Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Система моніторингу харчування на основі аналізу фотозображень страв та формування персоналізованих рекомендацій
(тема)

Виконав:
здобувач другого року навчання,
групи СШІм-24-1
Богдан КІЗІЧЕНКО
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва освітньої програми)
Керівник проф. Євгеній БОДЯНСЬКИЙ
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Лариса ЧАЛА
(підпис) (власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

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

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

Освітня програма Системи штучного інтелекту
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Кізіченку Богдану Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Система моніторингу харчування на основі аналізу фотозображень страв та формування персоналізованих рекомендацій

затверджена наказом університету від 6 квітня 20 26 р. № 269Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 19 травня 20 26 р.

3. Вихідні дані до роботи наукові статті та документацію з області комп'ютерного зору та аналізу харчування, роботи, присвячені методам розпізнавання страв на фотозображеннях та оцінювання нутріційних показників, документацію OpenAI Vision API, Foundation Models, SwiftUI та SwiftData, а також приклади реалізації систем моніторингу харчування на основі штучного інтелекту.

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної галузі та постановка задачі
2) Теоретичні дослідження
3) Проектування системи
4) Програмна реалізація

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	06.04.2026	виконано
2	Визначення проблеми	10.04.2026	виконано
3	Аналіз конкурентних рішень	25.04.2026	виконано
4	Проектування компонентів системи	28.04.2026	виконано
5	Реалізація додатку	01.05.2026	виконано
6	Оформлення пояснювальної записки	09.05.2026	виконано
7	Захист	19.05.2026	виконано

Дата видачі завдання 6 квітня 20 26 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Євгеній БОДЯНСЬКИЙ
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 94 с., 28 рис., 1 табл., 1 дод., 20 джерел.

АНАЛІЗ ЗОБРАЖЕНЬ, КОМП'ЮТЕРНИЙ ЗІР, МОНІТОРИНГ ХАРЧУВАННЯ, ПЕРСОНАЛІЗОВАНІ РЕКОМЕНДАЦІЇ, ШТУЧНИЙ ІНТЕЛЕКТ, APPLE INTELLIGENCE, FOUNDATION MODELS, ON-DEVICE AI, OPENAI VISION API.

Об'єктом дослідження є процес цифрового моніторингу харчування людини з використанням сучасних інформаційних технологій та мобільних програмних систем.

Предметом дослідження є методи аналізу харчових зображень, обробки даних про раціон користувача та формування персоналізованих нутріційних рекомендацій на основі технологій штучного інтелекту.

Метою роботи є розробка мобільного застосунку для інтелектуального моніторингу харчування, який забезпечує автоматичний аналіз фотографій їжі та формування персоналізованих рекомендацій щодо оптимізації раціону.

У роботі використано методи комп'ютерного зору, машинного та глибокого навчання, аналізу зображень, а також підходи до використання мультимодальних моделей штучного інтелекту.

У результаті виконання роботи розроблено програмну реалізацію мобільного застосунку, що забезпечує автоматичне розпізнавання страв за фотографіями, підрахунок калорійності та нутріційних показників, а також інтелектуальний аналіз історії харчування користувача.

Отримані результати можуть бути використані для створення мобільних сервісів моніторингу харчування, систем підтримки здорового способу життя та цифрових інструментів контролю раціону.

ABSTRACT

Master’s thesis contains: 94 pp., 28 fig., 1 tabl., 1 ann., 20 references.

ANALYSIS OF IMAGES, APPLE INTELLIGENCE, ARTIFICIAL INTELLIGENCE, COMPUTER VISION, FOUNDATION MODELS, NUTRITION MONITORING, ON-DEVICE AI, OPENAI VISION API, PERSONALIZED RECOMMENDATIONS.

The object of the research is the process of digital nutrition monitoring using modern information technologies and mobile software systems.

The subject of the research is methods for food image analysis, processing of user dietary data, and generation of personalized nutritional recommendations based on artificial intelligence technologies.

The purpose of the work is to develop a mobile application for intelligent nutrition monitoring that provides automatic analysis of food photographs and generation of personalized recommendations for diet optimization.

The work uses methods of computer vision, machine learning and deep learning, image analysis, as well as approaches to the use of multimodal artificial intelligence models.

