. Для українців, що перебувають за кордоном, ситуація ще складніша, бо упаковка може бути виключно мовою країни перебування, а самостійно розшифрувати склад без знання відповідної термінології практично неможливо.

Існуючі мобільні застосунки для оцінювання харчових продуктів лише частково вирішують цю проблему. Більшість із них працює на основі сканування штрих-кодів та пошуку в базах даних, що робить їх непридатними для роботи з новими, локальними або рідкісними товарами. Окремі нові застосунки, що використовують мультимодальні великі мовні моделі (multimodal large language models, MLLMs) для аналізу фотографій етикеток, працюють нестабільно та не орієнтовані на українського споживача. Жоден із розглянутих застосунків не

забезпечує повноцінного пояснення кожного інгредієнта українською мовою з урахуванням його можливої користі та шкоди для здоров’я.

Наразі активно розвивається напрям MLLMs, здатних одночасно аналізувати зображення та текст. На відміну від класичного оптичного розпізнавання символів (Optical Character Recognition, OCR), який лише розпізнає символи та потребує окремої обробки зображення, MLLMs виконують розпізнавання та смисловий аналіз за один крок. Провідні моделі сімейств GPT, Gemini та Claude вже продемонстрували здатність якісно зчитувати текст із фотографій упаковок та виконувати змістовний аналіз складу, що відкриває можливість для створення принципово нового класу застосунків, незалежних від баз даних.

Актуальність роботи полягає у тому, що поточний стан ринку мобільних застосунків не пропонує рішення, яке б забезпечувало повноцінне розпізнавання та пояснення складу харчових продуктів українською мовою із будь-якої упаковки. У зв’язку з цим дана робота присвячена розробленню мобільного застосунку Android, який на основі фотографії упаковки автоматично отримує текст складу продукту, перекладає його українською мовою та виконує смисловий аналіз кожного інгредієнта з використанням моделі Gemini 2.5 Pro. Окрім базового аналізу, застосунок надає можливість додаткової персоналізованої перевірки продукту відповідно до індивідуальних потреб користувача.

Особлива увага приділяється аналізу якості роботи різних MLLMs для задачі розпізнавання та аналізу складу харчових продуктів, а також визначенню оптимальної конфігурації обраної моделі для забезпечення найкращого балансу між точністю аналізу та швидкістю роботи.

1 ПОТОЧНИЙ СТАН ПРОБЛЕМИ АВТОМАТИЧНОГО АНАЛІЗУ СКЛАДУ ХАРЧОВИХ ПРОДУКТІВ

1.1 Опис проблеми та підходи до вирішення

Сучасний споживач постійно стикається з великою кількістю упакованих харчових продуктів, від простих цукерок, батончиків і печива до складних напівфабрикатів та готових страв. У більшості випадків рішення про покупку приймається за кілька секунд, хоча саме склад продукту суттєво впливає на здоров’я людини в довгостроковій перспективі. За даними Всесвітньої організації охорони здоров’я, нездорове харчування є одним із ключових факторів ризику розвитку неінфекційних захворювань, таких як серцево-судинні хвороби, діабет та деякі види раку [1]. Це підкреслює важливість того, щоб інформація про склад продуктів не просто формально була присутня на упаковці, а реально використовувалась споживачем для усвідомленого вибору.

З формальної точки зору проблема наче вже вирішена, бо у країнах ЄС та багатьох інших державах діють регламенти, які зобов’язують виробників наносити на упаковку детальну інформацію про склад, харчову та енергетичну цінність, алергени тощо. Прикладом є Регламент (ЄС) №1169/2011, який встановлює вимоги до того, як саме має бути подана інформація для споживача [2, 3]. В Україні також діє закон, який висвітлює вимоги до оформлення етикеток [4]. Однак на практиці навіть виконання цих норм не гарантує, що пересічний покупець легко розбереться в складі конкретного продукту.

Ряд досліджень показує, що значна частина споживачів або взагалі не розуміє дані на етикетках, або розуміє їх лише частково. Так, у роботі R. Goyal зазначається, що понад половина опитаних респондентів не розуміють інформацію на харчових етикетках, а ще значна частина лише частково її інтерпретує [5]. Інші дослідження відзначають проблеми з дрібним шрифтом, спеціальною термінологією та складними формулюваннями, що ускладнюють сприйняття навіть для мотивованих споживачів [6, 7]. На українському ринку

помічаються аналогічні проблеми. У підсумку інформація формально є доступною, але фактично часто залишається неінтерпретованою.

Також поширеною проблемою є те, що продавці часто не можуть пояснити значення певного інгредієнта, яка якість товару, як він впливає на організм, тощо. Особливо це помічається, коли мова йде про імпортні товари і інформація про них представлена іноземною мовою [8]. Таким чином, окремим чинником є мовний бар'єр. За час війни на полицях магазинів з'явилось багато імпортних товарів, тож українські споживачі частіше почали купувати іноземні продукти харчування, бо це стало більш доступно, і в деяких випадках навіть дешевше ніж товари українських постачальників. Етикетки таких товарів можуть бути надруковані кількома іноземними мовами, а український варіант поданий у вигляді маленького наклейкового стікера з дрібним шрифтом. Для українців, що перебувають за кордоном, ситуація ще складніша, бо упаковка може бути виключно мовою країни перебування, а розшифрувати склад «на око» без знання термінів та харчових добавок практично неможливо.

Сучасні тенденції свідомого споживання та турбота про здоров'я формують потребу в інструментах, які допомагають аналізувати якість та корисність продуктів харчування для підтримки здорового способу життя та збалансованого раціону.

Типові підходи, які сьогодні використовує споживач для аналізу складу продуктів, можна умовно поділити на кілька груп. Найпростіший і найпоширеніший спосіб це просто уважно прочитати список інгредієнтів на упаковці. На практиці він має ряд недоліків, таких як дрібний шрифт, велика кількість тексту та кілька мов одночасно, використання спеціальних хімічних чи технологічних термінів, відсутність пояснення, корисний інгредієнт чи небажаний. У результаті покупець часто або пропускає цей етап, або обмежується поверхневим переглядом, орієнтуючись лише на бренд, ціну та рекламні написи.

Другий підхід це ввести назву в пошуковий рядок Google і подивитися, що пишуть про продукт. Зазвичай відкриваються сторінка виробника з офіційним

описом, відгуки на інтернет-торгових платформах, іноді огляди блогерів або статті з аналізом складу. Цей спосіб дозволяє знайти додаткову інформацію, але є неструктурованим, бо різні сайти можуть давати суперечливі дані, а розібратися в кількох вкладках на телефоні прямо в магазині досить складно.

Ще один варіант сфотографувати етикетку і пропустити її через мобільний перекладач. Це частково знімає мовний бар'єр, але не вирішує проблему розуміння суті інгредієнтів. Перекладена назва «аскорбінова кислота» чи «карбоксиметилцелюлоза» часто нічого не говорить неспеціалісту про користь чи ризики, а побудова повноцінного аналізу (що можна, що небажано, що точно протипоказано) все одно залишається на совісті самого користувача.

Також останніми роками з'явився клас застосунків, які дозволяють сканувати штрих-код продукту та отримувати агреговану інформацію про його якість, поживну цінність, наявність добавок чи алергенів. Основний недолік таких систем полягає в тому, що вони зазвичай залежать від наявності продукту в базі. Якщо товар рідкісний, локальний або новий, інформація може бути відсутня або неповною. Детальніший аналіз таких застосунків буде представлено в наступному підрозділі.

У деяких випадках споживачі намагаються розшифровувати склад поодинці і вводять у пошук конкретні назви добавок, читають статті, дивляться списки шкідливих Е-добавок тощо. Це дозволяє отримати більше деталей, але вимагає значних часових затрат і базового розуміння, як зіставити всю цю інформацію з власним станом здоров'я.

Окремо можна виділити використання MLLMs. У цьому підході споживач робить фото етикетки або виписує список інгредієнтів і передає його у вигляді запиту в діалогову систему з проханням пояснити, наскільки продукт є безпечним, корисним чи придатним за наявних обмежень. Проте на практиці такий варіант доступний переважно «просунутим» користувачам, бо по-перше, не всі споживачі взагалі знають про існування подібних інструментів штучного інтелекту або не довіряють їм. По-друге, необхідно вміти сформулювати коректний запит, чітко описати свої потреби та обмеження. По-третє,

інтерпретація відповіді також потребує критичного мислення і розуміння того, що модель може помилятися або надавати неповну інформацію. Тому, незважаючи на високий потенціал MLLM для персоналізованого аналізу складу, на даному етапі цей підхід залишається інструментом для відносно вузького кола технічно підготовлених користувачів.

1.2 Наявність готових рішень на ринку

Перед розробкою власного мобільного застосунку доцільно проаналізувати вже наявні рішення, які допомагають споживачам оцінювати склад харчових продуктів. Наведена інформація актуальна станом на початок лютого 2026 року та отримана під час пошуку з території України. Пошук застосунків здійснювався у Google Play, через пошукову систему Google, а також за допомогою сервісів Perplexity і ChatGPT за запитами на кшталт «застосунок для розшифровки складу продуктів», «сканер продукту харчування», «food label scanner», тощо. У результаті було виявлено низку популярних застосунків: Foodstr, Open Food Facts, PureCheck, Nutri-Scan, Food Scanner, Food Lookup, Food Analyzer тощо. Для уточнення довідкових відомостей використовувалися аналітичні сервіси Sensor Tower [9] (країна виробник, дата публікації, платформи, країни поширення) та AppBrain [10] (кількість завантажень і популярність). Отримані дані та результати тестування узагальнено у порівняльних таблицях (додаток А), які відображають методи пошуку застосунків, загальну характеристику застосунків відібраних для аналізу, їх функціональні можливості та враження від використання.

Український застосунок Foodstr (опублікований 16.09.2020), призначений для оцінювання корисності продуктів за 100-бальною шкалою та аналізу харчових добавок за допомогою штрих-коду [11], має одну з найбільших баз українських товарів (понад 600 000 найменувань). Наразі завантажити програму з офіційних маркетів Google Play та App Store неможливо.

Міжнародний безкоштовний додаток Open Food Facts базується на відкритій базі даних (понад 3,5 млн продуктів) і дозволяє отримувати інформацію про склад, харчову цінність, харчові добавки, алергени та екологічний вплив [12]. Користувач може сканувати штрих-код або шукати продукт за назвою. Практичне тестування показало, що формат подання інгредієнтів не зручний, інгредієнти представлені на 3 різних мовах, не всі інгредієнти виділені (рис. 1.1). Додаткове тестування виявило випадки повної відсутності продуктів або інформації про їхній склад у базі даних. Open Food Facts є потужним відкритим інструментом, але якість даних сильно залежить від наповненості бази та коректності введеної інформації.

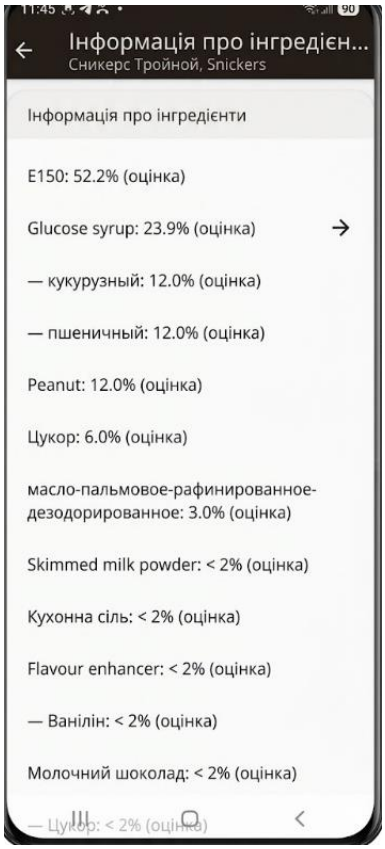


Рисунок 1.1 – Аналіз інгредієнтів продукту «Snickers» за допомогою застосунку Open Food Facts

Додаток Yuka (Франція, 2017) є одним із найвідоміших світових рішень для сканування штрих-кодів продуктів та косметики [13]. Yuka аналізує склад продуктів і відносить їх до категорій «відмінно», «добре», «посередньо» або

«погано». Через регіональні обмеження сервіс є недоступним в Україні, тому провести його практичне тестування було неможливо.

Nutri-Scan дає змогу отримати список інгредієнтів і їх відносну вагу в продукті [14]. Працює виключно через сканування штрих-кодів, і продукт шукається в базі. Практичне тестування показало, що при скануванні різних товарів застосунок часто видавав помилку «This product does not contain food ingredients … » (рис. 1.2). Також багато товарів взагалі не було знайдено через обмеженість бази даних.

Рисунок 1.2 – Помилка при використанні Nutri-Scan

PureCheck працює як сканер харчових продуктів і косметики, що дозволяє зчитувати штрих-коди або виконувати ручний пошук для отримання інформації про склад, харчову цінність і алергени [15]. Під час тестування виявлено обмежене покриття українських товарів і недостатню якість аналізу (мовні неточності, відсутність чітких пояснень інгредієнтів).

Tasty Healthy – Food Analyzer – це мобільний застосунок для оцінювання якості харчових продуктів за фотографією етикетки [16]. Застосунок формує

загальну оцінку корисності продукту, пояснює вплив інгредієнтів і показників поживної цінності (жири, цукри, білки тощо). Під час тестування відзначено зручний інтерфейс і наявність детальної аналітики, проте виявлено проблеми зі зчитуванням інгредієнтів, затримки в обробці фото та відсутність підтримки української мови.

Food Scanner орієнтований насамперед на підрахунок калорій та контроль макронутрієнтів, а також перевірку інгредієнтів [17]. Є формально безкоштовним (є 3 безкоштовні спроби сканування на тиждень), проте використовує модель підписки та реклами. Основний спосіб взаємодії – сканування штрих-коду продукту. За результатами тестування на українських продуктах було встановлено, що частина з них знаходиться, але не надається детальний аналіз, а деякі товари взагалі відсутні в базі даних. Це свідчить про те, що застосунок не орієнтований на український ринок і має обмежене наповнення бази товарів.

Food Lookup орієнтований насамперед на надання харчової та калорійної інформації про продукти та є платним [18]. Під час тестування застосунок знайшов продукт за штрих-кодом, однак надав обмежену інформацію, переважно щодо калорійності та макронутрієнтів. Детальної інформації про інгредієнти та їх пояснення не представлено. Поглиблений аналіз не було виконано і до узагальнюючих таблиць в додатку А не додано, бо за за підходом він подібний до Open Food Facts і Nutri-Scan, але якість роботи нижча.

Food Label Scanner & Checker (маркетингова назва – Nutelligence) – це застосунок, який пропонує аналізувати саме список інгредієнтів з етикетки за допомогою MLLM, а не покладатися виключно на базу штрихкодів [19]. Ідея цього рішення дуже близька до концепції кваліфікаційної роботи. Під час тестування було виявлено обмеження на кількість безкоштовних сканувань через ліміти MLLM. Практична перевірка на українських товарах показала суттєві недоліки в роботі системи. Програма некоректно розпізнавала текст і пропускала частину інгредієнтів. У результаті користувач отримував неповний список складників. Розшифровка та пояснення надавалися лише для окремих слів у

переліку.

Застосунок Read the Labels – Food Scanner – застосунок для сканування штрих-коду та оцінки продукту за складом [20]. Практичне тестування не виконувалося, оскільки застосунок платний і навіть для пробного доступу потрібно оформлювати підписку.

BioBrief (Scanner Alimente) дозволяє шукати продукти за штрих-кодом або через фотографію етикетки [21]. Під час тестування сервіс успішно зчитував інгредієнти та показував результати аналізу. Проте застосунок не орієнтований на український ринок. В інтерфейсі відсутня українська мова, а в результатах часто трапляється змішування іноземних слів. Система зазвичай надає лише загальну оцінку без детальної розшифровки кожного складника.

Eat Safe – AI Food Scanner (Eat Safe: AI Additives Scanner) перевіряє складу продуктів за допомогою сканування штрих-коду або фотографування списку інгредієнтів, з подальшим поясненням кожного інгредієнта [22]. Практичне тестування показало, що застосунок в цілому коректно зчитує та подає пояснення інгредієнтів, однак українська мова не підтримується, що вказує на відсутність орієнтації на український ринок. Також можливі неточності розпізнавання, а розшифровка інгредієнтів це просто їх пояснення, без вказання користі або шкоди.

На основі характеристик застосунків, які було можливо протестувати, і які мають відповідне цільове призначення, було побудовано гістограму розподілу застосунків за роком публікації та способом введення і аналізу даних (рис. 1.3). Згідно з гістограмою, більшість відібраних застосунків опубліковано у 2025 році, і всі вони використовують MLLM для аналізу складу на основі фото упаковки.

Аналіз існуючих застосунків для розпізнавання та аналізу харчових продуктів показав, що багато із них працюють на основі штрих-кодів та даних із власних або відкритих баз. Це обмежує можливості користувачів, бо у випадку відсутності продукту в базі даних інформація або не надається взагалі, або подається неповною. Багато застосунків зосереджуються лише на калорійності чи загальній харчовій цінності. Окремі нові застосунки, які використовують

MLLM, дають можливість аналізувати склад продукту безпосередньо з етикетки. Однак практичне тестування показує, що такі рішення працюють нестабільно і не орієнтовані на українського споживача.

Рисунок 1.3 – Гістограма, яка відображає кількість відібраних застосунків за роком публікації та способом введення і аналізу даних

1.3 Методи та інструменти детектування та розпізнавання тексту на зображеннях

Комп'ютерний зір охоплює задачі розпізнавання облич [23, 24], підрахунку людей [25, 26], аналізу текстур [27] та розпізнавання текстової інформації [28, 29]. Аналіз складу харчових продуктів із фотографій упаковки належить саме до задач розпізнавання тексту.

Для того щоб мобільний застосунок міг прочитати склад продукту з фотографії упаковки, необхідно вирішити задачу розпізнавання тексту на зображенні. Сьогодні для цього існує два підходи. Перший – класичний OCR, де зображення спочатку обробляється, текстові ділянки виявляються і символи перетворюються на рядок тексту, який потім аналізується окремо. Другий –

MLLMs, які здатні одночасно «дивитися» на зображення і «розуміти» його вміст, виконуючи розпізнавання і смисловий аналіз за один крок. Обидва підходи мають свої переваги і можуть застосовуватись у розробці застосунку для аналізу складу харчових продуктів.

Розпізнавання тексту на фотографії упаковки починається з підготовки зображення. Перш ніж застосовувати будь-який OCR-рушій, зображення переводять у відтінки сірого, зменшують шум, підвищують контраст і виконують бінаризацію [30]. Також важливим кроком є виправлення нахилу та перспективи, що особливо актуально для круглих пляшок або пом’ятих упаковок. Для цих задач використовуються бібліотеки OpenCV [31], Pillow [32] та scikit-image [33]. В них вже реалізовано методи нормалізації зображень, наприклад, методи нормалізації на основі дескрипторів [34–39]. Після підготовки алгоритм визначає координати текстових блоків на зображенні, і лише потім виконується безпосереднє розпізнавання символів (рис. 1.4).