As a result of the work, a software implementation of a mobile application was developed that provides automatic food recognition from photographs, calorie and nutritional value calculation, as well as intelligent analysis of the user’s nutrition history.

The obtained results can be used for the development of mobile nutrition monitoring services, healthy lifestyle support systems, and digital diet control tools.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 9
Вступ 10
1 Аналіз предметної галузі та постановка задачі 12
1.1 Цифровий моніторинг харчування 12
1.1.1 Значення контролю харчування у сучасному суспільстві 13
1.1.2 Підходи до оцінювання калорійності та нутріційного складу страв 14
1.1.3 Роль інформаційних технологій у системах контролю раціону 16
1.2 Використання комп'ютерного зору для аналізу харчових продуктів 17
1.2.1 Методи розпізнавання об'єктів на зображеннях 17
1.2.2 Особливості розпізнавання страв на фотозображеннях 18
1.2.3 Використання моделей штучного інтелекту для аналізу харчових зображень 20
1.3 Сучасні цифрові рішення для відстеження харчування 21
1.3.1 Огляд існуючих мобільних застосунків для контролю харчування 22
1.3.2 Порівняльний аналіз функціональних можливостей аналогів . 23
1.3.3 Визначення обмежень існуючих підходів 24
1.4 Визначення вимог до системи моніторингу харчування 25
1.4.1 Визначення цільової аудиторії системи 26
1.4.2 Аналіз основних сценаріїв використання застосунку 27
1.4.3 Функціональні вимоги до системи 27
1.4.4 Нефункціональні вимоги до програмного продукту 28
1.5 Постановка задачі 29
2 Теоретичні дослідження 30
2.1 Аналіз харчових зображень як задача комп'ютерного зору 30
2.1.1 Особливості аналізу зображень харчових продуктів 30

2.1.2 Класифікація задач комп'ютерного зору у системах розпізнавання їжі 31
2.1.3 Основні етапи аналізу фотозображень страв 33
2.2 Методи розпізнавання об'єктів на зображеннях 34
2.2.1 Класичні підходи комп'ютерного зору 35
2.2.2 Методи машинного навчання для аналізу зображень 36
2.2.3 Переваги та обмеження традиційних методів 38
2.3 Глибоке навчання у задачах аналізу харчових зображень 39
2.3.1 Згорткові нейронні мережі у задачах розпізнавання їжі 40
2.3.2 Архітектури нейронних мереж для класифікації зображень 41
2.3.3 Сучасні моделі комп'ютерного зору 42
2.4 Використання OpenAI Vision API для аналізу зображень їжі 43
2.4.1 Принцип роботи мультимодальних моделей 44
2.4.2 Можливості Vision API для аналізу харчових зображень 45
2.4.3 Переваги використання хмарних AI-сервісів для задач розпізнавання їжі 45
2.5 Використання Foundation Models та on-device AI у мобільних застосунках 46
2.5.1 Концепція Apple Intelligence та Foundation Models 46
2.5.2 Особливості використання on-device AI у мобільних системах 47
2.5.3 Застосування генеративних моделей для формування персоналізованих рекомендацій 48
3 Проектування системи 49
3.1 Аналіз вимог до системи моніторингу харчування 49
3.1.1 Функціональні вимоги до мобільного застосунку 49
3.1.2 Нефункціональні вимоги до програмного продукту 50
3.1.3 Технічні вимоги до мобільної платформи 51
3.1.4 Вимоги до користувацького інтерфейсу 52
3.2 Функціональна модель застосунку 53
3.2.1 Модуль аналізу фотографій їжі 55