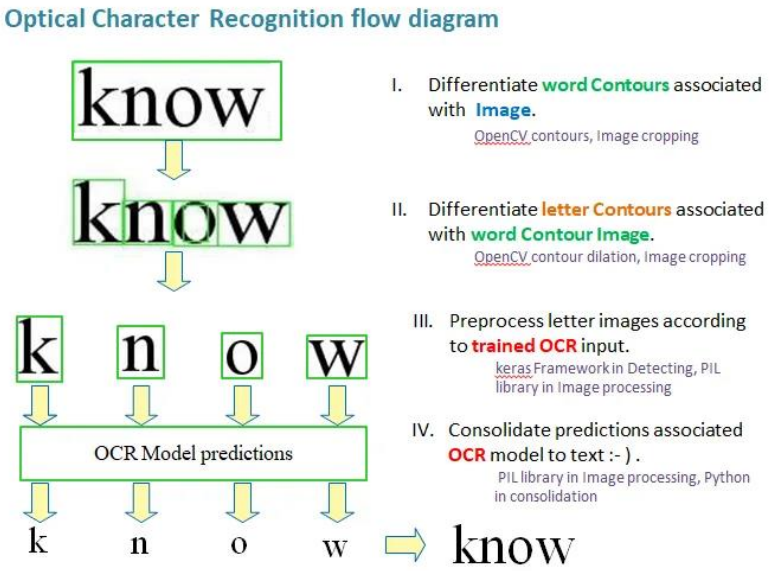


Рисунок 1.4 – Етапи обробки зображення тексту для OCR

Найвідомішим класичним OCR-рушієм є Tesseract [40], розроблений у Hewlett-Packard і зараз підтримуваний спільнотою за участю Google. Він безкоштовний, підтримує українську мову і добре працює зі сканованими

документами та чистими друкованими текстами. Проте на фотографіях реальних упаковок з відблисками, нерівним освітленням або дрібним шрифтом точність Tesseract помітно знижується без якісної попередньої обробки.

Для роботи з текстом у реальних сценах краще підходять спеціалізовані бібліотеки. EasyOCR [41] реалізована на PyTorch, підтримує понад 80 мов включно з кирилицею і вводиться у проєкт кількома рядками коду. Вона добре справляється з контрастними етикетками і є найпростішим варіантом для швидкого старту. PaddleOCR [42] є більш повноцінним фреймворком із окремими модулями для детекції та розпізнавання тексту, здатним працювати з нахиленим або дрібним текстом на вигнутих поверхнях. Бібліотека keras-ocr [43] цікава для експериментів, однак потребує більших зусиль на налаштування і рідше обирається як основний рушій у першій версії продукту.

Окрім відкритих бібліотек, існують хмарні OCR-сервіси, такі як Google Cloud Vision [44] і Amazon AWS Textract [45]. Вони забезпечують високу якість розпізнавання без необхідності налаштовувати власні моделі, проте потребують постійного інтернет-з’єднання та оплати за кожен запит.

MLLMs відрізняються від класичного OCR тим, що не просто розпізнають символи, а одразу розуміють зміст зображення. Це робить MLLM особливо привабливими для задачі аналізу складу продуктів.

Найпоширенішими хмарними MLLM є GPT від OpenAI, Gemini від Google та Claude від Anthropic. Моделі GPT підтримують зображення у запитах і здатні виконувати OCR, аналізувати таблиці та відповідати на запитання по зображенню [46]. Доступ здійснюється через API з оплатою за токени. Вартість GPT-5 становить 3,75 доларів за 1 мільйон вхідних і 15 доларів за 1 мільйон вихідних токенів, а легша версія mini коштує значно дешевше [47, 48]. Gemini від Google також із самого початку розроблялася як мультимодальна і підтримує текст, зображення та відео [49]. Легкі варіанти Flash/Flash-Lite мають дуже низьку ціну, приблизно десятки центів за 1 мільйон токенів, що робить їх зручними для масштабованих застосунків [50]. Claude від Anthropic добре справляється з документоподібними зображеннями, наприклад читає скани,

таблиці, витягує структуровану інформацію [51]. Для лінійки Sonnet вартість становить близько 3 доларів за 1 мільйон вхідних і 15 доларів за 1 мільйон вихідних токенів.

Альтернативою хмарним сервісам є відкриті MLLM, наприклад DeepSeek [52], які можна розгорнути на власному сервері. Перевага такого підходу полягає у повному контролі над даними, бо фото упаковок не передаються стороннім сервісам. Недоліком є необхідність у потужному залізі та часі на налаштування.

Порівняно з класичним OCR, MLLM є зручнішим рішенням для задач, де важливий не лише текст, а й його змістовна інтерпретація. Саме тому в рамках кваліфікаційної роботи доцільно розглядати MLLM як основний інструмент аналізу складу харчових продуктів.

1.4 Огляд програмного забезпечення для розробки мобільного застосунку

Сучасний мобільний застосунок складається з двох частин, а саме фронтенду і бекенду. Фронтенд – це те, що бачить користувач на екрані (інтерфейс, кнопки, форми). Бекенд працює на сервері і виконує основну логіку (обробляє запити, звертається до баз даних і повертає результат). Такий поділ дозволяє розвивати обидві частини незалежно і масштабувати систему при потребі. З погляду клієнтської частини існують два основні підходи: нативна розробка окремо під Android та iOS і кросплатформні фреймворки з єдиною кодовою базою.

Нативна розробка передбачає створення окремих застосунків для кожної платформи. Для Android використовується Android Studio з мовою Kotlin або Java – офіційне середовище з вбудованими інструментами для дизайну, емуляції та налагодження [53, 54]. iOS застосунки розробляються мовою Swift у середовищі Xcode [55, 56]. Нативний підхід забезпечує повний доступ до можливостей платформи, але вимагає підтримки двох окремих кодових баз.

Кросплатформні фреймворки дозволяють писати код один раз і запускати його на Android та iOS. React Native використовує JavaScript або TypeScript і збирає інтерфейс з нативних компонентів платформи, забезпечуючи доступ до камери, сховища та сенсорів через готові бібліотеки [57]. Flutter натомість використовує мову Dart і власний графічний рушій, що гарантує однаковий вигляд застосунку на всіх платформах, включно з вебом і десктопом [58]. Обидва підходи суттєво скорочують час розробки порівняно з нативним варіантом.

Бекенд надає API, через який мобільний клієнт передає зображення або текст, а сервер виконує обробку і повертає результат. Для його реалізації використовують Node.js із фреймворками Express або NestJS, і це зручно, якщо фронтенд теж написаний на JavaScript [59]. Альтернативою є Python із FastAPI, що особливо доречно, коли бекенд інтегрує бібліотеки машинного навчання на кшталт PyTorch, OpenCV або Hugging Face [60].

В результаті, для розробки мобільного застосунку доцільно використовувати архітектуру «мобільний клієнт і серверний API». На рівні фронтенду можна обрати нативну розробку (Kotlin/Android Studio або Swift/Xcode) або кросплатформний фреймворк (React Native чи Flutter). На рівні бекенду поширеними варіантами є Node.js або Python із FastAPI. Такий поділ забезпечує гнучкість, масштабованість і зручність подальшого розвитку системи.

1.5 Постановка задачі

Таким чином, аналіз поточного стану показав, що попри наявність різноманітних мобільних застосунків для оцінки корисності продуктів, усе ще бракує рішень, здатних працювати безпосередньо з фотографією складу на упаковці та надавати користувачу зрозумілу розшифровку інгредієнтів. Більшість існуючих сервісів або прив’язані до баз даних штрихкодів, або орієнтовані переважно на інші ринки та мови, а тому працюють нестабільно з

українськими товарами, не завжди коректно розпізнають інгредієнти та часто фокусуються лише на калорійності чи узагальненому рейтингу продукту. Персонал магазину також рідко може допомогти з розшифровкою інгредієнтів. Додатковою проблемою є складні умови зчитування складу з упаковки, наприклад дрібний шрифт, кілька мов одночасно тощо. Таким чином, питання самостійного аналізу складу продукту залишається невирішеним.

Класичний OCR потребує попередньої обробки зображення і додаткового налаштування, що може ускладнювати роботу з реальними фотографіями упаковок. Паралельно розвивається новий напрям, такий як MLLMs, які здатні працювати одночасно з текстом і зображеннями.

У зв’язку з цим постає актуальна задача створення мобільного застосунку, який на основі фотографії упаковки зможе автоматично отримати текст складу продукту та виконати його смисловий аналіз, орієнтований на потреби українських споживачів у багатомовному середовищі. При цьому основний акцент робиться на використанні MLLM.

Об’єктом роботи є процес розпізнавання та аналізу складу харчових продуктів на основі зображення упаковки.

Метою роботи є розроблення мобільного застосунку Android для розпізнавання та пояснення складу харчових продуктів з автоматизованим перекладом українською мовою.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– провести аналіз поточного стану проблеми інформованого вибору харчових продуктів;

– проаналізувати існуючі мобільні застосунки для сканування продуктів, оцінити їхню якість аналізу складу, виявити їхні переваги та обмеження щодо роботи з українськими товарами та українською мовою;

– оглянути сучасні підходи до отримання тексту з зображень, такі як пряме використання MLLMs;

– сформувати власний набір даних фотографій етикеток харчових продуктів, який охоплює україномовні та імпортні товари різних типів упаковок з еталонною розміткою для подальшого оцінювання якості роботи моделей;

– провести експериментальну оцінку якості зчитування тексту й коректності аналізу складу з різними MLLM і обрати найкращу модель;

– провести серію експериментів щодо якості обробки персональних запитів користувача щодо складу продукту;

– обґрунтувати вибір технологій для фронтенду та бекенду мобільного застосунку та визначити архітектуру клієнт-серверної взаємодії;

– описати предметну область та визначити структуру даних для представлення результатів аналізу у застосунку;

– спроєктувати інтерфейс мобільного застосунку, що забезпечить зручний сценарій роботи;

– реалізувати мобільний застосунок з використанням обраного технологічного стека та інтеграцією з MLLM через API.

2 РОЗРОБКА МЕТОДУ ДЛЯ РОЗПІЗНАВАННЯ ТА ПОЯСНЕННЯ СКЛАДУ ХАРЧОВИХ ПРОДУКТІВ З АВТОМАТИЗОВАНИМ ПЕРЕКЛАДОМ УКРАЇНСЬКОЮ МОВОЮ НА ОСНОВІ УПАКОВКИ

2.1 Постановка задачі експериментів

Метою експериментів є визначення найбільш ефективної MLLM для розпізнавання та пояснення складу харчових продуктів на основі фотографії упаковки з автоматизованим перекладом українською мовою. Під час випробувань особлива увага приділяється якості зчитування тексту зі складних зображень, коректності аналізу інгредієнтів, стабільності роботи моделей, швидкості отримання результату та вартості використання API. Обрана модель у подальшому буде інтегрована до мобільного застосунку Android, тому важливим є не лише рівень точності, а й практична придатність моделі для реального використання.

Для проведення експериментів потрібно сформувати набір даних фотографій упаковок харчових продуктів з відповідною розміткою. Наступним етапом є розробка критеріїв оцінювання якості роботи моделей. Для забезпечення однакових умов тестування необхідно розробити єдиний запит для всіх обраних для аналізу моделей. Після формування набору даних та критеріїв оцінювання буде виконано відбір MLLMs для тестування. Для кожної моделі планується провести серію експериментів із однаковими вхідними даними та однаковими параметрами запиту. Отримані результати будуть оцінюватися експертним способом за визначеними критеріями. Оскільки розширення набору даних у майбутньому може значно збільшити обсяг оцінювання, додатково планується розглянути підхід автоматизованого оцінювання якості за допомогою великої мовної моделі. Важливим доповненням до базового функціоналу є можливість виконання додаткового аналізу продукту відповідно до потреб користувача. Для перевірки працездатності цієї функції планується окремо сформувати набір даних та визначити критерії оцінювання якості відповіді

моделі на такі запити.

За результатами проведених експериментів буде обрано MLLM, яка найкраще підходить для інтеграції у мобільний застосунок та забезпечує оптимальне співвідношення між якістю аналізу, швидкістю роботи, стабільністю та вартістю використання.

2.2 Огляд наборів даних

Для проведення експериментів необхідним етапом є пошук наборів даних, що містять різноманітні фотографії упаковок харчових продуктів та текст складу інгредієнтів з поясненнями.

На першому етапі було розглянуто популярні платформи для пошуку наборів даних, серед яких Kaggle [61], Google Dataset Search [62], Mendeley Data [63], Roboflow Universe [64] та AWS Open Data Registry [65]. Дані ресурси містять велику кількість наборів даних для задач комп'ютерного зору, розпізнавання тексту та аналізу зображень упаковок продуктів.

Під час аналізу було знайдено декілька наборів даних, які частково відповідають поставленій задачі. Зокрема, набір даних Ingredients Image from Food Label [66] містить англомовні фотографії, набір OpenFoodFacts Ingredient Detection [67] містить велику кількість фотографій продуктів різних мов та текстові списки інгредієнтів. Також було розглянуто декілька наборів даних з Roboflow Universe. До них належать Food Packaging Recognition [68], Comp 4102 Project [69], Ingredients Label Detection [70], Ingredients [71], Ingredient List Detection [72] та Ingredient Label [73]. Дані набори можуть бути корисними для пошуку імпортних фотографій етикеток, однак більшість із них орієнтовані переважно на задачі виявлення об'єктів або локалізації текстових областей.

Набір FoodRepo [74] містить фотографії упаковок продуктів з різних країн та може бути корисним для аналізу багатомовних етикеток. Також було знайдено набір Iranian Nutritional Fact Label [75], що містить іранські етикетки харчових

продуктів, та Korean Product Labels Image Dataset [76], який містить корейські етикетки. Крім цього, було знайдено набір даних Food Labels Dataset [77], який за описом відповідає поставленій задачі, однак на момент проведення експериментів доступ до його повного завантаження був відсутній.

У результаті аналізу існуючих наборів даних було встановлено, що жоден із доступних наборів не забезпечує необхідного поєднання характеристик для проведення експериментів. Зокрема, відсутні набори даних, які одночасно містять фотографії складу продуктів, багатомовні етикетки, відповідну розмітку інгредієнтів та детальний аналіз компонентів українською мовою. У зв'язку з цим було прийнято рішення створити власний набір даних етикеток харчових продуктів.

2.3 Створення власного набору даних етикеток

Фотографії упаковок для набору даних були отримані двома способами. Частина зображень була зроблена самостійно під час фотографування продуктів у магазинах, а інша частина була взята з відкритих джерел, розглянутих у попередньому підрозділі. Під час формування набору даних особлива увага приділялась різноманітності типів упаковок, мов написання складу та умов фотографування. Приклади фотографій етикеток наведено на рисунку 2.1.

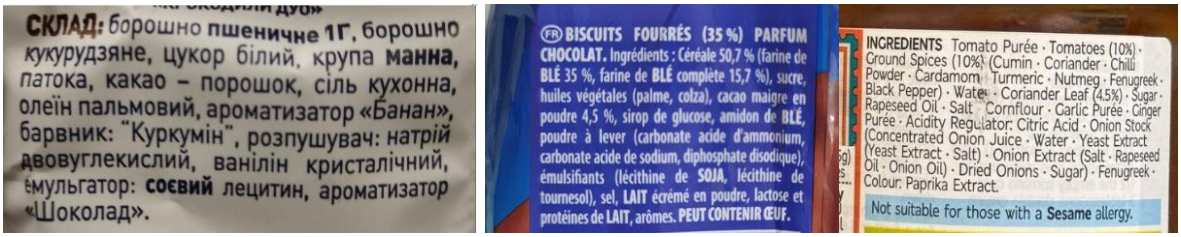


Рисунок 2.1 – Приклади зображень з власного набору даних

У межах роботи одиницею набору даних є один харчовий продукт. Для кожного продукту передбачається збереження від 1 до 4 зображень упаковки.

Кількість фотографій залежить від форми упаковки та способу розміщення тексту складу. Під час експериментального збору даних було встановлено, що чотирьох фотографій достатньо для повного зчитування складу будь-якого продукту, який був проаналізований у межах роботи. Структура набору даних організована у вигляді окремих директорій для кожного продукту (рис. 2.2).

```
dataset/
├── README.md              # Опис датасету коротко
├── metadata.json          # Загальні характеристики (кількість, рік створення, тощо)
│
├── products/              # Всі продукти
│   ├── UA_RECT_001/       # Код: [мова]_[форма]_[номер]
│   │   ├── images/
│   │   │   ├── 1.jpg      # Головна сторона
│   │   │   ├── 2.jpg      # Зворотня сторона
│   │   │   └── ...        # До 4 фото
│   │   └── 001.json       # Еталонна відповідь
│   │
│   ├── UA_ROUND_002/
│   │   ├── images/
│   │   └── 002.json
│   │
│   ├── IMPORT_RECT_003/
│   │   ├── images/
│   │   └── 003.json
│   ...
```

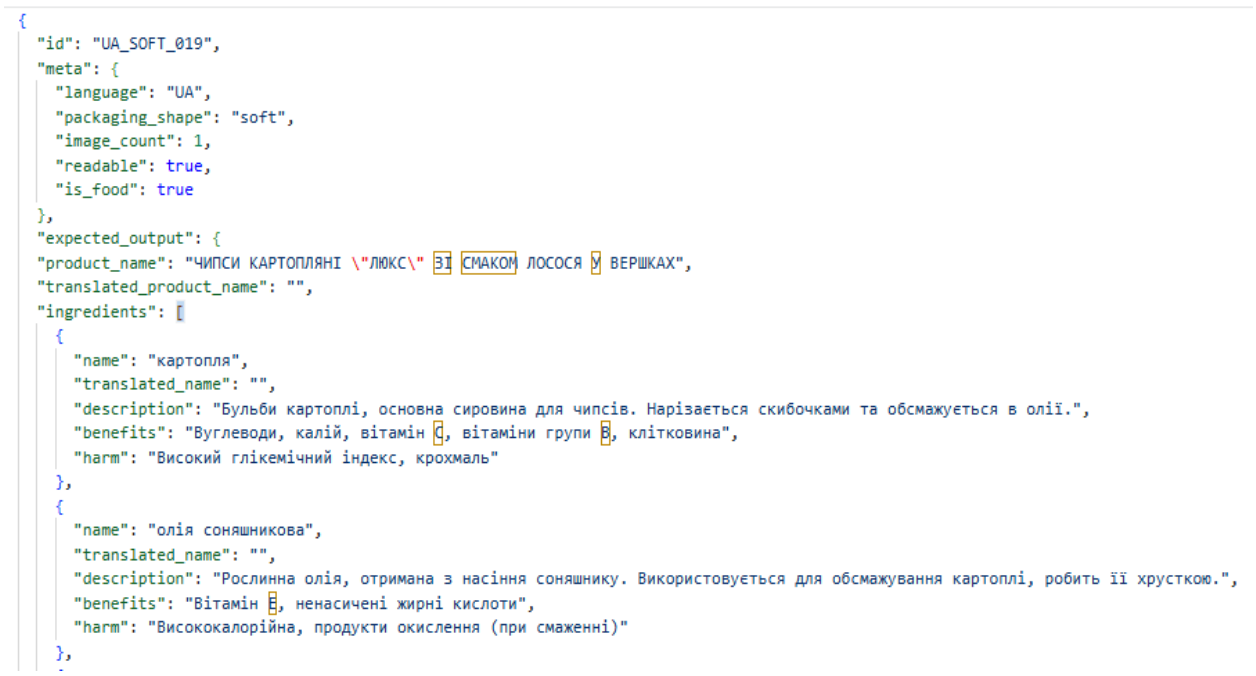
Рисунок 2.2 – Структура власного набору даних етикеток харчових продуктів

Для ідентифікації продуктів використовується спеціальний код у форматі [МОВА]_[ФОРМА]_[НОМЕР]. Перша частина коду визначає основну мову етикетки. Значення UA використовується для україномовних етикеток, а IMPORT для етикеток іноземними мовами. Друга частина коду визначає форму упаковки. Значення RECT використовується для прямокутних або квадратних упаковок, ROUND для круглих упаковок, а SOFT для м’яких. Остання частина коду є порядковим номером продукту в наборі даних.

Для кожного продукту окремо зберігаються фотографії упаковки та файл розмітки у форматі JSON. Файл розмітки містить еталонну відповідь, яка використовується під час оцінювання якості роботи моделей (рис. 2.3).

JSON-файл містить ідентифікатор продукту, метаінформацію та очікуваний результат аналізу. До метаінформації належать мова етикетки, форма упаковки, кількість фотографій, ознака читабельності тексту та інформація про

те, чи є об’єкт харчовим продуктом. Основна частина файлу містить назву продукту, переклад назви українською мовою за потреби та список інгредієнтів.



```
{
  "id": "UA_SOFT_019",
  "meta": {
    "language": "UA",
    "packaging_shape": "soft",
    "image_count": 1,
    "readable": true,
    "is_food": true
  },
  "expected_output": {
  "product_name": "ЧИПСИ КАРТОПЛЯНІ \"ЛЮКС\" ЗІ СМАКОМ ЛОСОСЯ У ВЕРШКАХ",
  "translated_product_name": "",
  "ingredients": [
    {
      "name": "картопля",
      "translated_name": "",
      "description": "Бульби картоплі, основна сировина для чипсів. Нарізається скибочками та обсмажується в олії.",
      "benefits": "Вуглеводи, калій, вітамін C, вітаміни групи B, клітковина",
      "harm": "Високий глікемічний індекс, крохмаль"
    },
    {
      "name": "олія соняшникова",
      "translated_name": "",
      "description": "Рослинна олія, отримана з насіння соняшнику. Використовується для обсмажування картоплі, робить її хрусткою.",
      "benefits": "Вітамін E, ненасичені жирні кислоти",
      "harm": "Висококалорійна, продукти окислення (при смаженні)"
    },
```


Рисунок 2.3 – Приклад частини JSON-файлу розмітки продукту

Для кожного інгредієнта передбачено окремий об’єкт із декількома полями. Зберігається оригінальна назва інгредієнта, його переклад українською мовою, текстовий опис з поясненням компонента простими словами, а також окремо перераховуються можливі корисні та шкідливі властивості та речовини.

Сформований набір даних враховує особливості реальних фотографій упаковок харчових продуктів, включаючи різні мови, типи упаковок, умови освітлення та складність зчитування тексту. Всього було розмічено 20 продуктів.

2.4 Розробка критерію оцінювання якості

Для проведення експериментів було розроблено систему оцінювання якості роботи MLLM, яка комплексно оцінює якість зчитування тексту, коректність аналізу інгредієнтів, правильність перекладу та відповідність

структури відповіді заданому формату. Відповідь моделі порівнюється з еталонною розміткою з власного набору даних за визначеними критеріями, після чого для кожного блоку обчислюється окрема оцінка та формується підсумковий результат.

Режим А використовується для випадків, коли зображення містить склад харчового продукту та текст на упаковці є читабельним. Для цього режиму застосовуються всі блоки критеріїв оцінювання. Фінальна оцінка формується як зважена сума оцінок усіх блоків.

Режим Б використовується у випадках, коли зображення не містить складу харчового продукту, текст на упаковці є нечитабельним або надане фото є довільним і не стосується аналізу складу. У такому випадку оцінюється лише коректність структури відповіді та здатність моделі правильно обробляти нестандартні ситуації.

Для режиму А фінальна оцінка обчислюється за формулою:

$$Score = \sum_{i=1}^{7} \left(\frac{S_i}{M_i} \cdot W_i \right) \cdot 10, \quad (2.1)$$

де i – номер блоку;

S_i – сума балів за блок;

M_i – максимально можлива кількість балів для блоку;

W_i – вага відповідного блоку.

Для режиму Б використовується окрема формула:

$$Score = \frac{S_1}{3} \cdot 10, \quad (2.2)$$

де S_1 – сума балів першого блоку.

Під час оцінювання критерії, які не застосовуються до конкретного продукту, позначаються значенням *N/A* та виключаються з розрахунку максимальної кількості балів відповідного блоку.

Система оцінювання складається із семи блоків критеріїв. Перший блок призначений для перевірки структури відповіді та оброблення нестандартних ситуацій (табл. Б.1). До нього входить перевірка валідності JSON, правильності класифікації типу продукту та коректності оброблення edge cases.

Другий блок використовується для оцінювання назви продукту (табл. Б.2). Оцінюється правильність визначення назви, а також якість перекладу українською мовою для імпортних товарів. Якщо назва продукту відсутня в еталонній розмітці, перевіряється здатність моделі коректно залишити відповідне поле порожнім.

Третій блок оцінює повноту списку інгредієнтів (табл. Б.3). Це найважливіший блок, бо відсутність навіть одного інгредієнта, може суттєво спотворити якість аналізу. Під час оцінювання враховується кількість знайдених інгредієнтів, наявність вигаданих компонентів та збереження порядку інгредієнтів відповідно до упаковки продукту.

Четвертий блок використовується для перевірки коректності назв інгредієнтів (табл. Б.4). Оцінюється точність відтворення оригінальних назв, а також якість перекладу для імпортних продуктів.

П’ятий блок оцінює якість поля опису, яке містить пояснення інгредієнтів простими словами (табл. Б.5). Під час оцінювання враховується фактична коректність опису та зрозумілість формулювань для людини без спеціальних знань у галузі харчування чи медицини.

Шостий блок призначений для оцінювання поля користі, яке містить інформацію про ключові поживні властивості або корисні речовини інгредієнта (табл. А.6). Оцінюється відповідність тверджень реальним науковим даним та відсутність вигаданих або перебільшених тверджень.

Сьомий блок використовується для оцінювання поля шкоди, що містить інформацію про шкідливі для здоров’я речовини або властивості інгредієнту (табл. Б.7). Під час оцінювання враховується фактична коректність та повнота інформації.

Для забезпечення комплексного оцінювання кожному блоку було

призначено окрему вагу залежно від його важливості для поставленої задачі. Найбільшу вагу отримав блок повноти інгредієнтів, оскільки саме правильне зчитування складу є основною задачею системи. Зведену таблицю ваг та максимальних внесків кожного блоку у фінальну оцінку наведено у таблиці 2.1.

Таблиця 2.1 – Зведена таблиця ваг

Блок	Що оцінюємо	Максимально балів у блоці	Вага	Максимальний внесок у фінал
1	Структура + edge cases	2 або 3	0,15	1,50
2	Назва продукту	1, або 2, або 3	0,10	1,00
3	Повнота інгредієнтів	5	0,30	3,00
4	Назви інгредієнтів	2 або 4	0,15	1,50
5	Опис інгредієнта	3	0,15	1,50
6	Користь інгредієнта	2	0,075	0,75
7	Шкода інгредієнта	2	0,075	0,75
	Сума		1,00	10,00

2.5 Розробка запиту для моделей

Для проведення коректного порівняння MLLMs необхідно використовувати єдиний запит для всіх експериментів (рис. 2.4). Це дозволяє забезпечити однакові умови тестування.

Під час створення запиту враховувались особливості фотографій упаковок харчових продуктів. Зокрема, було передбачено можливість передавання від одного до чотирьох зображень одного продукту. Також враховувались ситуації з частково нечітким текстом, складною формою упаковки та багатомовними етикетками.

Розроблений запит складається з декількох логічних частин. У першій частині моделі надається загальна задача щодо аналізу складу харчового

продукту на основі фотографій упаковки. Далі задаються обмеження та правила оброблення нестандартних ситуацій. Зокрема, модель повинна окремо обробляти випадки, коли на фотографії зображено нехарчовий продукт або коли текст складу неможливо прочитати. У таких ситуаціях модель повинна повертати лише JSON-об’єкт з полем error без будь-якого додаткового тексту.

```
Проаналізуй склад харчового продукту з наданих фотографій упаковки (може бути від 1 до 4 фото). Кілька
фото можуть бути надані через форму упаковки або тому, що назва і склад розміщені на різних боках, тощо.
Якщо на фото зображено не харчовий продукт для людей (наприклад, корм для тварин, косметика, побутова
хімія, ліки, тощо), то поверни JSON з полем "error": "Це не харчовий продукт для людей" і більше нічого.
Якщо склад продукту на фото неможливо прочитати взагалі, то поверни JSON з полем "error": "Склад не
вдалося прочитати" і більше нічого.
Аналіз надавай виключно українською мовою, навіть якщо текст на етикетці іншою мовою. Якщо якийсь текст
на фото нечіткий або частково прихований, то аналізуй лише те, що вдалося прочитати, не вигадуй
інгредієнтів, яких не бачиш. Аналізуй лише склад (інгредієнти). Зберігай оригінальний порядок
інгредієнтів, як на упаковці. Виділяй усі інгредієнти без винятку.
Результат надай строго у форматі JSON, без будь-якого додаткового тексту до або після, за такою
структурою:
{
"product_name": "Назва продукту з фото (як на упаковці). Якщо назву не видно, то залиши порожній рядок,
сам не вигадуй",
"translated_product_name": "Переклад назви продукту українською. Якщо продукт український, то залиши
порожній рядок",
"ingredients": [
{
"name": "Назва інгредієнта (як на упаковці)",
"translated_name": "Переклад назви інгредієнта українською. Якщо продукт український, то залиши порожній
рядок",
"description": "Поясни що це таке, звідки береться, для чого додається до продукту, яку функцію виконує.
Пиши просто і зрозуміло, як для людини без медичної освіти, якщо використовуєшь якісь терміни, пиши їх
пояснення.",
"benefits": "Користь для здоров'я. Коротко вкажи ключові поживні властивості або корисні речовини,
наприклад “клітковина, залізо, вітамін С”. Не пояснюй механізм впливу на організм цих корисних
елементів, просто переліч їх. Якщо реальної користі немає, то написати: 'Для здоров'я користі не має'",
"harm": "Шкода для здоров'я. Коротко напиши шкідливі для здоров’я речовини або властивості цього
інгредієнту, наприклад “кофеїн, глютен, висококалорійний”. Не пояснюй механізм впливу цих шкідливих
компонентів на організм, не вказуй тут хвороби які ці компоненти можуть викликати, просто переліч. Якщо
шкода не виявлена при помірному вживанні, то написати: 'При помірному вживанні шкоди не виявлено'"
}
]
}
```

Рисунок 2.4 – Розроблений запит для MLLM

Окремий блок запиту визначає правила роботи зі складом продукту. Моделі забороняється вигадувати інгредієнти, яких немає на фотографії, а також змінювати порядок компонентів відносно упаковки. Усі результати аналізу повинні формуватись виключно українською мовою незалежно від мови оригінальної етикетки.

Наступна частина запиту визначає структуру вихідного JSON-об’єкта. Для кожного продукту модель повинна повернути назву продукту, переклад назви українською мовою за потреби та список усіх інгредієнтів. Для кожного

інгредієнта додатково формується опис, у якому простими словами пояснюється його походження, призначення та функція у продукті. Також окремо формуються поля користі та шкоди, які містять коротку інформацію про корисні та потенційно шкідливі властивості інгредієнта.

Розроблений запит використовувався без змін для всіх MLLMs, що брали участь у тестуванні. Це дозволило забезпечити коректність порівняння результатів експериментів та зменшити вплив сторонніх факторів на підсумкове оцінювання.

2.6 Відбір моделей для тестування

На початковому етапі роботи було проведено аналіз сучасних MLLMs, здатних працювати із зображеннями та виконувати завдання розпізнавання тексту. Для цього було розглянуто спеціалізовані рейтинги, аналітичні платформи та огляди моделей, присвячені задачам комп’ютерного зору, OCR та мультимодального аналізу [78–82].

Під час аналізу особлива увага приділялась якості розпізнавання тексту із фотографій, здатності працювати зі складними зображеннями, швидкості формування відповіді, стабільності результатів та можливості використання через API у мобільному застосунку.

Також було знайдено приклад схожої задачі [83]. У межах цього дослідження для задачі отримання інгредієнтів із фотографій упаковок було протестовано різні MLLMs, серед яких найкращі результати показали моделі сімейств GPT [84], Claude [85] та Gemini [86]. Цей досвід було враховано під час формування переліку моделей-кандидатів для власних експериментів.

На етапі попереднього відбору було протестовано модель DeepSeek R1. У процесі роботи було виявлено суттєві проблеми зі швидкістю оброблення зображень. Крім цього, у деяких випадках виникали проблеми із завантаженням зображень до моделі (рис. 2.5). Таким чином, її було виключено з аналізу.

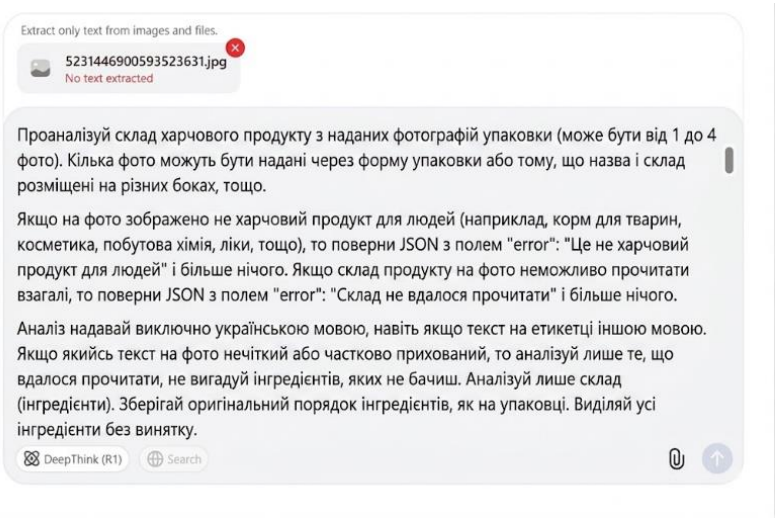


Рисунок 2.5 – Збій при завантаженні фото в DeepSeek R1

Також було протестовано модель Qwen3.6-Plus. Під час експериментів спостерігались помилки при зчитуванні інгредієнтів, нестабільність результатів та порівняно тривалий час формування відповіді. У деяких випадках модель формувала JSON із помилками у структурі та змісті аналізу (рис. 2.6). Також далі ця модель не буде розглядатись.

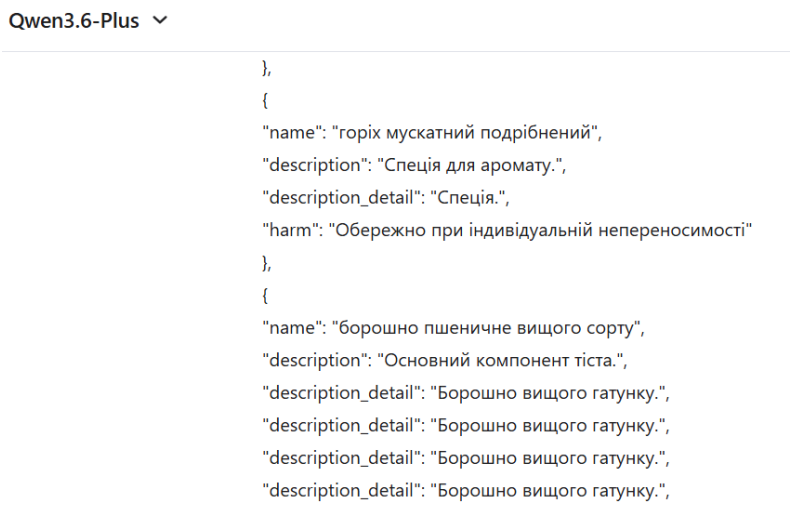


Рисунок 2.6 – Сформований JSON з помилками від Qwen3.6-Plus

Модель GPT-5.3 Instant продемонструвала високу швидкість роботи та стабільну структуру JSON-відповідей. Інгредієнти здебільшого зчитувались коректно та у повному обсязі. Водночас у змістовних полях аналізу

спостерігались неточності. Також модель не завжди стабільно відтворювала назву продукту і змінювала порядок інгредієнтів.

Claude Sonnet 4.6 показала дуже високий рівень зчитування тексту та коректне виділення інгредієнтів. Модель формувала досить деталізований аналіз. Але були присутні окремі логічні неточності у поясненнях.

Gemini 2.5 Flash продемонструвала хорошу якість аналізу інгредієнтів, однак виявилась нестабільною при повторних запусках. У різних спробах змінювались як назви інгредієнтів, так і повнота списку складу.

Gemini 2.5 Pro показала один із найкращих результатів серед протестованих моделей. Модель забезпечувала коректне зчитування інгредієнтів, достатньо якісний аналіз та працювала досить стабільно. Помилки траплялись рідко та не мали критичного впливу на результат.

Gemini 3 Flash Preview продемонструвала хорошу швидкість роботи, однак якість змістовного аналізу виявилась нижчою порівняно з іншими моделями. Частина пояснень була надто поверховою, а між різними запусками спостерігались значні відмінності у структурі та змісті відповіді.

Gemini 3.1 Flash Lite Preview працювала дуже швидко, однак якість аналізу виявилась недостатньою для поставленої задачі. У відповідях були присутні помилки у поясненнях інгредієнтів, а також спостерігались випадки втрати окремих компонентів складу.

Gemini 3.1 Pro Preview показала один із найкращих загальних результатів. Модель стабільно зчитувала склад продукту, зберігала порядок інгредієнтів та формувала достатньо якісний аналіз українською мовою. Відповіді між повторними запусками були подібними, а більшість змін стосувались лише формулювання тексту.

Також було протестовано моделі GPT-5.4 та GPT-5.2. Модель GPT-5.4 показала якісний результат аналізу, однак суттєвих переваг над GPT-5.2 виявлено не було. При цьому GPT-5.4 працювала повільніше та вимагала більших витрат на використання API. З цієї причини для подальших експериментів було обрано GPT-5.2.

GPT-5.4 mini працювала швидко та стабільно, однак якість аналізу виявилась занадто поверховою для задачі детального пояснення інгредієнтів.