3.2.2 Модуль збереження та аналізу історії харчування 56
3.2.3 Модуль генерації персоналізованих рекомендацій..... 57
3.3 Модель взаємодії користувача із системою..... 57
3.3.1 Основні ролі користувачів системи..... 59
3.3.2 Основні сценарії використання застосунку 60
3.3.3 Сценарій аналізу фотографії страви 61
3.3.4 Сценарій отримання рекомендацій щодо харчування..... 63
3.4 Архітектура мобільного застосунку 64
3.4.1 Вибір технологічного стеку та його обґрунтування 65
3.4.2 Архітектурний шаблон 67
3.4.3 Інтеграція Vision API для аналізу зображень 68
3.4.4 Використання Foundation Models для формування персоналізованих рекомендацій 69
4 Програмна реалізація 71
4.1 Опис технологічного стеку..... 71
4.2 Архітектура програмної системи..... 72
4.2.1 Загальна структура мобільного застосунку 73
4.2.2 Організація архітектури MVVM..... 75
4.2.3 Взаємодія основних модулів системи 76
4.3 Реалізація моделей даних та локального зберігання..... 77
4.4 Реалізація модуля аналізу фотозображень страв..... 80
4.4.1 Отримання фотозображення прийому їжі 81
4.4.2 Попередня обробка зображення..... 82
4.4.3 Інтеграція OpenAI Vision API для визначення складу страви .. 83
4.4.4 Формування результатів нутріційного аналізу..... 84
4.5 Реалізація користувацького інтерфейсу..... 86
Висновки..... 90
Перелік джерел посилання 92
Додаток А Відомість кваліфікаційної роботи..... 94

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AI – Artificial Intelligence – штучний інтелект;

API – Application Programming Interface – програмний інтерфейс для взаємодії між програмними компонентами;

Base64 – метод кодування двійкових даних у текстовий формат;

BMI – Body Mass Index – індекс маси тіла;

Foundation Models – генеративні моделі штучного інтелекту для створення рекомендацій та аналізу тексту;

GPT – Generative Pre-trained Transformer – генеративна трансформерна модель попереднього навчання;

iOS – операційна система для мобільних пристроїв Apple;

JSON – JavaScript Object Notation – текстовий формат обміну структурованими даними;

JPEG – формат стиснення цифрових зображень;

MVVM – Model – View – ViewModel – архітектурний шаблон побудови програмних застосунків;

OCR – Optical Character Recognition – технологія оптичного розпізнавання тексту;

OpenAI Vision API – сервіс комп’ютерного зору для аналізу фотозображень;

Prompt – текстова інструкція для моделі штучного інтелекту;

SwiftData – фреймворк локального зберігання даних у застосунках Apple;

UI – User Interface – користувацький інтерфейс;

UX – User Experience – користувацький досвід;

Vision API – програмний інтерфейс для аналізу зображень за допомогою моделей комп’ютерного зору.

ВСТУП

Сучасний розвиток інформаційних технологій та методів штучного інтелекту суттєво трансформує підходи до аналізу даних у різних галузях, зокрема у сфері здоров’я та харчування. Одним із ключових напрямів є автоматизація процесів моніторингу раціону людини, що має важливе значення для підтримання здорового способу життя, профілактики захворювань та підвищення якості життя. У цьому контексті особливої актуальності набуває використання методів комп’ютерного зору та машинного навчання для аналізу фотозображень страв і визначення їх нутріційних характеристик.

На сучасному етапі проблема моніторингу харчування вирішується переважно за допомогою мобільних застосунків, які базуються на ручному введенні інформації про спожиті продукти. Такий підхід має низку суттєвих недоліків, серед яких – значні витрати часу користувача, суб’єктивність оцінювання порцій, низька точність визначення калорійності та складу страв. У результаті це знижує ефективність використання подібних систем і обмежує їх практичну цінність. Водночас стрімкий розвиток технологій глибинного навчання, зокрема згорткових нейронних мереж і трансформерів, відкриває нові можливості для автоматизованого розпізнавання об’єктів на зображеннях, включаючи складні сцени з їжею.

Аналіз сучасних наукових досліджень і комерційних рішень показує, що вже існують підходи до визначення типу страви за фотографією, однак більшість із них або обмежуються класифікацією, або не забезпечують достатньої точності оцінювання нутріційних показників. Крім того, значна частина існуючих систем не враховує індивідуальні особливості користувача та не формує персоналізованих рекомендацій щодо харчування. Таким чином, проблема створення інтегрованої системи, яка поєднує автоматичний аналіз зображень, оцінювання харчової цінності та

інтелектуальну генерацію рекомендацій, залишається відкритою і потребує подальших досліджень.

Актуальність даної кваліфікаційної роботи обумовлена необхідністю підвищення ефективності та зручності процесу моніторингу харчування за рахунок використання сучасних методів штучного інтелекту. Використання аналізу фотозображень дозволяє значно спростити взаємодію користувача із системою, мінімізувати людський фактор та підвищити точність оцінювання раціону. Додатково, інтеграція модулів аналітики та рекомендацій забезпечує перехід від простого обліку калорій до інтелектуальної підтримки прийняття рішень щодо харчування. Відповідність теми сучасному стану науки та її спрямованість на вирішення прикладної задачі підтверджує доцільність виконання даної роботи .

Підставою для виконання роботи є необхідність створення інноваційного програмного рішення, що відповідає сучасним вимогам до мобільних застосунків у сфері здоров’я, а також розвиток напрямів комп’ютерного зору, обробки зображень та генеративного штучного інтелекту. Запропонована система базується на використанні сучасних інструментів, зокрема моделей аналізу зображень та мовних моделей для генерації рекомендацій, що дозволяє реалізувати комплексний підхід до розв’язання поставленої задачі .

Можливі сфери застосування результатів роботи включають мобільні застосунки для контролю харчування, системи персоналізованих рекомендацій у сфері здоров’я, телемедичні платформи, фітнес-сервіси, а також освітні та наукові дослідження у галузі комп’ютерного зору та штучного інтелекту.

Таким чином, розробка системи моніторингу харчування на основі аналізу фотозображень страв є актуальною науково-практичною задачею, вирішення якої сприятиме підвищенню ефективності контролю раціону та впровадженню інтелектуальних технологій у сферу здорового способу життя.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Цифровий моніторинг харчування

Цифровий моніторинг харчування є одним із сучасних напрямів розвитку інформаційних систем, спрямованих на підтримку здорового способу життя та контроль харчового раціону людини. Завдяки поширенню мобільних технологій та доступності обчислювальних ресурсів стало можливим автоматизувати процеси фіксації, аналізу та інтерпретації даних про споживання їжі (рисунок 1.1).

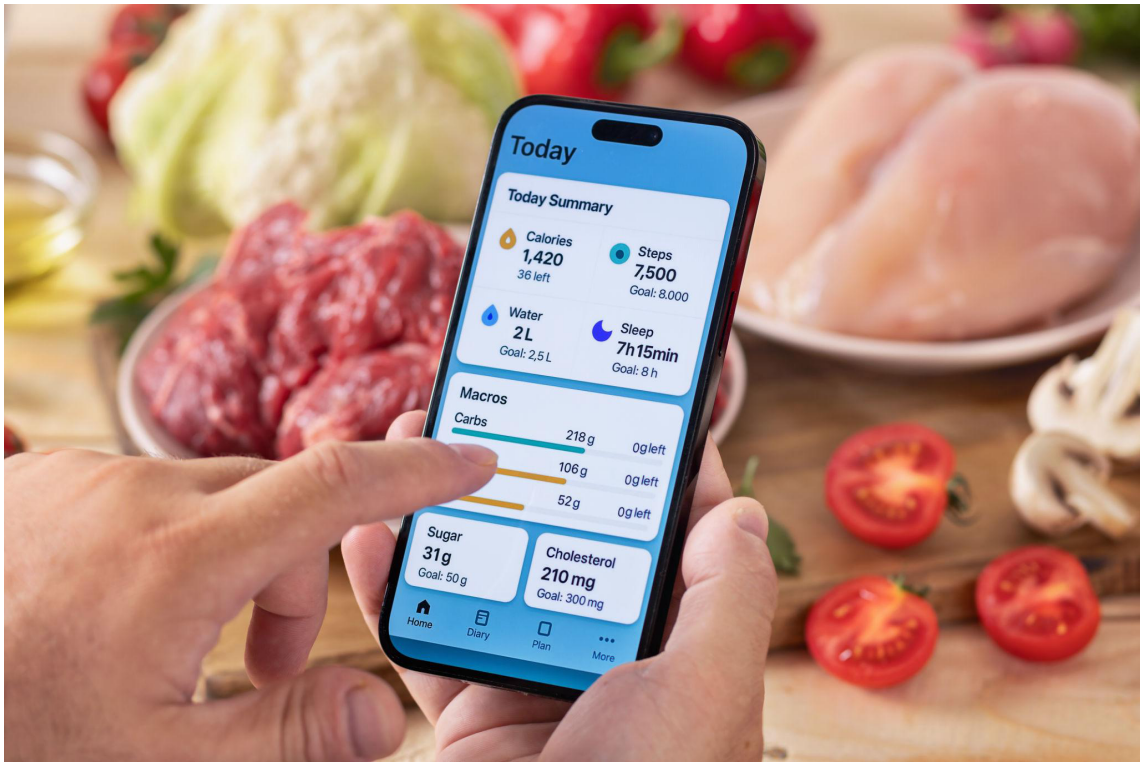


Рисунок 1.1 – Приклад використання мобільного застосунку для відстеження щоденного харчування