У результаті проведеного аналізу для подальших експериментальнтів було обрано три MLLMs, а саме GPT-5.2, Gemini 2.5 Pro та Gemini 3.1 Pro Preview. Модель Claude Sonnet 4.6 була виключена з фінального переліку через найвищу вартість використання API.

У таблиці 2.2 наведено фінальний перелік моделей-кандидатів для порівняльного аналізу та інформацію щодо вартості використання API.

Таблиця 2.2 – Моделі-кандидати для порівняльного аналізу

Модель	Провайдер	Вхід ($/1M токенів)	Вихід ($/1M токенів)
GPT-5.2	OpenAI	$1,75	$14,00
Gemini 3.1 Pro Preview	Google	$2,00	$12,00
Gemini 2.5 Pro	Google	$1,25	$10,00

2.7 Експеременти з моделями-кандидатами

2.7.1 Еспертне оцінювання якості по критеріям

За результатами експериментів та експертного оцінювання найвищу середню оцінку отримала модель Gemini 2.5 Pro (табл. В.1). Середній підсумковий бал моделі склав 9,220 із 10 можливих. Модель продемонструвала високу якість розпізнавання тексту, коректне збереження структури JSON та достатньо якісний аналіз інгредієнтів українською мовою. Модель завжди правильно зчитувала всі інгредієнти та не вигадувала додаткових компонентів. Також модель коректно обробляє нечитабельні зображення або нехарчові продукти. Водночас середній час формування відповіді для Gemini 2.5 Pro становив приблизно 29,3 с, що є досить тривалим для практичного використання у магазині.

Модель Gemini 3.1 Pro Preview також показала високий рівень якості

(табл. В.2). Середній підсумковий бал склав 9,193. Модель формувала достатньо якісний та зрозумілий аналіз інгредієнтів, добре працювала з перекладом та коректно обробляла більшість тестових прикладів. Проте основною проблемою цієї моделі стала швидкість роботи. Середній час формування відповіді становив приблизно 37,2 с. У деяких випадках оброблення запиту тривало понад хвилину. Також під час тестування часто спостерігались випадки зациклення генерації відповіді.

Модель GPT-5.2 показала найвищу швидкість роботи серед усіх протестованих моделей (табл. В.3). Середній час формування відповіді становив приблизно 18,3 с. Модель стабільно формувала валідний JSON та добре обробляла нестандартні ситуації. Проте якість змістовного аналізу виявилась нижчою порівняно з моделями Gemini. У деяких випадках модель пропускала окремі інгредієнти або порушувала їхній порядок відносно упаковки. Також спостерігались поверхові або неточні пояснення для окремих компонентів складу. Середній підсумковий бал якості роботи моделі склав 8,792.

Отримані результати показали, що Gemini 2.5 Pro продемонструвала найкращий баланс між якістю аналізу та швидкістю роботи, хоча час формування відповіді все ще залишається досить великим для комфортного використання у реальних умовах.

2.7.2 Вибір LLM для автоматизування оцінювання якості відповідей Gemini 2.5 Pro

Підхід LLM-as-a-Judge передбачає використання великої мовної моделі як автоматизованого експерта для оцінювання якості відповідей людини, або інших моделей. У такому підході модель-оцінювач отримує еталонну відповідь, згенеровану відповідь тестованої моделі та набір критеріїв оцінювання, після чого формує підсумкову оцінку [87, 88]. Основною перевагою цього підходу є можливість автоматизації складного експертного оцінювання.

Використання автоматизованого оцінювання є доцільним у межах даної роботи, оскільки власний набір даних планується поступово розширювати. Проведення повного експертного оцінювання для великої кількості тестових прикладів потребує значних часових витрат, тому виникає необхідність автоматизації цього процесу із збереженням максимальної наближеності до людської експертної оцінки.

Для автоматизованого оцінювання було розглянуто три моделі: GPT-5.4, Gemini 2.5 Pro та Gemini 3.1 Pro Preview. Для всіх моделей було розроблено спеціалізований запит (додаток Г), у якому модель-оцінювач отримує еталонну відповідь, відповідь згенеровану Gemini 2.5 Pro, зображення упаковки продукту, а також детальну систему критеріїв оцінювання. Запит містить формули розрахунку підсумкової оцінки, правила оброблення складених інгредієнтів, нестандартних ситуацій та вимоги до формату JSON-відповіді.

У результаті проведених експериментів встановлено, що модель GPT-5.4 найбільш точно відтворює результати експертного оцінювання (табл. Г.1). Моделі Gemini 2.5 Pro та Gemini 3.1 Pro Preview мали тенденцію до завищення підсумкових оцінок, особливо для частково коректних відповідей (табл. Г.2, Г.3). Крім того, GPT-5.4 показала більш стабільне дотримання правил оцінювання та коректніше врахування неповних або неоднозначних відповідей моделей.

2.7.3 Оптимізація часу роботи Gemini 2.5 Pro

Під час попередніх експериментів було встановлено, що моделі Google Gemini 2.5 Pro та Gemini 3.1 Pro Preview демонструють високу якість аналізу, проте характеризуються тривалим часом обробки запитів. Особливо це критично для сценарію використання мобільного застосунку в магазині, де користувач очікує швидко отримати результат. Тому було проведено додаткові експерименти для оптимізації часу.

З документації моделей було встановлено, що для Gemini 2.5 Pro

доступний параметр thinking_budget, а для Gemini 3.1 Pro Preview – параметр thinking_level. Дані параметри відповідають за обсяг внутрішніх міркувань моделі під час генерації відповіді. Чим вищі їх значення, тим більше часу модель витрачає на аналіз запиту, що потенційно може підвищувати якість відповіді, але збільшує час обробки. Також використовувався параметр temperature, який визначає рівень випадковості генерації, де нижчі значення роблять відповіді більш стабільними та детермінованими, а вищі більш різноманітними. Згідно документації для Gemini 3 рекомендовано використовувати значення temperature=1,0, оскільки це зменшує ризик зациклювання моделі. Для Gemini 2.5 Pro допускається використання нижчих значень, тому в роботі було обрано temperature=0,3 для підвищення стабільності відповідей.

Для Gemini 3.1 Pro Preview були проведені експерименти зі значеннями thinking_level=low та та thinking_level=medium. Це дозволило скоротити час генерації та усунути зациклювання, однак якість аналізу погіршилась і модель почала частіше пропускати інгредієнти та менш точно формувати описові поля. Через це використання Gemini 3.1 Pro Preview було визнано недоцільним.

Подальші експерименти проводились із моделлю Gemini 2.5 Pro та параметром thinking_budget. Було протестовано значення 128, 256 та 512. Результати оцінювання наведені у таблицях Д.1–Д.3. Встановлено, що при thinking_budget=512 середня оцінка якості становила 9,230 бала при середньому часі обробки 18,8 с, що майже відповідає результату моделі з параметрами за замовчуванням (9,220 бала при 29,3 с). Таким чином, вдалося скоротити час роботи приблизно на 36 % без втрати якості аналізу.

При thinking_budget=256 середня оцінка знизилась до 9,090 бала, хоча час обробки скоротився до 16,6 с. Для thinking_budget=128 середня оцінка становила 8,906 бала при середньому часі 16,4 с, що свідчить про суттєвішу втрату якості аналізу. Отже, оптимальним варіантом було обрано використання Gemini 2.5 Pro з параметрами thinking_budget=512 та temperature=0,3, оскільки така конфігурація забезпечує найкращий баланс між якістю аналізу та швидкістю роботи системи.

2.7.4 Оцінювання стабільності

Оскільки MLLMs мають недетермінований характер генерації, важливо було оцінити стабільність роботи обраної моделі Gemini 2.5 Pro. Для цього було проведено експеримент, у межах якого три різні продукти запускались по три рази з однаковими параметрами генерації. Метою експерименту було перевірити, наскільки суттєво можуть відрізнятись результати аналізу одного й того ж продукту між різними запусками моделі.

Для продукту UA_RECT_001 результати виявились практично ідентичними. Усі три JSON-відповіді мали однакову структуру, однаковий список інгредієнтів та правильний порядок їх розташування. Відмінності стосувались лише стилістики текстових описів у полях опису, користі та шкоди, бо модель використовувала різні синоніми, змінювала деталізацію пояснень або формулювання користі та шкоди окремих інгредієнтів. При цьому зміст інформації залишався незмінним. Після автоматичного оцінювання всі три результати отримали однакову фінальну оцінку 9,250 бала, що свідчить про високу стабільність моделі.

Для продукту UA_SOFT_004 було виявлено більш помітні розбіжності між генераціями. Вони стосувались насамперед способу обробки вкладених інгредієнтів. У двох запусках модель повністю розгортала складені інгредієнти у плаский список, через що в JSON з’являлись дублікати окремих компонентів. У третьому запуску модель намагалася зберегти вкладену структуру, однак допустила помилку в ієрархії інгредієнтів, винісши частину компонентів зі складу вкладеного інгредієнта на основний рівень. Незважаючи на це, усі генерації залишались загалом коректними та придатними для подальшої обробки. Оцінки склали 8,875, 9,625 та 8,875 бала відповідно. Це демонструє, що найбільша варіативність виникає саме у випадках складних багаторівневих складів продукту. Але при цьому всі інгредієнти було зчитано та проаналізовано.

Для продукту IMPORT_RECT_012 результати знову виявились дуже стабільними. Усі три генерації коректно витягнули однаковий список

інгредієнтів, зберегли правильний порядок та правильно обробили відсоткові значення у складі. Відмінності спостерігались лише у формулюваннях текстових пояснень та інтерпретаціях користі окремих інгредієнтів. Оцінки трьох запусків становили 9,625, 10,000 та 9,250 бала.

Отримані результати свідчать, що Gemini 2.5 Pro працює достатньо стабільно для задачі аналізу складу харчових продуктів. У більшості випадків модель формує майже ідентичні JSON-відповіді, а виявлені розбіжності переважно стосуються стилістики текстових описів або способу представлення вкладених інгредієнтів. При цьому навіть у складних випадках фінальні оцінки залишаються високими та не демонструють критичних просідань якості.

2.7.5 Висновки щодо обрання моделі

За результатами проведених експериментів для використання у мобільному застосунку було обрано модель Gemini 2.5 Pro. Вона продемонструвала найкращий баланс між якістю аналізу, стабільністю роботи та швидкістю обробки запитів. У порівнянні з GPT-5.2 модель Gemini 2.5 Pro забезпечує значно точніше розпізнавання складу продуктів, не пропускає інгредієнти та краще працює з описовими полями. У свою чергу Gemini 3.1 Pro Preview хоча й показала близький рівень якості, проте характеризувалась значно більшим часом генерації та схильністю до зациклювання.

Додаткові експерименти з параметром thinking_budget дозволили суттєво скоротити середній час роботи Gemini 2.5 Pro без помітної втрати якості аналізу. Також було встановлено, що модель демонструє достатньо стабільні результати між повторними запусками, а виявлені відмінності переважно стосуються стилістики текстових пояснень або способу представлення вкладених інгредієнтів.

Отже, для подальшої реалізації системи було обрано Gemini 2.5 Pro з параметрами thinking_budget=512 та temperature=0,3, оскільки така конфігурація

забезпечує найкращий компроміс між точністю аналізу, стабільністю та швидкодією системи.

2.8 Оцінювання якості оброблення персонального запиту користувача

2.8.1 Створення набору даних

Для експериментальної перевірки якості оброблення персональних запитів користувача було сформовано окремий власний набір даних. Його основною метою є перевірка здатності MLLM аналізувати склад продукту з урахуванням індивідуальних потреб користувача, наприклад наявності алергенів, небажаних компонентів або обмежень, пов’язаних зі станом здоров’я. Структуру набору даних наведено на рисунку 2.7.

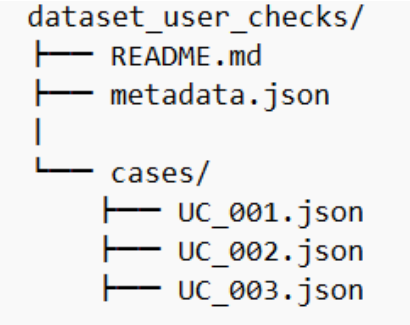


Рисунок 2.7 – Структуру власного набору даних для персонального запиту користувача

Один елемент набору даних називається кейс (case), тобто випадок, а не «продукт», оскільки один і той самий продукт може використовуватись у багатьох різних сценаріях перевірки. Наприклад, для одного й того ж продукту можуть бути створені окремі кейси для перевірки на наявність лактози, оцінки безпечності під час вагітності або аналізу нерелевантного запиту.

Загалом було сформовано 12 кейсів різних типів, що дозволило перевірити роботу системи у різних сценаріях використання, а саме від простих перевірок окремих компонентів до комбінованих та частково нерелевантних запитів.

Кожен кейс зберігається у форматі JSON. Приклад структури одного елемента набору даних наведено на рисунку 2.8.

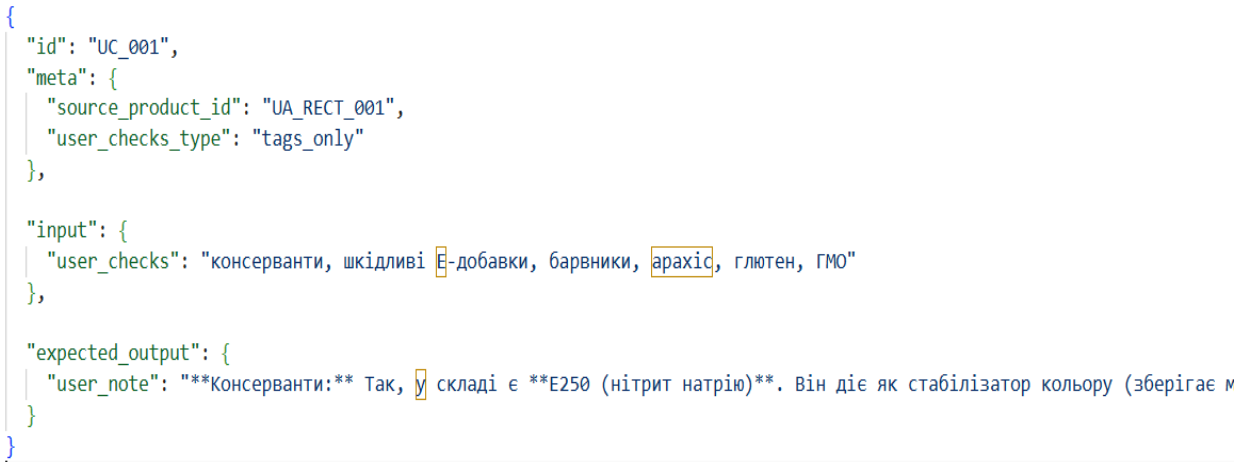


Рисунок 2.8 – Приклад структури кейсу набору даних

Поле id містить унікальний ідентифікатор кейсу. Блок meta використовується для службової інформації. Поле source_product_id вказує, з якого продукту основного набору даних були використані фотографії упаковки. Це дозволяє уникнути дублювання зображень та зберігати зв’язок між наборами даних. Поле user_checks_type визначає тип персонального запиту користувача. Блок input містить текст запиту користувача, який передається моделі під час аналізу. Блок expected_output містить еталонну відповідь, сформовану експертом вручну після аналізу складу продукту.

Для експерименту було визначено набір попередньо заданих перевірок, які користувач зможе обрати у застосунку. До них належать: горіхи, арахіс, глютен, лактоза, яйця, соя, морепродукти, сульфіти, ароматизатори, барвники, консерванти, підсилювачі смаку, кофеїн, пальмова олія, трансжири, шкідливі Е-добавки, ГМО, цукор, сіль, а також перевірки «чи можна при діабеті», «чи можна при вагітності» та «чи можна дітям». Окрім вибору готових перевірок, у застосунку також буде передбачено окреме текстове поле, у якому користувач може самостійно ввести довільний персональний запит.

Залежно від типу запиту було виділено різні категорії кейсів, такі як tags_only, де лише попередньо визначені перевірки, потім custom_only, де лише

релевантний довільний текстовий запит, також mixed, де релевантна комбінація попередньо визначених перевірок і текстового запиту, і irrelevant, де повністю нерелевантний запит, ще mixed_irrelevant, де частково релевантний запит, що містить як релевантні, так і нерелевантні елементи.

2.8.2 Розробка системи оцінювання

Для оцінювання якості оброблення персональних запитів користувача було розроблено окрему систему критеріїв оцінювання. Її основною метою є перевірка того, наскільки коректно модель аналізує додатковий запит користувача з урахуванням складу конкретного харчового продукту.

Режим А використовується для релевантних або частково релевантних запитів, а режим Б використовується для повністю нерелевантних запитів. Релевантними вважаються запити, що стосуються складу продукту, алергенів, протипоказань, наявності певних компонентів або впливу продукту на здоров'я. Нерелевантними вважаються запити, які не пов'язані з аналізом харчових продуктів або не мають змістового зв'язку із задачею системи. У випадку частково релевантного запиту модель повинна коректно проігнорувати нерелевантні складники та надати відповідь лише на релевантну частину запиту користувача.

Для режиму А фінальна оцінка розраховується як сума оцінок усіх блоків:

$$Score = \sum_{i=1}^{3} \left(\frac{s_i}{m_i} \cdot w_i \right) \cdot 10, \tag{2.3}$$

де i – номер блоку;

s_i – сума балів за блок;

m_i – максимально можлива кількість балів для блоку;

w_i – вага відповідного блоку.

Для режиму Б використовується окрема формула:

$$Score = \frac{s_1}{2} \cdot 10, \quad (2.4)$$

де s_1 – сума балів першого блоку.

Максимальна фінальна оцінка для одного елемента набору даних (кейсу) становить 10 балів.

Критерії оцінювання було згруповано у три основні блоки. Повний перелік критеріїв наведено у додатку Е.

Перший блок оцінює коректність класифікації запиту користувача. У межах цього блоку перевіряється, чи правильно модель визначила релевантність запиту, а також чи змогла вона коректно відмовити у випадку нерелевантного запиту.

Другий блок відповідає за фактичну коректність відповіді. Тут оцінюється достовірність наданої інформації та повнота відповіді на всі релевантні пункти запиту користувача.

Третій блок оцінює якість сформованої відповіді. У цьому блоці враховується простота мови та конкретність відповіді, тобто наскільки модель спирається саме на склад аналізованого продукту, а не на загальні знання.

2.8.3 Оцінювання якості відповідей моделі на персональний запит користувача за розробленою системою оцінювання

Спочатку було проведено експертне оцінювання результатів роботи моделі Gemini 2.5 Pro з визначенами параметрами вище за розробленою системою критеріїв. Результати наведено в таблиці 2.3.

Отримані результати демонструють дуже високу якість роботи моделі. Середня оцінка склала 9,903 бала з 10 можливих. Майже всі кейси отримали максимальну оцінку, а незначне зниження оцінки спостерігалось лише в окремих

випадках через менш конкретні формулювання відповіді або недостатню деталізацію окремих пунктів запиту.

Таблиця 2.3 – Результати експертного оцінювання якості оброблення персональних запитів користувача

Номер кейсу	1.1	1.2	2.1	2.2	3.1	3.2	БЛОК 1	БЛОК 2	БЛОК 3	Оцінка	Час (с)
UC_001	1	N/A	2	2	1	2	3,000	3,500	3,500	10,000	24,7
UC_002	1	1	N/A	N/A	N/A	N/A	10,000	N/A	N/A	10,000	18,0
UC_003	1	N/A	2	2	1	1	3,000	3,500	2,333	8,833	25,6
UC_004	1	N/A	2	2	1	2	3,000	3,500	3,500	10,000	15,6
UC_005	1	N/A	2	2	1	2	3,000	3,500	3,500	10,000	14,3
UC_006	1	N/A	2	2	1	2	3,000	3,500	3,500	10,000	17,6
UC_007	1	N/A	2	2	1	2	3,000	3,500	3,500	10,000	21,6
UC_008	1	N/A	2	2	1	2	3,000	3,500	3,500	10,000	19,9
UC_009	1	N/A	2	2	1	2	3,000	3,500	3,500	10,000	11,9
UC_010	1	N/A	2	2	1	1	3,000	3,500	3,500	10,000	15,1
UC_011	1	1	N/A	N/A	N/A	N/A	10,000	N/A	N/A	10,000	12,3
UC_012	1	N/A	2	2	1	1	3,000	3,500	3,500	10,000	13,3
Середня оцінка (Gemini 2.5 Pro)										9,903	17,5

Окремо також було проаналізовано час виконання запитів. Середній час оброблення одного кейсу склав 17,5 секунд. Це свідчить про те, що додатковий персональний аналіз користувацького запиту не створює критичного збільшення часу роботи системи та може використовуватись у практичному застосуванні без суттєвого погіршення користувацького досвіду.

Оскільки подальше експертне оцінювання великої кількості кейсів є трудомістким, було вирішено автоматизувати цей процес за допомогою підходу LLM-as-a-Judge. Для цього було використано модель GPT-5.4, для якої було розроблено окремий запит із критеріями оцінювання. Повний текст запиту наведено у додатку Ж.

Результати автоматизованого оцінювання наведено в таблиці 2.4.

Результати GPT-5.4 виявились дуже близькими до експертного оцінювання. Середня оцінка склала 9,757 бала, що лише незначно відрізняється

від результатів ручного аналізу. Основні розбіжності спостерігались у випадках, де оцінка залежала від інтерпретації повноти або конкретності відповіді. Водночас модель стабільно зберігала загальну логіку оцінювання та коректно застосовувала розроблені критерії.

Таблиця 2.4 – Результати автоматизованого оцінювання якості оброблення персональних запитів користувача за допомогою GPT-5.4

Номер кейсу	1.1	1.2	2.1	2.2	3.1	3.2	БЛОК 1	БЛОК 2	БЛОК 3	Оцінка	Час (с)
UC_001	1	N/A	2	2	1	2	3,000	3,500	3,500	10,000	24,7
UC_002	1	1	N/A	N/A	N/A	N/A	10,000	N/A	N/A	10,000	18,0
UC_003	1	N/A	2	1	1	2	3,000	2,625	3,500	9,125	25,6
UC_004	1	N/A	2	2	1	1	3,000	3,500	2,333	8,833	15,6
UC_005	1	N/A	2	2	1	2	3,000	3,500	3,500	10,000	14,3
UC_006	1	N/A	2	2	1	2	3,000	3,500	3,500	10,000	17,6
UC_007	1	N/A	2	2	1	2	3,000	3,500	3,500	10,000	21,6
UC_008	1	N/A	1	2	1	2	3,000	2,625	3,500	9,125	19,9
UC_009	1	N/A	2	2	1	2	3,000	3,500	3,500	10,000	11,9
UC_010	1	N/A	2	2	1	2	3,000	3,500	3,500	10,000	15,1
UC_011	1	1	N/A	N/A	N/A	N/A	10,000	N/A	N/A	10,000	12,3
UC_012	1	N/A	2	2	1	1	3,000	3,500	3,500	10,000	13,3
Середня оцінка (Gemini 2.5 Pro)										9,757	17,5

Таким чином, GPT-5.4 може бути використана для автоматизованого оцінювання якості оброблення персональних запитів користувача.

2.9 Розробка алгоритму

На основі проведеного аналізу, сформованих наборів даних, систем оцінювання та результатів експериментів було розроблено загальний алгоритм роботи системи аналізу складу харчових продуктів з використанням MLLM.

Загальна схема роботи системи починається зі збору вхідних даних від користувача. Користувач завантажує від одного до чотирьох зображень упаковки

продукту.

Після завантаження фотографій користувач за бажанням може сформувати персональний запит для додаткової перевірки продукту. Для цього система підтримує два способи введення. Перший спосіб передбачає вибір готових запитів із попередньо визначеного переліку, наприклад перевірка на наявність алергенів, лактози, глютену, консервантів або оцінка можливості вживання продукту при вагітності чи діабеті. Другий спосіб дозволяє користувачу самостійно ввести довільний текстовий запит.

Далі система формує запит до MLLM. Якщо користувач не задав додаткових побажань, використовується базовий запит для аналізу складу продукту. Якщо ж користувач сформував персональний запит, система динамічно доповнює основний запит відповідними інструкціями для моделі. Таким чином забезпечується адаптація процесу аналізу до конкретних потреб користувача.

Далі зображення упаковки та текстовий запит передаються до MLLM Gemini 2.5 Pro. Модель виконує аналіз зображень, розпізнає текст упаковки, визначає склад продукту та формує структуровану відповідь у форматі JSON відповідно до заданої схеми.

На наступному етапі серверна частина застосунку отримує JSON-відповідь, виконує її перевірку та десеріалізацію. Після цього інформація перетворюється у внутрішні об’єкти системи та передається до клієнтського застосунку для відображення користувачу.

У результаті користувач отримує структурований аналіз продукту, що містить перелік інгредієнтів, їх переклад, опис, можливу користь та шкоду, а також відповідь на персональний запит, якщо він був заданий.

3 РОЗРОБЛЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ ANDROID ДЛЯ РОЗПІЗНАВАННЯ ТА ПОЯСНЕННЯ СКЛАДУ ХАРЧОВИХ ПРОДУКТІВ З АВТОМАТИЗОВАНИМ ПЕРЕКЛАДОМ УКРАЇНСЬКОЮ МОВОЮ

3.1 Вибір та налаштування програмного середовища

Для розробки мобільного застосунку було обрано архітектуру, що складається з мобільного клієнта на основі React Native та серверної частини на основі FastAPI.

React Native було обрано як фреймворк для розробки мобільного клієнта, оскільки він дозволяє створювати застосунки для Android з використанням JavaScript та TypeScript. Порівняно з нативною розробкою на Kotlin, React Native суттєво скорочує час розробки завдяки єдиній кодовій базі та великій кількості готових бібліотек. Оскільки застосунок орієнтований виключно на платформу Android, використання кросплатформного фреймворку залишає можливість розширення на iOS у майбутньому без значних змін у коді.

Для запуску та тестування React Native застосунку використовувалося Expo Go, яке дозволяє переглядати зміни в застосунку в реальному часі на фізичному пристрої без необхідності щоразу збирати повний APK-файл. Налаштування середовища здійснювалось відповідно до офіційної документації Expo.

Серверну частину реалізовано на мові Python з використанням фреймворку FastAPI. Цей вибір обумовлений кількома причинами. По-перше, FastAPI забезпечує високу продуктивність завдяки асинхронній обробці запитів. По-друге, Python має широку підтримку бібліотек для роботи з машинним навчанням та API зовнішніх сервісів, що є важливим для інтеграції з Gemini API. Для запуску серверу використовувався Uvicorn як асинхронний інтерфейс шлюзу сервера (ASGI). Обробка зображень, що надходять від мобільного клієнта, здійснюється за допомогою бібліотеки Pillow. Для роботи з Gemini API використовувалась офіційна бібліотека google-genai. Взаємодія між клієнтом та

сервером реалізована через передачу зображень у форматі multipart/form-data, для підтримки якого підключено бібліотеку python-multipart. Керування змінними середовища, зокрема ключем API, здійснювалося через бібліотеку python-dotenv. Налаштування серверного середовища також виконувалось відповідно до офіційної документації FastAPI та Uvicorn.

Як середовище розробки використовувався Visual Studio Code, який підходить як для написання коду на TypeScript у частині React Native, так і для роботи з Python у частині FastAPI. Редактор забезпечує зручну навігацію між файлами проєкту, підсвічування синтаксису та вбудований термінал для запуску команд.

3.2 Технічне завдання

Метою розробки є мобільний застосунок для платформи Android, який дозволяє користувачу сфотографувати упаковку харчового продукту та отримати детальний аналіз його складу українською мовою з поясненням кожного інгредієнта.

Застосунок має забезпечувати розпізнавання тексту складу продукту з фотографії упаковки, опис кожного інгредієнта з поясненням що це таке, звідки береться і яку функцію виконує у продукті, а також виявлення корисних і шкідливих властивостей кожного інгредієнта для здоров’я. Для імпортних товарів передбачено автоматичний переклад назви продукту та назв інгредієнтів українською мовою. Застосунок також має надавати відповідь на персональний запит користувача щодо складу продукту. Для цього підтримуються двадцять два готових теги-фільтри, серед яких горіхи, арахіс, глютен, лактоза, яйця, соя, морепродукти, сульфіти, ароматизатори, барвники, консерванти, підсилювачі смаку, кофеїн, пальмова олія, трансжири, шкідливі Е-добавки, ГМО, цукор, сіль, придатність при діабеті, при вагітності та для дітей, а також поле для введення довільного запиту у вільній формі.

Застосунок має працювати у режимі 24/7 без перерв у доступності, а час обробки одного запиту до моделі не повинен перевищувати п'ятдесят секунд. Застосунок не зберігає інформацію про попередні аналізи, оскільки кожен запит є незалежним. Реєстрація або авторизація користувача не передбачена, адже метою застосунку є швидке надання інформації про склад продукту без зайвих кроків.

Детальну функціональність застосунку наведено на рисунку 3.1 у вигляді діаграми варіантів використання. Основним актором є користувач, а зовнішнім сервісом, що бере участь в обробці запитів, є Gemini API.

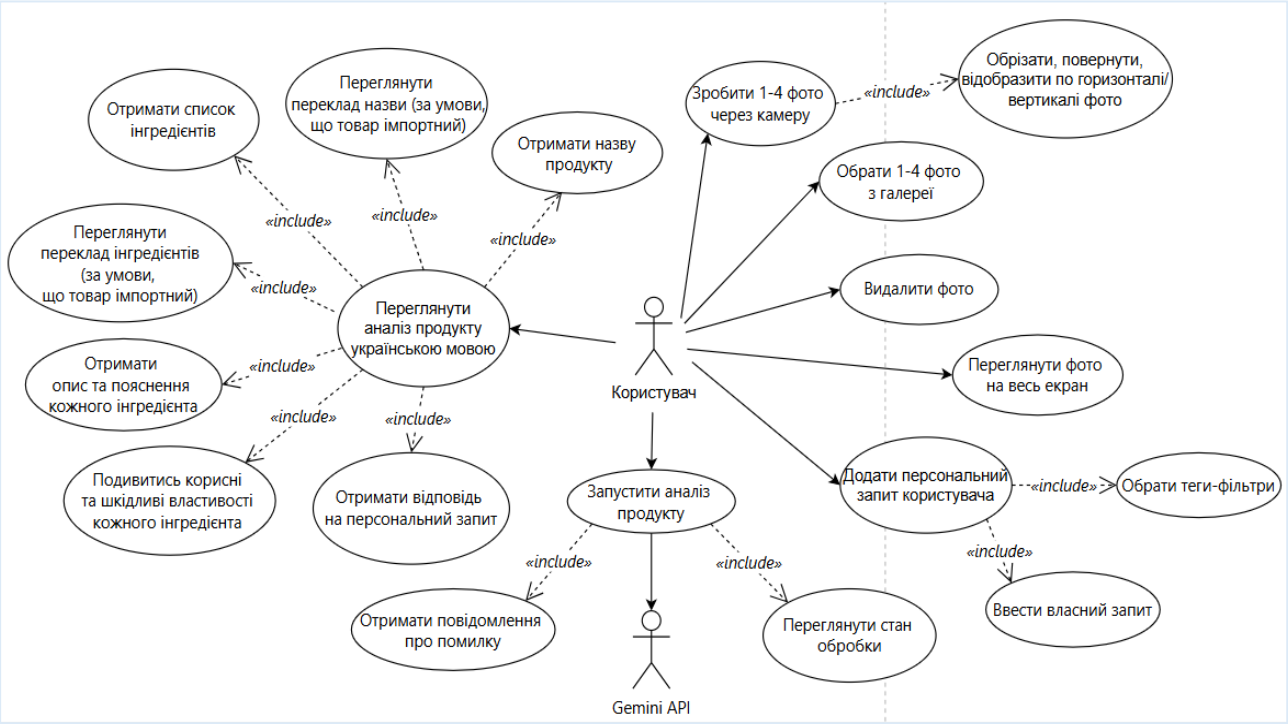


Рисунок 3.1 – Діаграма варіантів використання

3.3 Проєктування дизайну користувацького інтерфейсу

Перед розробкою застосунку було проведено проєктування користувацького інтерфейсу у середовищі Figma. Для цього було розроблено User Flow діаграму та створено макети екранів. Це дозволило заздалегідь продумати логіку навігації між екранами, визначити розміщення елементів на

кожному екрані та узгодити загальний візуальний стиль застосунку до початку написання коду.

Для кращого розуміння логіки переходів між станами застосунку спочатку було визначено умовні позначення, які використовуються в User Flow діаграмі. Легенду наведено на рисунку 3.2.

Рисунок 3.2 – Легенда до User Flow діаграми застосунку

Відповідно до легенди, прямокутники з блакитним фоном позначають екрани застосунку, жовті ромби є точками прийняття рішень, зелені прямокутники відповідають діям користувача, червоні прямокутники позначають стани помилок, а фіолетові прямокутники містять зміст результата аналізу складу.

На основі визначених позначень було побудовано User Flow діаграму, яка відображає повний шлях користувача в застосунку від запуску до отримання результату аналізу з урахуванням усіх можливих розгалужень і помилок. Діаграму наведено на рисунку 3.3.

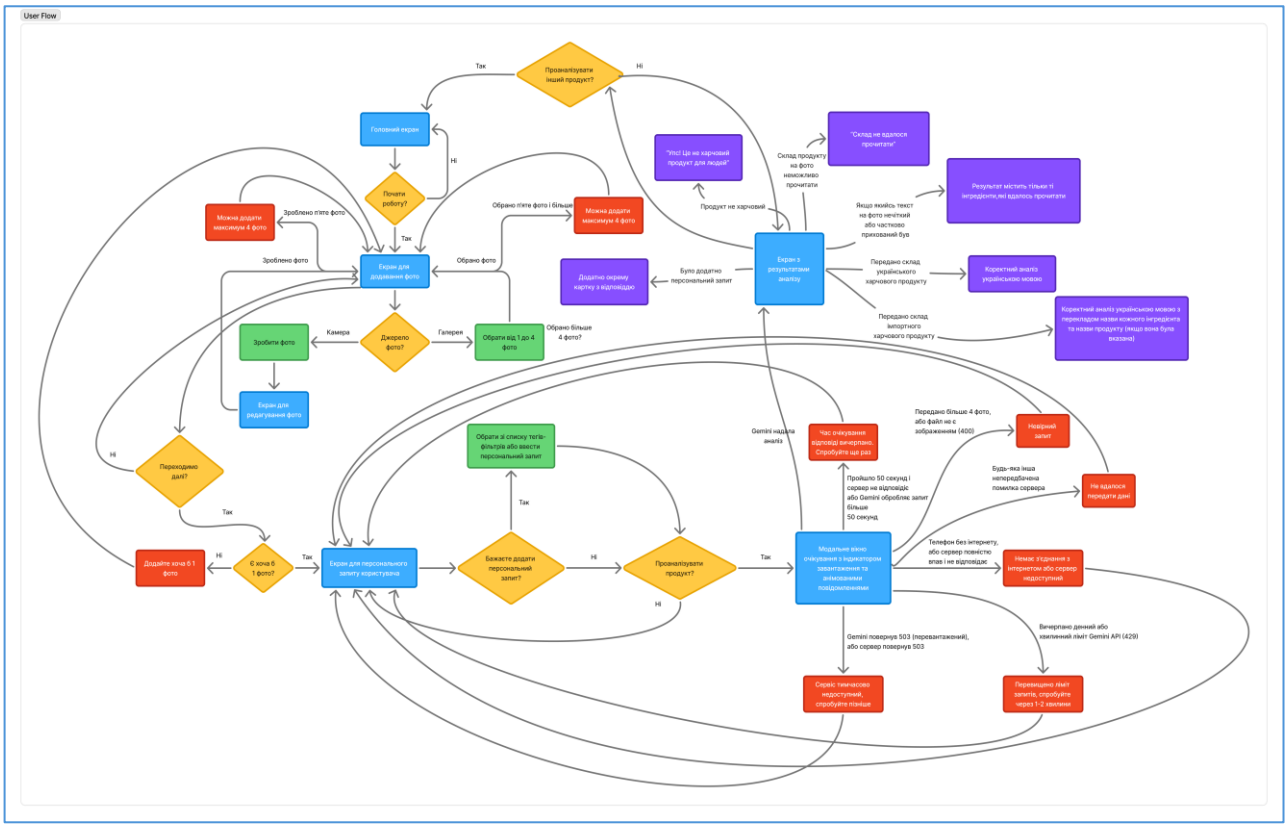


Рисунок 3.3 – User Flow діаграма мобільного застосунку аналізу складу харчових продуктів

Діаграма охоплює чотири основні гілки розвитку подій. Перша гілка описує успішний сценарій, коли користувач додає фото, за бажанням вводить персональний запит і отримує повний аналіз складу продукту. Друга гілка охоплює ситуацію, коли користувач намагається перейти далі без додавання фото, і застосунок показує відповідне повідомлення. Третя гілка описує помилки на етапі аналізу, наприклад перевищення часу очікування, відсутність інтернету або помилку сервера, після чого користувач залишається на екрані персонального запиту і може спробувати ще раз. Четверта гілка охоплює ситуації на екрані результату, коли продукт не є харчовим або склад не вдалося прочитати з фотографії.

Таким чином, застосунок складається з чотирьох основних екранів. Перший екран є головним і відображає логотип та назву застосунку, а також містить кнопку для переходу до аналізу продукту.

Другий екран призначений для додавання фотографій. На ньому

реалізовано запит дозволу на використання камери та доступу до галереї. Користувач може зробити фото через камеру з можливістю обрізки або обрати до чотирьох фотографій з галереї одночасно. Додані фотографії відображаються у сітці 2 на 2, кожну з них можна переглянути на весь екран або видалити. У разі спроби додати більше 4 фотографій застосунок показує відповідне повідомлення. Зверху має бути підказка щодо кількості фото. Внизу екрану розміщено підказку щодо якості фото та кнопку переходу до наступного екрану. Перейти на наступний екран можна тільки після додавання хочаб 1 фото.