Системи моніторингу харчування дозволяють користувачам вести детальний облік спожитих продуктів, оцінювати їх калорійність та визначати співвідношення основних нутрієнтів. Такий підхід сприяє

формуванню усвідомленого ставлення до власного раціону, допомагає контролювати енергетичний баланс та забезпечує підтримку здоров’я.

Цифрові рішення у цій сфері використовують різні методи введення інформації, зокрема ручне введення продуктів, сканування штрихкодів, а також аналіз фотозображень страв. Останній підхід є одним із найбільш перспективних, оскільки дозволяє мінімізувати участь користувача у процесі введення даних та підвищує зручність використання системи.

1.1.1 Значення контролю харчування у сучасному суспільстві

Контроль харчування відіграє важливу роль у підтриманні фізичного здоров’я та профілактиці багатьох захворювань. У сучасному суспільстві значна частина населення стикається з проблемами надмірної або недостатньої ваги, що пов’язано з порушенням балансу між споживанням і витратами енергії. Нераціональне харчування може призводити до розвитку серцево-судинних захворювань, діабету, порушень обміну речовин та інших негативних наслідків.

Одним із ключових аспектів контролю харчування є розуміння складу продуктів та їх енергетичної цінності. Людина повинна враховувати не лише загальну калорійність їжі, але й співвідношення білків, жирів та вуглеводів, а також наявність мікроелементів і вітамінів.

Прикладом наочного представлення принципів збалансованого харчування є модель тарілки здорового раціону, яка демонструє оптимальне співвідношення основних груп продуктів у щоденному меню (рисунок 1.2). У межах такої моделі значна частина раціону повинна складатися з овочів та фруктів. Інша частина відводиться білковим продуктам, таким як м’ясо, риба, яйця та бобові, а також зерновим продуктам, які забезпечують організм необхідною енергією. Окрему роль відіграють корисні жири, що містяться у рослинних оліях, горіхах та інших продуктах.

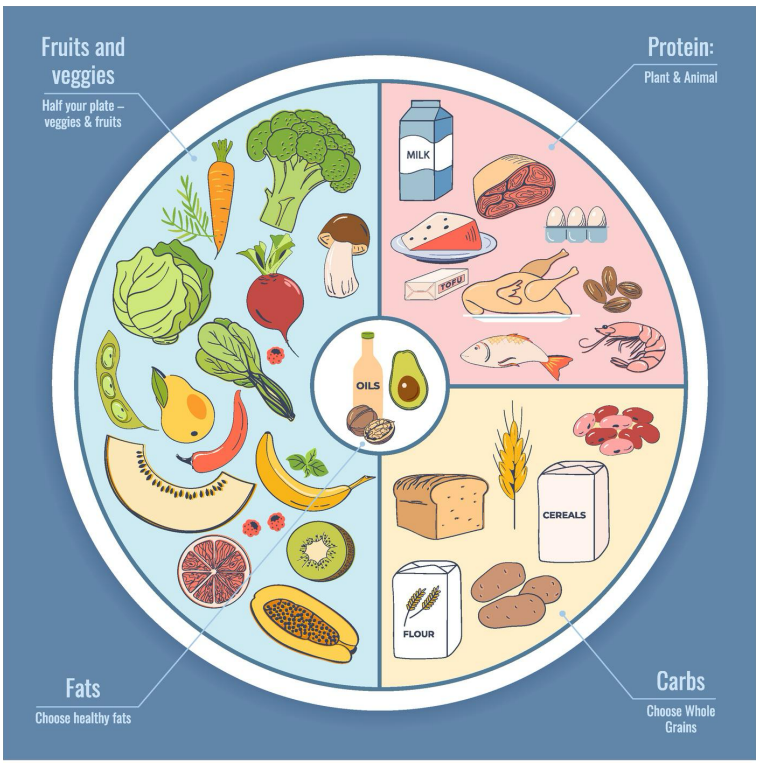


Рисунок 1.2 – Інфографіка про здорове харчування з збалансованою тарілкою, фруктами, овочами, білками та вуглеводами

У зв’язку з підвищенням рівня поінформованості населення щодо здорового способу життя зростає інтерес до інструментів, що допомагають відстежувати споживання їжі. Саме тому системи цифрового моніторингу харчування набувають дедалі більшого поширення серед різних категорій користувачів.

1.1.2 Підходи до оцінювання калорійності та нутріційного складу страв

Існує декілька підходів до визначення калорійності продуктів. Традиційно ця інформація визначається на основі лабораторних досліджень і фіксується у спеціальних таблицях харчової цінності. У подальшому ці дані використовуються для розрахунку калорійності окремих страв шляхом сумування енергетичної цінності інгредієнтів.

Для задач розпізнавання страв на основі зображень використовується значна кількість відкритих наборів даних, які відрізняються за кількістю категорій страв, обсягом зображень та типом представленої кухні. Згідно з таблиці 1.1, одним із ранніх прикладів є набір Pittsburgh Fast-food Image Dataset, що містить 61 категорію фастфуду та 1089 зображень. У подальшому з'явилися більш масштабні набори даних, зокрема UEC-Food 100, який включає 100 категорій страв і приблизно 10 тисяч зображень, а також NTU-FOOD із 50 категоріями та 5000 зображень.

Таблиця 1.1 – Набори даних про харчові продукти, що використовуються як вхідні дані для систем розпізнавання харчових продуктів на основі зображень

Назва	Кількість категорій	Кількість зображень	Кухня
PFID	61	1089	Фастфуд
UEC-Food 100	100	10,000	Японська
NTU-FOOD	50	5000	Багатоетнічна
UNICT-FD889	889	3583	Багатоетнічна
Food-101	101	101,000	Багатоетнічна
UEC-Food 256	256	31,397	Багатоетнічна
Ambient Kitchen	12	1800	Багатоетнічна
UPMC Food-101	101	90,840	Багатоетнічна
Dishes	3832	117,504	Багатоетнічна
Menu-Match	41	646	Азійська, Італійська

Подальший розвиток досліджень у сфері комп'ютерного зору для аналізу харчування призвів до створення більш великих і різноманітних датасетів. Наприклад, набір Food-101 містить понад 101 тисячу зображень, що охоплюють 101 категорію страв, а UEC-Food 256 розширює кількість

категорій до 256 та включає понад 31 тисячу зображень. Окремі набори даних, такі як UNICT-FD889 або Dishes, характеризуються значною кількістю категорій страв і використовуються для навчання моделей, здатних розпізнавати різноманітні типи кухонь.

Також у таблиці представлені датасети, що підтримують зображення з кількома об’єктами їжі на одному знімку, наприклад UEC-Food 100, UEC-Food 256 та Ambient Kitchen. Це дозволяє дослідникам розробляти моделі, здатні визначати декілька страв одночасно, що є важливим для практичного використання систем аналізу харчування. Таким чином, наведені у таблиці набори даних відіграють ключову роль у розробці та вдосконаленні алгоритмів комп’ютерного зору, які використовуються у сучасних системах автоматизованого моніторингу харчування.

1.1.3 Роль інформаційних технологій у системах контролю раціону

Інформаційні технології відіграють ключову роль у розвитку систем контролю харчування. Завдяки використанню сучасних програмних платформ та алгоритмів машинного навчання стало можливим автоматизувати процес аналізу харчових продуктів та отримувати більш точні результати оцінювання раціону.

Одним із перспективних напрямів розвитку таких систем є застосування технологій комп’ютерного зору. Використовуючи алгоритми розпізнавання зображень, система може аналізувати фотографії страв, визначати їх склад та оцінювати приблизну калорійність. Крім того, інформаційні технології дозволяють зберігати історію харчування, візуалізувати статистику споживання нутрієнтів та формувати рекомендації щодо покращення раціону. Інтеграція мобільних застосунків із хмарними сервісами та аналітичними алгоритмами створює передумови для формування персоналізованих систем підтримки здорового харчування.