Третій екран призначений для введення персонального запиту. Користувач може обрати один або кілька готових тегів-фільтрів із двадцяти двох доступних варіантів, серед яких горіхи, глютен, лактоза, кофеїн, ГМО, придатність для дітей тощо. Повторне натискання на тег знімає його вибір. Також є поле для введення власного довільного запиту, при цьому клавіатура не перекриває поле вводу. Якщо користувач не обирає жодного тегу і не вводить запит, застосунок виконує аналіз за замовчуванням. Після натискання кнопки «Проаналізувати» відображається модальне вікно очікування з індикатором завантаження та анімованими повідомленнями, що змінюються кожні чотири секунди. Передбачено таймаут у п’ятдесят секунд на випадок відсутності відповіді від сервера, а також обробку серверних помилок із зрозумілими повідомленнями для користувача.

Четвертий екран відображає результати аналізу. На початку екрану розміщено попередження про те, що аналіз виконано штучним інтелектом і може містити неточності. Далі відображається назва продукту з упаковки та її переклад українською мовою у випадку, якщо продукт є іноземним. Якщо користувач вводив персональний запит, відповідь на нього виділяється в окрему картку. Основну частину екрану займає список усіх інгредієнтів зі збереженням оригінального порядку, де для кожного інгредієнта наводяться оригінальна назва з упаковки, переклад українською мовою за потреби, опис інгредієнта, а також його користь і шкода для здоров’я. У разі якщо продукт не є харчовим або склад не вдалося прочитати, застосунок показує зрозуміле повідомлення про помилку.

На цьому екрані також доступна кнопка повернення на головний екран для аналізу іншого продукту.

На всіх екранах передбачено кнопку повернення на попередній екран.

Макети всіх чотирьох екранів, розроблені у Figma, наведено на рисунку 3.4.

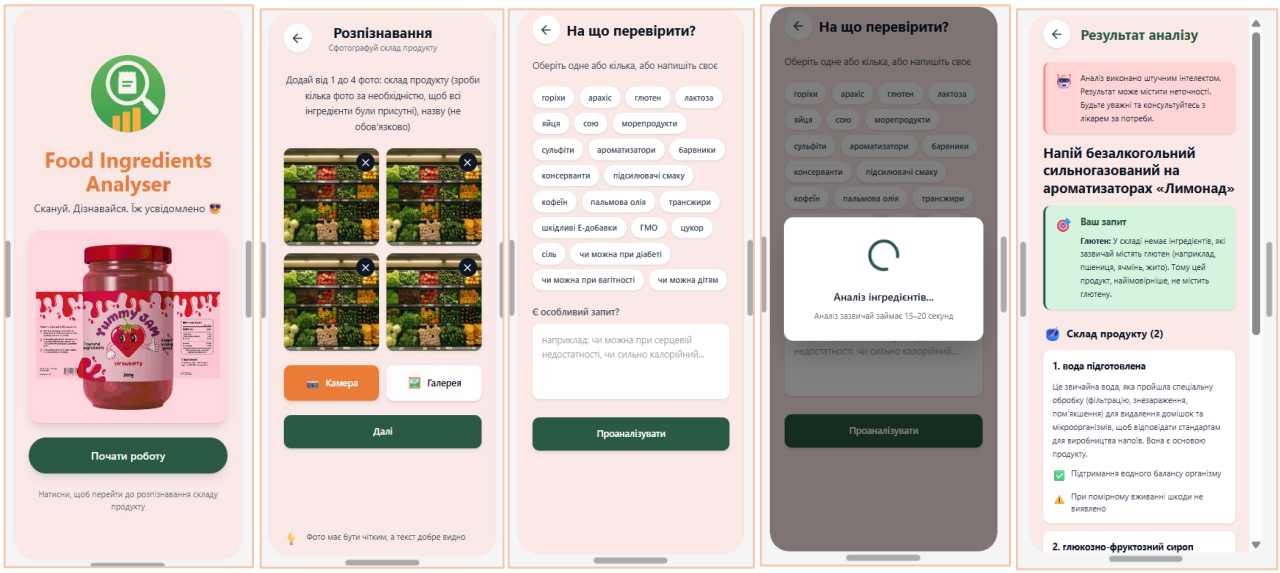


Рисунок 3.4 – Макети екранів мобільного застосунку, розроблені у Figma

3.4 Розробка архітектури застосунку

Застосунок отримав назву Food Ingredients Analyser. Його побудовано за клієнт-серверною архітектурою та складається з трьох рівнів: мобільного клієнта, сервера та зовнішнього сервісу. Такий підхід дозволяє чітко розмежувати відповідальність між частинами системи та спростити підтримку і розширення застосунку в майбутньому. Загальну структуру проєкту наведено на рисунку 3.5.

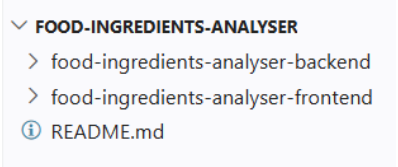


Рисунок 3.5 – Структура кореневого каталогу проєкту Food Ingredients Analyser

Мобільний клієнт розроблено на React Native і він відповідає за взаємодію з користувачем, а саме за отримання фотографій, вибір параметрів перевірки та відображення результату аналізу. Серверна частина реалізована на FastAPI і виступає посередником між мобільним застосунком і мовною моделлю. Сервер приймає зображення та параметри від клієнта, формує структурований запит до моделі, отримує відповідь і повертає її клієнту у форматі JSON. Для аналізу складу продукту використовується MLLM Gemini 2.5 Pro, до якої сервер звертається через офіційний Python SDK google-genai. Модель отримує зображення разом із текстовим промтом і повертає структурований результат у форматі JSON із детальним описом кожного інгредієнта. Такий розподіл на три рівні є доцільним, оскільки вся обчислювальна робота з обробки зображень і взаємодії зі стороннім сервісом виконується на сервері, а мобільний клієнт залишається легким і відповідає лише за відображення даних та взаємодію з користувачем.

Взаємодія між клієнтом і сервером відбувається через REST API. Клієнт надсилає HTTP POST запит у форматі multipart/form-data, що містить фотографії продукту та додаткові параметри користувача. Вибір формату multipart/form-data обумовлений необхідністю передавати одночасно бінарні дані у вигляді зображень і текстові параметри в одному запиті. Сервер обробляє запит, звертається до Gemini API і повертає клієнту структуровану JSON відповідь, яку застосунок відображає на екрані результату.

Серверна частина застосунку організована у вигляді модульної структури з чітким розподілом відповідальностей між окремими компонентами. Структуру серверної частини наведено на рисунку 3.6.

Точкою входу серверної частини є файл main.py, який створює FastAPI додаток, підключає налаштування CORS та реєструє роутер з усіма ендпоінтами. Налаштування CORS політики винесено в окремий модуль core/cors.py, що дозволяє керувати переліком дозволених джерел запитів незалежно від основної логіки застосунку. Зчитування змінних середовища з файлу .env, зокрема ключа Gemini API, здійснюється через модуль core/config.py з використанням

бібліотеки pydantic-settings, що забезпечує валідацію конфігурації під час запуску сервера. Єдиний POST ендпоінт /analyze визначено у файлі api/analysis_routes.py, який приймає фотографії та параметри запиту від клієнта, валідує вхідні дані та передає їх до сервісного шару для подальшої обробки, після чого повертає готовий результат клієнту.

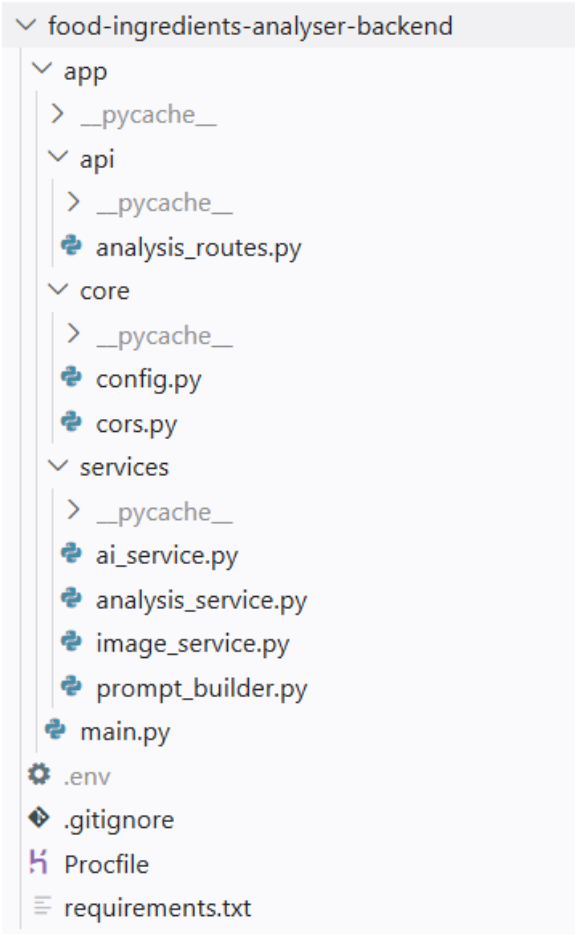


Рисунок 3.6 – Структура серверної частини застосунку

Бізнес-логіку серверної частини реалізовано у папці services і розподілено між чотирма окремими модулями, кожен з яких відповідає за свою частину роботи. Файл image_service.py відповідає за обробку завантажених файлів. Він перевіряє що вони є коректними зображеннями і перетворює їх на PIL об’єкти, придатні для передачі до MLLM. Файл prompt_builder.py відповідає за формування текстового промту для Gemini залежно від параметрів, які користувач вказав у застосунку. Якщо користувач обрав теги-фільтри або ввів

власний запит, вони включаються у промт у структурованому вигляді. Файл analysis_service.py координує весь процес аналізу і передає підготовлений промт разом із зображеннями до ai_service.py. Файл ai_service.py безпосередньо звертається до Gemini API через офіційний SDK, надсилає запит і обробляє отриману відповідь. Також цей модуль відповідає за обробку різних типів помилок, зокрема перевищення часу очікування, перевищення ліміту запитів та недоступності сервісу, і повертає клієнту зрозумілі повідомлення про помилки замість технічних деталей. Окрім модулів із кодом, серверна частина містить файл requirements.txt зі списком усіх залежностей Python, необхідних для роботи застосунку, файл Procfile з інструкцією для запуску сервера на хостингу, а також файл .env, у якому зберігаються секретні змінні середовища і який не потрапляє до репозиторію.

Структуру клієнтської частини застосунку наведено на рисунку 3.7. Вона організована відповідно до підходу файлової маршрутизації Expo Router, де кожен файл у папці app відповідає окремому екрану застосунку і автоматично стає доступним як окремий маршрут навігації. Такий підхід спрощує організацію коду і робить структуру проєкту зрозумілою, оскільки назва файлу одразу вказує на те, який екран він реалізує.

Файл index.tsx реалізує головний екран застосунку, на якому відображається логотип і кнопка для початку аналізу продукту. Файл camera.tsx реалізує екран додавання фото і містить усю логіку роботи з камерою та галереєю пристрою, відображення доданих фотографій у сітці, можливість перегляду кожного фото на весь екран та його видалення. Файл preferences.tsx реалізує екран персонального запиту і відповідає за відображення тегів-фільтрів, поля для введення власного запиту, а також за запуск процесу аналізу та відображення анімації очікування під час обробки запиту сервером. Файл analysisResult.tsx реалізує екран результату і відображає назву продукту, відповідь на персональний запит користувача у виділеній картці та список усіх інгредієнтів із їхніми описами. Кореневий layout застосунку і навігацію між усіма екранами налаштовано у файлі _layout.tsx з використанням Expo Router.

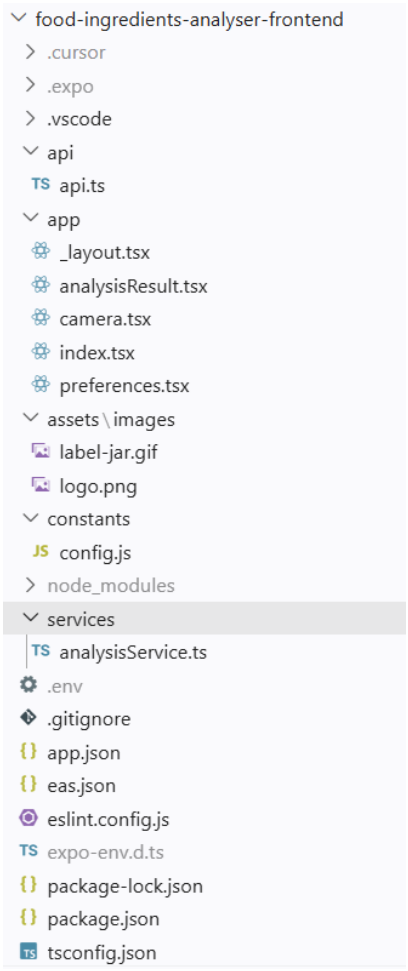


Рисунок 3.7 – Структура клієнтської частини застосунку

Мережеву взаємодію з сервером розподілено між двома модулями. Файл api/api.ts реалізує базову функцію для виконання HTTP запитів і централізовано обробляє помилки мережі та сервера, формуючи зрозумілі повідомлення для відображення користувачу. Файл services/analysisService.ts відповідає за підготовку запиту до сервера і формує multipart/form-data запит із фотографіями та параметрами користувача, після чого передає його до api.ts для виконання. URL бекенду зберігається у файлі constants/config.js, що дозволяє змінювати адресу сервера в одному місці без необхідності вносити зміни у код кількох файлів. При локальній розробці адреса сервера також зберігається у файлі .env як змінна середовища. Конфігурацію застосунку, зокрема його назву, іконку та налаштування для платформи Android, визначено у файлі app.json, а конфігурацію для збірки APK файлу через сервіс EAS Build визначено у файлі eas.json. У папці assets/images зберігаються логотип застосунку та анімація, яка

відображається на екрані персонального запиту під час очікування відповіді від сервера.

3.5 Приклади роботи застосунку

Для демонстрації роботи застосунку було проведено аналіз складу молочного шоколаду українського виробника. Фото було зроблено через камеру. Також додатково користувач захотів перевірити продукт на кофеїн, ГМО та мед. Процес додавання фото та вибору параметрів перевірки наведено на рисунку 3.8.

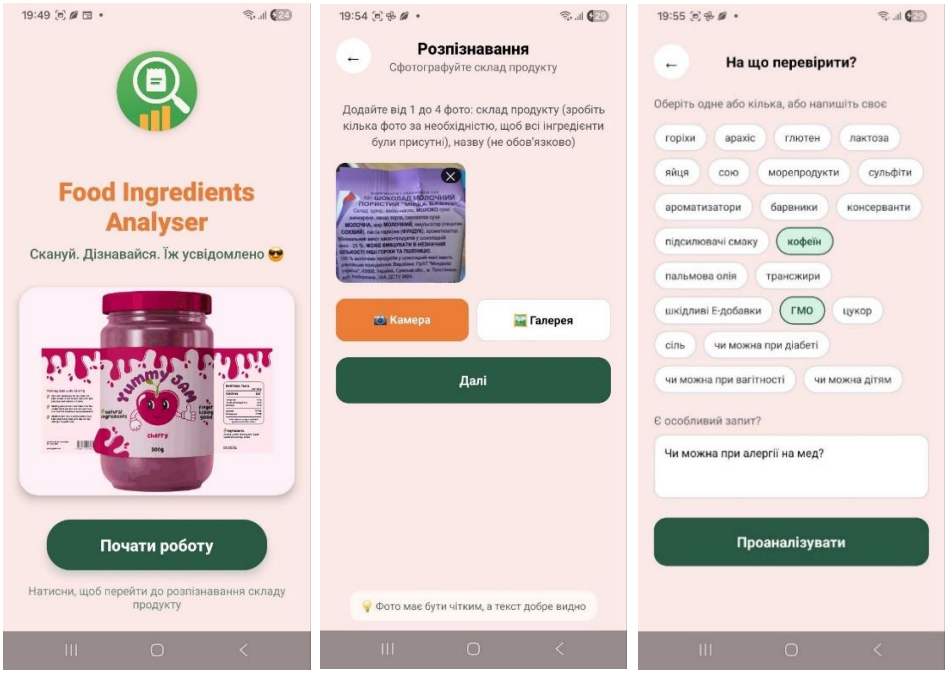


Рисунок 3.8 – Екрани додавання фото та вибору параметрів перевірки для молочного шоколаду

За результатами аналізу застосунок коректно зчитав та розпізнав усі дев’ять інгредієнтів зі складу продукту. Для кожного інгредієнта надано оригінальну назву з упаковки, опис того що це таке і яку функцію він виконує, а також інформацію про користь і можливу шкоду для здоров’я. Відповідь на персональний запит щодо кофеїну, ГМО та меду виділено в окрему картку на початку екрану результату. На рисунку 3.9 для прикладу наведено аналіз трьох

інгредієнтів зі складу, оскільки повний аналіз усіх дев’яти інгредієнтів займає значний обсяг.

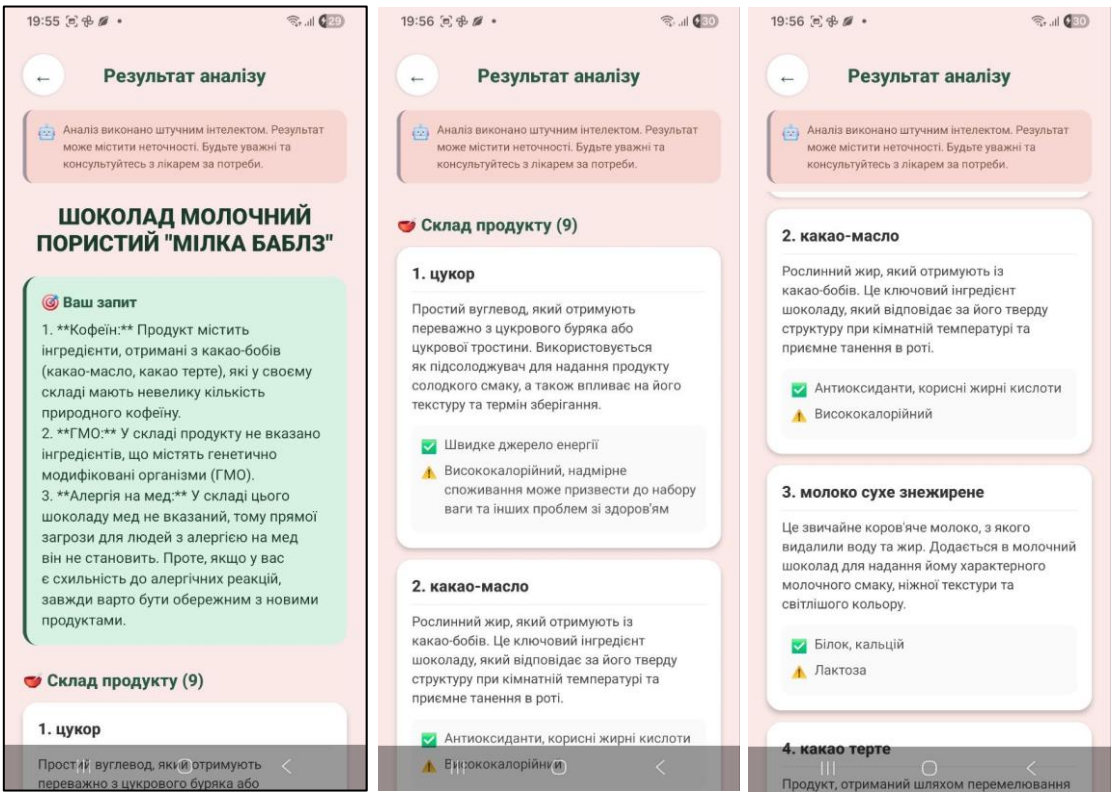


Рисунок 3.9 – Екран результату аналізу складу молочного шоколаду

Для демонстрації роботи застосунку з імпортними товарами було проведено аналіз складу іноземного продукту. Назву продукту наведено в оригіналі та у перекладі, а для кожного інгредієнта також надано переклад поряд з оригінальною назвою з упаковки. Весь аналіз, включаючи описи, користь і шкоду кожного інгредієнта, надано українською мовою. Результат аналізу імпортного продукту наведено на рисунку 3.10.