1.2 Використання комп’ютерного зору для аналізу харчових продуктів

Комп’ютерний зір є одним із напрямів штучного інтелекту, який дозволяє автоматизовано аналізувати та інтерпретувати візуальну інформацію. У сучасних інформаційних системах ця технологія широко використовується для розпізнавання об’єктів на зображеннях, класифікації візуальних даних та виділення важливих характеристик об’єктів. Завдяки розвитку алгоритмів машинного навчання та глибоких нейронних мереж комп’ютерний зір отримав можливість ефективно працювати з великими обсягами зображень і забезпечувати високий рівень точності при визначенні різних типів об’єктів.

У контексті систем моніторингу харчування технології комп’ютерного зору застосовуються для автоматичного аналізу фотографій страв та визначення їхнього складу. Такі системи можуть розпізнавати окремі продукти, визначати типи страв і використовувати отримані дані для подальшого оцінювання нутріційних характеристик. Використання візуального аналізу значно спрощує процес введення інформації для користувача, оскільки замість ручного додавання продуктів достатньо зробити фотографію їжі. Це створює передумови для розробки інтелектуальних застосунків, здатних автоматично вести облік харчування та надавати користувачам аналітичну інформацію про їхній раціон.

1.2.1 Методи розпізнавання об’єктів на зображеннях

Розпізнавання об’єктів на зображеннях є однією з ключових задач комп’ютерного зору. Його основною метою є автоматичне визначення об’єктів, що присутні на зображенні, та їх класифікація за певними категоріями. У ранніх підходах для вирішення цієї задачі використовувалися класичні методи обробки зображень, які базувалися на виділенні характерних ознак. До таких ознак належать контури, текстури,

кольорові характеристики та геометричні параметри об’єктів. Після виділення ознак застосовувалися алгоритми класифікації, зокрема метод опорних векторів або k-найближчих сусідів, що дозволяли віднести об’єкт до певного класу.

Із розвитком машинного навчання значного поширення набули методи, що базуються на використанні згорткових нейронних мереж. Ці моделі здатні автоматично виділяти релевантні ознаки без необхідності ручного визначення характеристик зображення. Згорткові нейронні мережі аналізують зображення на різних рівнях абстракції, починаючи з простих структур, таких як краї або текстури, і завершуючи складнішими формами об’єктів. Завдяки цьому вони демонструють високу ефективність у задачах класифікації та детекції об’єктів на зображеннях.

Для більш складних задач, де необхідно не лише визначити тип об’єкта, але й локалізувати його на зображенні, використовуються методи детекції об’єктів. До найбільш відомих підходів належать алгоритми сімейства R-CNN, а також сучасні моделі, такі як YOLO та SSD. Ці алгоритми дозволяють одночасно визначати декілька об’єктів на одному зображенні та встановлювати їхнє розташування у вигляді обмежувальних рамок. У задачах аналізу харчових продуктів подібні методи застосовуються для виявлення різних компонентів страви, що є важливим етапом подальшого визначення їхньої калорійності та нутріційного складу.

1.2.2 Особливості розпізнавання страв на фотозображеннях

Розпізнавання страв на фотозображеннях є складнішою задачею порівняно з традиційною класифікацією об’єктів, оскільки харчові продукти можуть мати значну варіативність зовнішнього вигляду. Одна й та сама страва може відрізнятися за формою, кольором, способом подачі або складом інгредієнтів залежно від рецептури, умов приготування та індивідуальних особливостей подачі. Ще однією особливістю є значний

вплив умов зйомки на якість розпізнавання. Освітлення, кут фотографування, фонові об’єкти та якість камери можуть суттєво змінювати візуальні характеристики зображення.

У зв’язку з цим для підвищення точності моделей використовуються великі та різноманітні набори навчальних даних, які включають зображення продуктів у різних умовах та варіаціях (рисунок 1.3).