Також було перевірено поведінку застосунку у двох граничних сценаріях. У першому випадку користувач завантажує фотографію нехарчового продукту, і застосунок відображає на екрані результату зрозуміле повідомлення про те, що переданий продукт не є харчовим. У другому випадку користувач передає розмите або нечітке фото упаковки, з якого неможливо прочитати склад, і застосунок повідомляє що склад не вдалося розпізнати. Обидва сценарії наведено на рисунку 3.11.

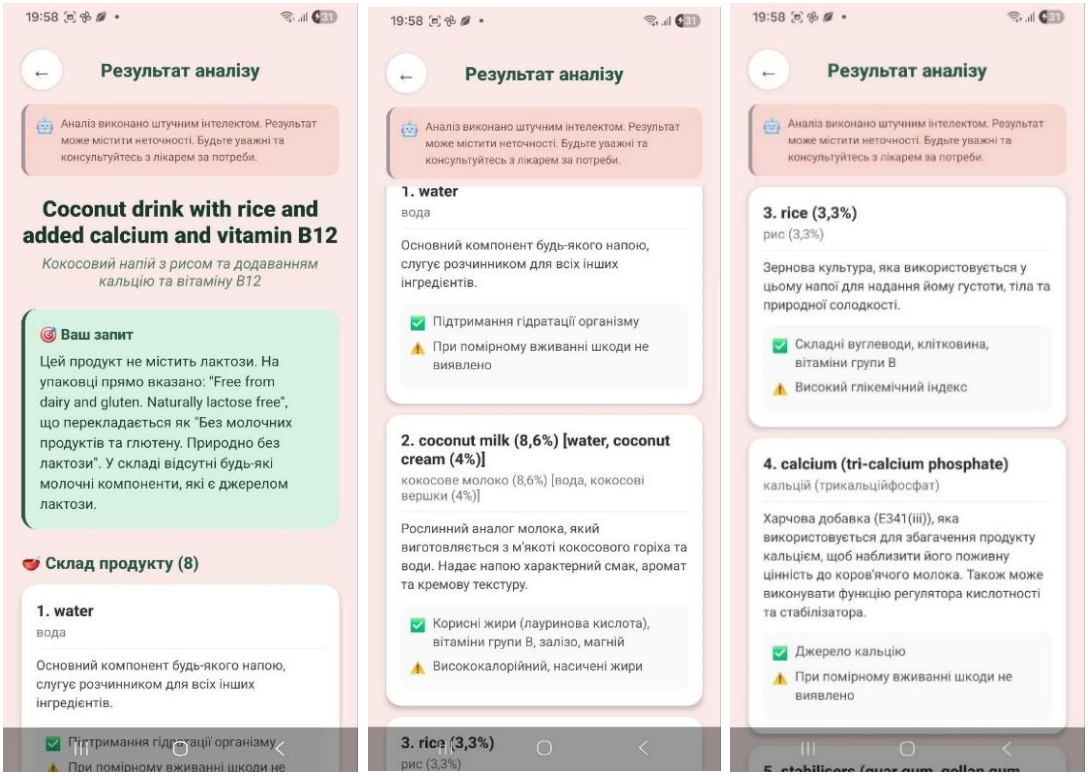


Рисунок 3.10 – Результат аналізу складу імпортного продукту з перекладом назв українською мовою

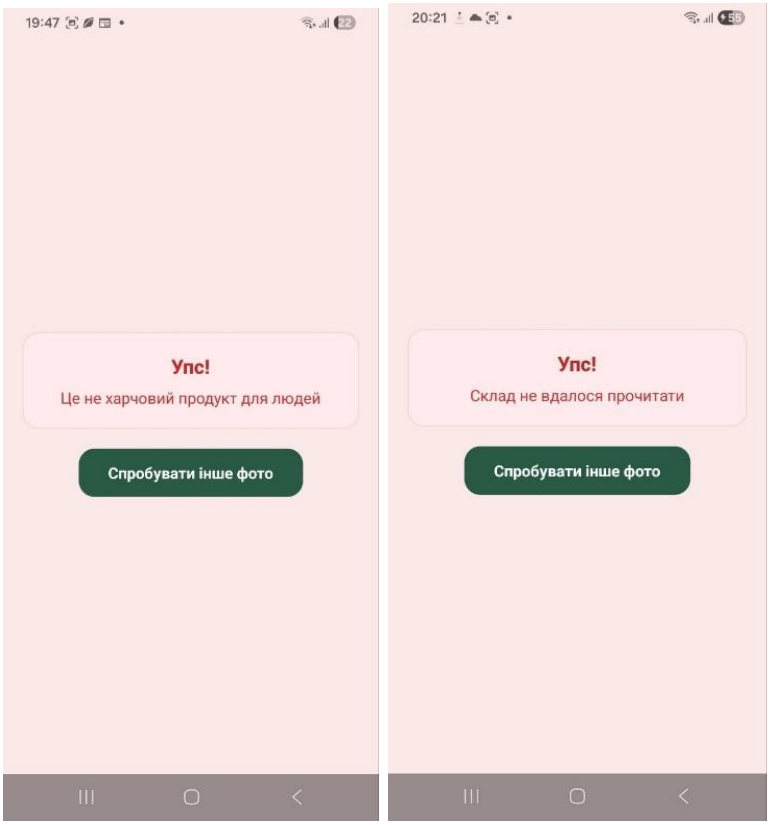


Рисунок 3.11 – Результати обробки нехарчового продукту та нечіткого фото

3.6 Оцінка якості роботи застосунку

Для оцінки якості роботи застосунку було проведено користувацьке тестування. Серверну частину застосунку розгорнуто на хмарній платформі Railway, що забезпечило доступність бекенду для реальних користувачів через інтернет. На основі конфігурації EAS Build було зібрано APK файл застосунку, який було передано користувачам для встановлення на пристрої Android.

Для збору відгуків користувачів було розроблено анкету у Google Forms, що складається з трьох секцій і охоплює дванадцять питань. Перша секція присвячена якості результатів аналізу, друга оцінці інтерфейсу застосунку, а третя загальним враженням від використання. Для оцінювання використовується шкала від однієї до п’яти зірок, де одна зірка відповідає оцінці «дуже погано», а п’ять зірок відповідають оцінці «відмінно». На момент написання роботи в опитуванні взяли участь 20 користувачів.

Перші три питання першої секції стосувались якості відповіді на персональний запит користувача, повноти списку знайдених інгредієнтів та зрозумілості й коректності їхніх описів. Усі три питання отримали переважно найвищі оцінки, що свідчить про високу якість аналізу та точність роботи моделі Gemini 2.5 Pro (рис. 3.12–3.14).

Рисунок 3.12 – Результати оцінювання якості відповіді на персональний запит користувача

Рисунок 3.13 – Результати оцінювання повноти списку знайдених інгредієнтів

Рисунок 3.14 – Результати оцінювання зрозумілості та коректності описів інгредієнтів

Питання щодо корисності інформації про вплив інгредієнтів на здоров’я отримали досить високі оцінки від учасників. Результати оцінювання корисності інформації про користь інгредієнтів наведено на рисунку 3.15, а про шкідливі властивості на рисунку 3.16. Більшість учасників оцінила обидва аспекти найвищим балом, що свідчить про те, що надана інформація є корисною та достатньо повною для користувачів.

Рисунок 3.15 – Результати оцінювання корисності інформації про користь інгредієнтів для здоров’я

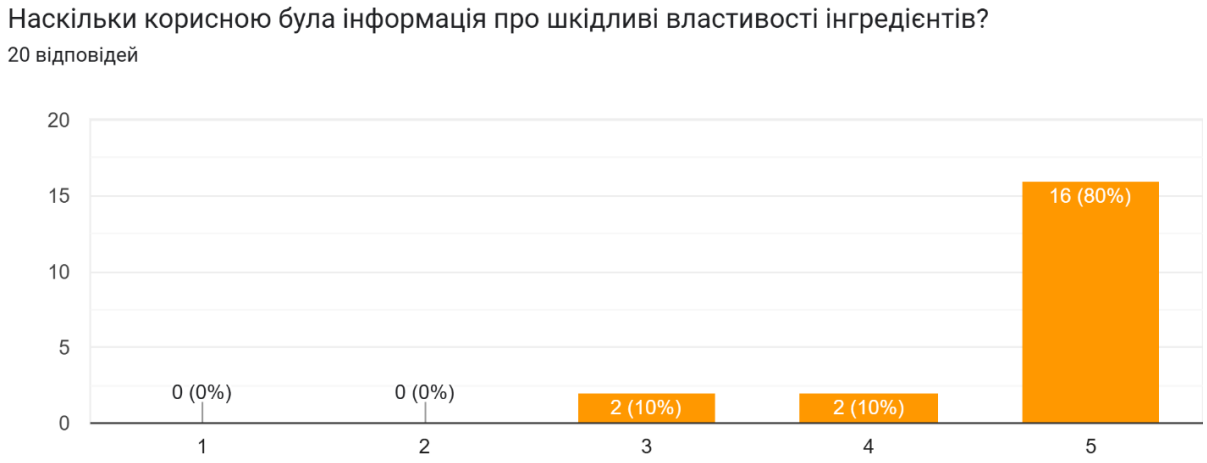


Рисунок 3.16 – Результати оцінювання корисності інформації про шкідливі властивості інгредієнтів

Оцінки швидкості отримання результату аналізу виявились дещо нижчими порівняно з іншими питаннями першої секції (рис. 3.17). Це пояснюється тим, що аналіз складу продукту приблизно займає від п’ятнадцяти до двадцяти секунд, що деякі користувачі сприймають як відносно тривале очікування.

Рисунок 3.17 – Результати оцінювання швидкості отримання результату аналізу

В другій секції були питання про зручність використання інтерфейсу та візуальне оформлення застосунку застосунку. Результати опитування свідчать про те, що розроблений інтерфейс в цілому є достатньо зручним і приємним для користувачів (рис. 3.18, 3.19).

Рисунок 3.18 – Результати оцінювання зручності інтерфейсу застосунку

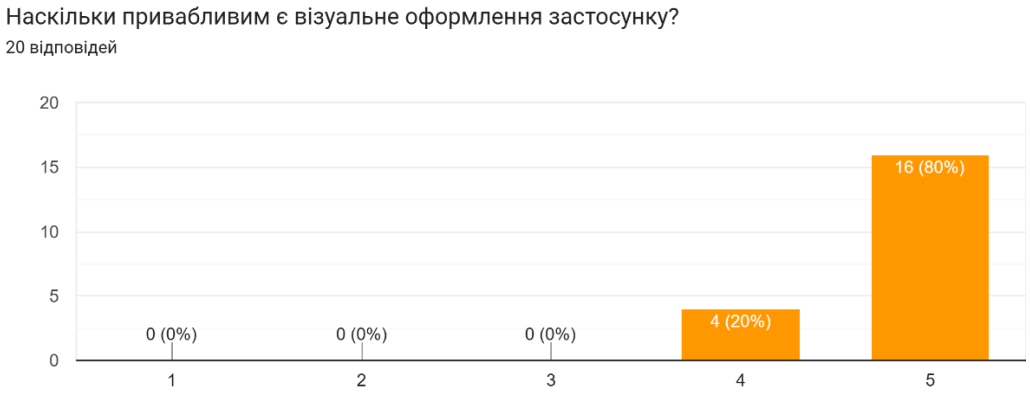


Рисунок 3.19 – Результати оцінювання візуального оформлення застосунку

Питання про особисту корисність отриманої інформації та ймовірність подальшого використання застосунку отримали переважно високі оцінки (рис. 3.20, 3.21). Більшість учасників підтвердила що інформація виявилась для них особисто корисною, а також висловила готовність користуватись застосунком надалі.

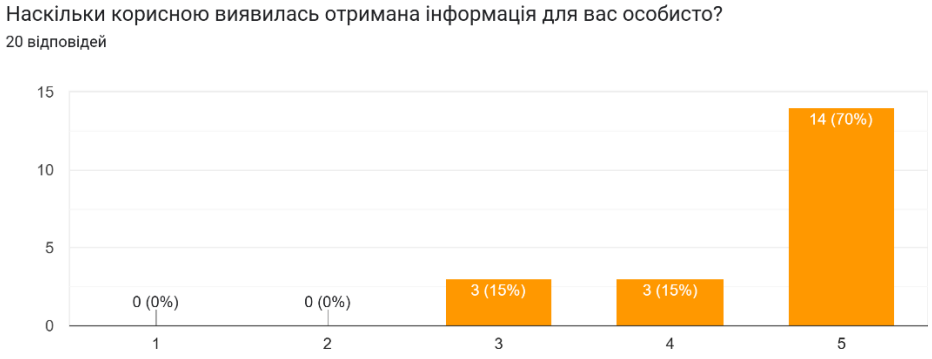


Рисунок 3.20 – Результати оцінювання особистої корисності отриманої інформації

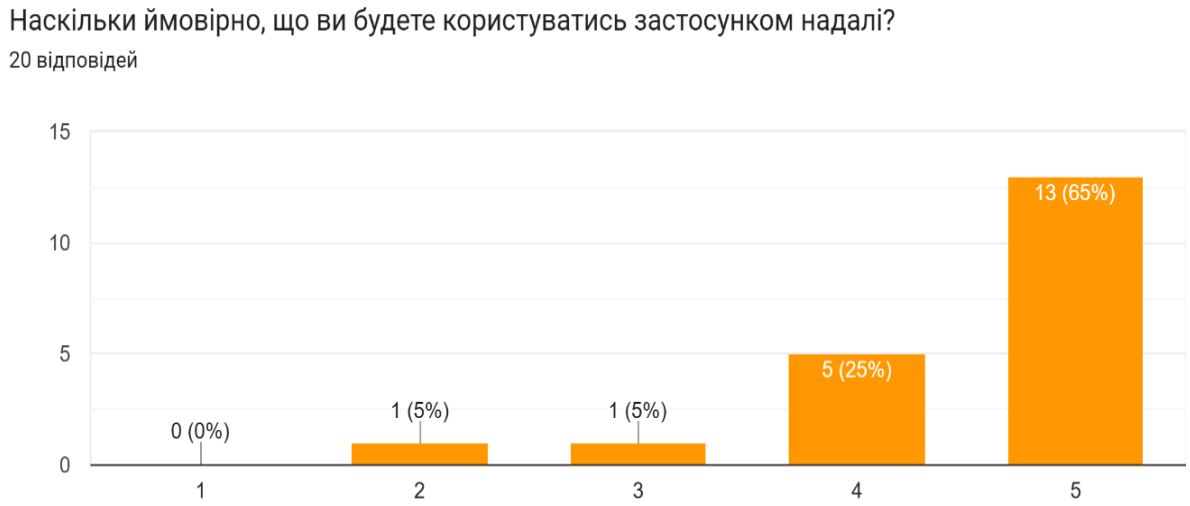


Рисунок 3.21 – Результати оцінювання ймовірності подальшого використання застосунку

Серед відгуків у довільній формі користувачі відзначали зручність використання, приємний дизайн та деталізованість наданої інформації. Кілька учасників окремо виділили можливість персонального запиту та отримання детальної відповіді на нього як одну з найцінніших функцій застосунку. Один із користувачів зазначив що застосунок виявився особливо корисним для перевірки складу продуктів для маленької дитини. Серед технічних зауважень один із учасників звернув увагу на незначні проблеми у інтерфейсі, зокрема нерівномірне розташування кнопки навігації та відображення службових символів у відповіді моделі. Серед побажань надійшли пропозиції щодо пришвидшення роботи застосунку та додавання можливості одночасного фотографування кількох сторін упаковки за один раз.

Останнє питання анкети стосувалось бажаних функцій у майбутніх версіях застосунку (рис. 3.22). Учасникам було запропоновано обрати серед таких варіантів як об’єднання однакових інгредієнтів, врахування кількості інгредієнта при оцінці його впливу на здоров’я, підтримка платформи iOS, персональний профіль з урахуванням алергій та обмежень, інформація про калорійність і поживну цінність, збереження історії аналізів, багатомовність, скорочений опис загальновідомих інгредієнтів, а також можливість додати власну пропозицію.

Рисунок 3.22 – Результати опитування щодо бажаних функцій у майбутніх версіях застосунку

Абсолютним лідером стала функція об’єднання однакових інгредієнтів, яку підтримала більшість учасників. На другому місці опинилась функція врахування кількості інгредієнта при оцінці його впливу на здоров’я, а на третьому підтримка платформи iOS. Значну підтримку також отримала функція персонального профілю з урахуванням алергій та обмежень користувача, яка увійшла до четвірки найбажаніших функцій. Більше половини учасників також зацікавились інформацією про калорійність і поживну цінність продукту та збереженням історії аналізів. Багатомовність та скорочений опис загальновідомих інгредієнтів отримали помірну підтримку. Окрім запропонованих варіантів двоє учасників додали власні ідеї. Один запропонував додати попередження про терміни придатності продуктів із можливістю сповіщення коли до закінчення терміну залишається небагато часу, а інший звернув увагу на те, що на деяких пристроях клавіатура перекриває поле введення персонального запиту.

Отримані результати свідчать про те, що застосунок в цілому отримав високі оцінки від користувачів за якістю аналізу, зручністю інтерфейсу та корисністю наданої інформації. Єдиним аспектом, який отримав дещо нижчі оцінки, є швидкість роботи.

ВИСНОВКИ

У рамках кваліфікаційної роботи розроблено і реалізовано мобільний застосунок Android для розпізнавання та пояснення складу харчових продуктів з автоматизованим перекладом українською мовою. Робота охоплює повний цикл створення застосунку, який починається з аналізу існуючих рішень і закінчується реалізацією функціоналу із використанням сучасних інструментів.

Для створення застосунку було вирішено такі завдання:

– проведено аналіз поточного стану проблеми інформованого вибору харчових продуктів, що дало можливість виявити основні труднощі, з якими стикаються споживачі при самостійному розшифруванні складу продуктів;

– проаналізовано існуючі мобільні застосунки для сканування продуктів, оцінено їхню якість аналізу складу та виявлено їхні обмеження щодо роботи з українськими товарами та українською мовою;

– оглянуто сучасні підходи до отримання тексту із зображень та обґрунтовано доцільність використання MLLMs як основного інструменту аналізу;

– сформовано власний набір даних фотографій етикеток харчових продуктів, що охоплює різні типи упаковок, мови написання складу та умови фотографування;

– проведено експериментальну оцінку якості роботи моделей-кандидатів із застосуванням підходу LLM-as-a-Judge для автоматизованого оцінювання, за результатами яких встановлено, що модель Gemini 2.5 Pro з параметром бюджету міркувань, що дорівнює 512, та параметром температури генерації, що дорівнює 0,3, забезпечує найкращий баланс між якістю аналізу, стабільністю та швидкодією;

– проведено експериментальні серію експериментів якості оброблення персональних запитів користувача щодо складу продукту, що підтвердило високу якість роботи обраної моделі у цьому сценарії;

– обґрунтовано вибір технологій для фронтенду та бекенду мобільного застосунку, і обрано для розроблення клієнтської частини React Native, для реалізації серверної частини FastAPI. Визначено архітектуру клієнт-серверної взаємодії, застосунок побудовано за трирівневою клієнт-серверною архітектурою, де фронтенд і бекенд взаємодіють через REST API, а бекенд виступає посередником між застосунком і мовною моделлю, формує запит, передає зображення та повертає структурований результат;

– спроєктовано і реалізовано мобільний застосунок із клієнт-серверною архітектурою та інтеграцією з Gemini 2.5 Pro через API.

Встановлено, що більшість існуючих рішень прив’язані до баз даних штрих-кодів, не підтримують українську мову та не забезпечують детального пояснення інгредієнтів. Розроблений застосунок позбавлений цих обмежень, бо він працює безпосередньо з фотографією упаковки, пояснює кожен інгредієнт із зазначенням його корисних та шкідливих властивостей, підтримує персональний запит користувача та автоматично перекладає склад українською мовою.

У подальшому було б доцільно реалізувати функцію об’єднання однакових інгредієнтів, які зустрічаються у складі продукту кілька разів, щоб результат аналізу був більш компактним і зручним для сприйняття. Також перспективним напрямом є врахування кількості інгредієнта у складі при оцінці його впливу на здоров’я, оскільки інгредієнт присутній у мінімальній кількості не чинить такого ж впливу як основний. Окрім того, розроблення версії застосунку для платформи iOS дозволило б розширити аудиторію користувачів.

Результати роботи апробовано на 3 конференціях: XXX Міжнародному молодіжному форумі «Радіоелектроніка та молодь у XXI столітті» [89], VI Заочній міжнародній науково-практичній конференції «Scientific vector of various sphere’ sevelopment: reality and future trends» (журнал «Grail of Science», № 62, 20.02.2026) [90] та X Міжнародній науково-практичній конференції «Theoretical and empirical scientific research: concept and trends» (збірник «Логос», Oxford, 08.05.2026) [91]. За результатами форуму здобуто друге місце на секції «Комп’ютерний зір та мультимедійні системи».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Healthy diet. *World Health Organization (WHO)*. URL: https://www.who.int/news-room/fact-sheets/detail/healthy-diet (дата звернення 14.12.2025).

2. Food Labelling: A Brief Analysis of European Regulation 1169/2011 - PMC. *PMC Home*. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC5076719/ (дата звернення 14.12.2025).

3. Regulation (EU) No 1169/2011 of the European Parliament and of the Council of 25 October 2011 on the provision of food information to consumers. URL: https://eur-lex.europa.eu/eli/reg/2011/1169/oj (дата звернення 14.12.2025).

4. Що ми їмо – нові правила читання етикетки: аналіз Закону України № 2639-VIII «Про інформацію для споживачів щодо харчових продуктів», – блог Володимира Лапи. URL: https://www.kmu.gov.ua/news/shcho-mi-yimo-novi-pravila-chitannya-etiketki-analiz-zakonu-2639-viii-pro-informaciyu-dlya-spozhivachiv-shchodo-harchovih-produktiv-blog-volodimira-lapi (дата звернення 14.12.2025).

5. Food label reading: Read before you eat - PMC. *PMC Home*. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC5903167/ (дата звернення 14.12.2025).

6. Song J. et al. The understanding, attitude and use of nutrition label among consumers. URL: https://www.aulamedica.es/nh/pdf/8791.pdf (дата звернення 14.12.2025).

7. Consumer Knowledge about Food Labeling and Fraud | MDPI. *MDPI*. URL: https://www.mdpi.com/2304-8158/10/5/1095 (дата звернення 14.12.2025).

8. Право споживача на інформацію про продукцію. URL: https://consumerhm.gov.ua/860-pravo-spozhivacha-na-informatsiyu-pro-produktsiyu (дата звернення 14.12.2025).

9. Sensor Tower - Market Intelligence for the Global Digital Economy. *Sensor Tower*. URL: https://app.sensortower.com (дата звернення 14.12.2025).

10. Top Android Apps and Games on Google Play | AppBrain.com. AppBrain. URL: https://www.appbrain.com (дата звернення 14.12.2025).

11. Foodstr. URL: https://foodstr.app/ua/index_ua/#second-section (дата звернення 14.12.2025).

12. Open Food Facts. URL: https://world.openfoodfacts.org/ (дата звернення 14.12.2025).

13. Yuka - The Mobile App That Scans Your Products. URL: https://yuka.io/en/ (дата звернення 14.12.2025).

14. Nutri-Scan. URL: https://www.nutriscanapp.com/ (дата звернення 14.12.2025).

15. PureCheck. URL: https://play.google.com/store/apps/details?id=com.pixsterstudio.purecheck&hl=uk (дата звернення 14.12.2025).

16. Tasty Healthy – Food Analyzer. URL: https://play.google.com/store/apps/details?id=com.tastewell (дата звернення 14.12.2025).

17. Food Scanner. URL: https://play.google.com/store/apps/details?id=food.scanner.calorie.tracker (дата звернення 14.12.2025).

18. FoodLookup. URL: https://play.google.com/store/apps/details?id=com.shervinkoushan.foodlookup (дата звернення 14.12.2025).

19. Food Label Scanner & Checker. URL: https://play.google.com/store/apps/details?id=com.zengfeiyue.healthscan (дата звернення 14.12.2025).

20. Застосунок Read the Labels – Food Scanner. URL: https://play.google.com/store/apps/details?id=com.readthelabelscanner&hl=uk (дата звернення 14.12.2025).

21. Scanner Alimente. URL: https://play.google.com/store/apps/details?id=com.codingshadows.scanneralimente&hl=uk (дата звернення 14.12.2025).

22. Eat Safe – AI Food Scanner. URL: https://play.google.com/store/apps/details?id=com.appsasap.additivesscanner&hl=uk (дата звернення 14.12.2025).

23. Yakovleva, O., Kovtunenko, A., Liubchenko, V., Honcharenko, V., & Kobylin, O. (2023). Face Detection for Video Surveillance-based Security System. CEUR Workshop Proceedings Vol. 3403. pp. 69-86.

24. Yakovleva, O., Kovač, M., Ardasov, V. & Yeremenko, I. (2023). Study on adding functionality to the Zoom online conference system for monitoring the participant activities. Public Administration and Regional Development, 19(1), pp. 158–183.

25. Yakovleva O., Matúšová S., Tvoroshenko I., Isaiev Y. (2024). Visitor counting based on video stream analysis from surveillance cameras. Scientific Journal of Bratislava University of Economics and Management «Public Administration and Regional Development, Economics, Management and Marketing», vol. 20, no. 1, pp. 67–87.

26. Cherednichenko, O., Yakovleva, O., & Hrechyshkin, D. (2025). *A data-centric approach to crowd counting: Synthetic dataset creation and evaluation*. In S. Vladov, Y. Bodyanskiy, V. Lytvyn, I. Aizenberg, & L. Chyrun (Eds.), *Proceedings of the International Workshop on Applied Intelligent Security Systems in Law Enforcement (AISSLE-2025)* CEUR Workshop Proceedings. pp. 22–34.

27. Yakovleva O., Nebeský L, Liakhov P. (2023). Research methods of texture image analysis to solve the texture search problem. Proceedings of the IV International Scientific and Practical Conference «The world of modern technologies and inventions». Vienna, Austria. 2023. pp. 252-261.

28. Ковтуненко, А. Р., Яковлева, О. В., Любченко, В. А., & Янголенко, О. В. (2020). Дослідження сумісного використання математичної морфології та згорткових нейронних мереж для вирішення задачі розпізнавання цінників. Вісник Національного технічного університету ХПІ (3). 24-31.

29. Yakovleva O., Matúšová S., Liubchenko V., Maksimov H. (2025). Innovative solutions based on AI in education, on the example of generating multimodal lecture notes. Scientific Journal of Bratislava University of Economics and Management «Public Administration and Regional Development, Economics, Management and Marketing», vol. 21, no. 1, pp. 140-163.

30. Гороховатський В., Творошенко І. (2026) Продуктивні моделі аналізу даних у методах розпізнавання зображень: монографія. Харків: ХНУРЕ, 153 с.

31. OpenCV. URL: https://opencv.org (дата звернення 14.12.2025).

32. Python Pillow. URL: https://python-pillow.github.io/ (дата звернення 14.12.2025).

33. Image processing in Python – scikit-image. URL: https://scikit-image.org/ (дата звернення 14.12.2025).

34. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2026) Feature space quantization and application of metrics in structural methods of image recognition, *IEEE Access*, vol. 14, pp. 17812-17824.

35. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2025) Image description compression in classification structural methods, *IEEE Access*, vol. 13, pp. 43631-43641.

36. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, 33(1), pp. 113-125.

37. Gorokhovatskyi, O., & Yakovleva , O. (2024). medoids as a packing of ORB image descriptors. *Advanced Information Systems*, 8(2), 5–11.

38. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, *IEEE Access*, vol. 12, pp. 73376-73385.

39. Yakovleva, O., & Nikolaieva, K. (2020). Research Of Descriptor Based Image Normalization And Comparative Analysis Of SURF, SIFT, BRISK, ORB, KAZE, AKAZE Descriptors. *Advanced Information Systems*, 4(4), 89-101.

40. Tesseract (software) - Wikipedia. URL: https://en.wikipedia.org/wiki/Tesseract_(software) (дата звернення 14.12.2025).

41. GitHub - JaidedAI/EasyOCR: Ready-to-use OCR with 80+ supported languages and all popular writing scripts including Latin, Chinese, Arabic, Devanagari, Cyrillic and etc. URL: https://github.com/JaidedAI/EasyOCR (дата звернення 14.12.2025).

42. Text Detection Module - PaddleOCR Documentation. URL: https://www.paddleocr.ai/main/en/version3.x/module_usage/text_detection.html (дата звернення 14.12.2025).

43. keras-ocr – keras_ocr documentation. URL: https://keras-ocr.readthedocs.io/en/latest/ (дата звернення 14.12.2025).

44. Detect text in images. Cloud Vision API. Google Cloud Documentation. URL: https://docs.cloud.google.com/vision/docs/ocr (дата звернення 14.12.2025).

45. Amazon Textract - Docs by LangChain. URL: https://docs.langchain.com/oss/python/integrations/document_loaders/amazon_textract (дата звернення 14.12.2025).

46. Images and vision | OpenAI API. OpenAI Documentation. URL: https://platform.openai.com/docs/guides/images-vision (дата звернення 14.12.2025).

47. Introducing GPT-5 | OpenAI API. OpenAI Documentation. URL: https://openai.com/index/introducing-gpt-5/ (дата звернення 14.12.2025).

48. OpenAI. Pricing. URL: https://openai.com/api/pricing/

49. Image understanding | Gemini API. *Google AI for Developers*. URL: https://ai.google.dev/gemini-api/docs/image-understanding (дата звернення 14.12.2025).

50. Gemini Developer API pricing | Gemini API. *Google AI for Developers*. URL: https://ai.google.dev/gemini-api/docs/pricing (дата звернення 14.12.2025).

51. Introducing Claude Sonnet 4.5. *Home \ Anthropic*. URL: https://www.anthropic.com/news/claude-sonnet-4-5 (дата звернення 14.12.2025).

52. Your First API Call | DeepSeek API Docs. URL: https://api-docs.deepseek.com/ (дата звернення 14.12.2025).

53. Download Android Studio & App Tools - Android Developers. *Android Developers*. URL: https://developer.android.com/studio (дата звернення 14.12.2025).

54. Kotlin and Android | Android Developers. URL: https://developer.android.com/kotlin (дата звернення 14.12.2025).

55. Swift - Apple Developer. URL: https://developer.apple.com/swift (дата звернення 14.12.2025).

56. Xcode - Apple Developer. URL: https://developer.apple.com/xcode (дата звернення 14.12.2025).

57. React Native. Learn once, write anywhere. URL: https://reactnative.dev (дата звернення 14.12.2025).

58. Flutter - Build apps for any screen. URL: https://flutter.dev/ (дата звернення 14.12.2025).

59. Node.js – Introduction to Node.js. *Node.js – Run JavaScript Everywhere*. URL: https://nodejs.org/en/learn/getting-started/introduction-to-nodejs (дата звернення 14.12.2025).

60. FastAPI. URL: https://fastapi.tiangolo.com/ (дата звернення 14.12.2025).

61. Kaggle Datasets. URL: https://www.kaggle.com/datasets (дата звернення 09.05.2026).

62. Google Dataset Search. URL: https://datasetsearch.research.google.com/ (дата звернення 09.05.2026).

63. Mendeley Data. URL: https://data.mendeley.com/ (дата звернення 09.05.2026).

64. Roboflow Universe. URL: https://universe.roboflow.com/ (дата звернення 09.05.2026).

65. AWS Open Data Registry. URL: https://registry.opendata.aws/ (дата звернення 09.05.2026).

66. Ingredients Image from Food Label. URL: https://www.kaggle.com/datasets/shensivam/ingredients-image-from-food-label (дата звернення 09.05.2026).

67. OpenFoodFacts Ingredient Detection. URL: https://huggingface.co/datasets/openfoodfacts/ingredient-detection (дата звернення 09.05.2026).

68. Food Packaging Recognition. URL: https://universe.roboflow.com/computer-vision-workspace/food-packaging-recognition (дата звернення 09.05.2026).

69. Comp 4102 Project. URL: https://universe.roboflow.com/comp4102-project/comp-4102-project (дата звернення 09.05.2026).

70. Ingredients Label Detection. URL: https://universe.roboflow.com/jainam-doshi-ajldi/ingredients-label-detection (дата звернення 09.05.2026).

71. Ingredients. URL: https://universe.roboflow.com/halal-emjap/ingredients-uyk03 (дата звернення 09.05.2026).

72. Ingredient List Detection. URL: https://universe.roboflow.com/ingredient-list-detection/ingredient-list-detection-htfv6 (дата звернення 09.05.2026).

73. Ingredient Label. URL: https://universe.roboflow.com/stephan-ah1fe/ingredient-label (дата звернення 09.05.2026).

74. FoodRepo. URL: https://www.foodrepo.org/en/products (дата звернення 09.05.2026).

75. Iranian Nutritional Fact Label. URL: https://www.kaggle.com/datasets/gheysar4real/iranian-nutritional-fact-label (дата звернення 09.05.2026).

76. Korean Product Labels Image Dataset. URL: https://huggingface.co/datasets/Kratos-AI/Korean_Product_Labels_Image_Dataset (дата звернення 09.05.2026).

77. Food Labels Dataset. URL: https://data.mendeley.com/datasets/58mfpfxksk/1 (дата звернення 09.05.2026).

78. Best AI for Image Understanding Leaderboard. LLM Stats. URL: https://llm-stats.com/leaderboards/best-ai-for-image-understanding (дата звернення 10.05.2026).

79. Vision and Multimodal Model Analysis. Artificial Analysis. URL: https://artificialanalysis.ai/models/multimodal/vision (дата звернення 10.05.2026).

80. Vision OCR Leaderboard. Arena AI. URL: https://arena.ai/leaderboard/vision/ocr (дата звернення 10.05.2026).

81. Best Multimodal Models. Roboflow Blog. URL: https://blog.roboflow.com/best-multimodal-models/ (дата звернення 10.05.2026).

82. Best Vision Models January 2026. WhatLLM. URL: https://whatllm.org/blog/best-vision-models-january-2026 (дата звернення 10.05.2026).

83. How Open Food Facts Used Open Source LLMs to Enhance Ingredients Extraction. Towards Data Science. URL: https://towardsdatascience.com/how-did-open-food-facts-use-open-source-llms-to-enhance-ingredients-extraction-d74dfe02e0e4/ (дата звернення 10.05.2026).

84. All models | OpenAI API. OpenAI Developers. URL: https://developers.openai.com/api/docs/models/all (дата звернення 10.05.2026).

85. Models overview. Claude API Docs. URL: https://platform.claude.com/docs/en/about-claude/models/overview (дата звернення 10.05.2026).

86. Gemini generateContent API | Google AI for Developers. Google AI for Developers. URL: https://ai.google.dev/gemini-api/docs (дата звернення 10.05.2026).

87. Науменко В.В. (2025). Аналіз можливостей LLMs щодо допомоги у вивченні мови sql. *Радіоелектроніка і молодь у XXI столітті: Тези доповідей 29-го Міжнародного молодіжного форуму* (Харків, 16–19 квітня 2025 р.). Харків: ХНУРЕ. Т. 7, с. 107–109.

88. Yakovleva, O., Matúšová, S., Naumenko, V. (2025, November 11-14) Developing prompts for automated evaluation of SQL scripts with Large Language Models for educational use. Proceedings of the XI International Scientific and Practical Conference «World science: problems, issues and prospects for development». Sofia, Bulgaria. 2025. Pp. 29-38.

89. Соколова А.М., Яковлева О.В. (2026) Проєктування мобільного Android-застосунку для розпізнавання та пояснення складу харчових продуктів з фокусом на українського користувача. *30-й Міжнародний молодіжний форум «Радіоелектроніка і молодь у XXI столітті». Збірник матеріалів форуму, 22–24 квітня 2026 року, Харків, Україна*, Т. 7, С. 156-158.

90. Yakovleva O., Sokolova A. Market Analysis of Mobile Applications for Explaining Food Product Ingredients with a Focus on Ukrainian Users. *Proceedings of the VI Correspondence International Scientific and Practical Conference "Scientific vector of various sphere' development: reality and future trends" in International scientific journal «Grail of Science»*. (February, 2026). Vinnytsia, Ukraine – Vienna, Austria. 2026. Vol. 62. P. 1098-1101.

91. Yakovleva O., Sokolova A., Datsok Y. (2026) Optimization of multimodal Gemini models for food product composition recognition and explanation with automated translation into Ukrainian. Theoretical and empirical scientific research: concept and trends: proceedings of the X international scientific and practical conference, May 8, 2026, Oxford, United Kingdom, pp. 117-120.