Рисунок 1.3 – Приклади наборів даних, що використовуються для навчання моделей розпізнавання харчових продуктів

Для навчання моделей розпізнавання їжі створюються спеціалізовані датасети, що містять тисячі або навіть сотні тисяч зображень різних страв та продуктів. Такі набори даних охоплюють різноманітні типи кухонь, інгредієнтів та способів подачі їжі. Наявність великої кількості прикладів дозволяє алгоритмам машинного навчання краще узагальнювати інформацію та підвищувати точність класифікації.

1.2.3 Використання моделей штучного інтелекту для аналізу харчових зображень

Сучасні системи аналізу харчових продуктів активно використовують моделі штучного інтелекту, зокрема алгоритми машинного навчання та глибокі нейронні мережі. Такі моделі здатні автоматично обробляти зображення страв, визначати наявні на них продукти та виконувати їх класифікацію. Завдяки навчанню на великих наборах даних алгоритми можуть виявляти складні візуальні закономірності, що дозволяє підвищити точність розпізнавання різних типів їжі. Використання штучного інтелекту дає змогу автоматизувати процес аналізу харчування та значно спростити взаємодію користувача із системою.

Типовий процес аналізу харчового зображення складається з декількох послідовних етапів. Спочатку система виконує сегментацію зображення, під час якої визначаються окремі області, що відповідають різним компонентам страви. Після цього здійснюється класифікація знайдених об’єктів, яка дозволяє визначити тип кожного продукту або інгредієнта. Такий підхід забезпечує більш точний аналіз складних страв, що містять декілька різних продуктів (рисунок 1.4). Після етапу класифікації може виконуватися оцінювання кількісних характеристик продуктів, зокрема визначення приблизного об’єму або маси компонентів страви.

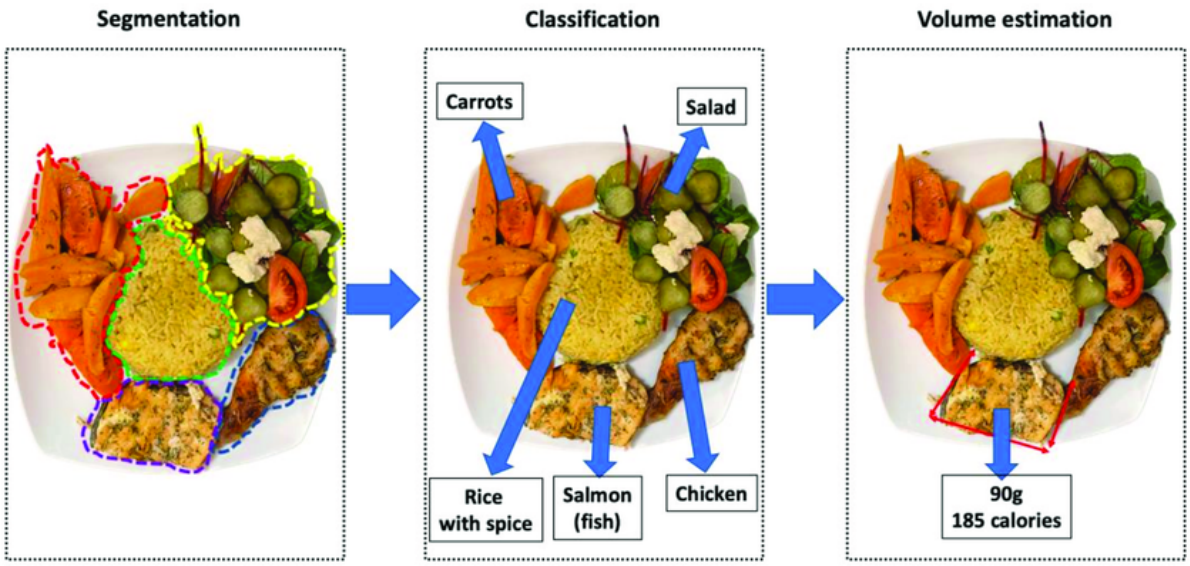


Рисунок 1.4 – Етапи аналізу страви на зображенні у системах розпізнавання їжі

На основі цих даних система здатна розраховувати орієнтовну калорійність та нутріційний склад їжі. Як зображено вище, поєднання етапів сегментації, класифікації та оцінювання об’єму дозволяє створити комплексний підхід до автоматичного аналізу харчових зображень, що є основою для реалізації сучасних систем моніторингу харчування.

1.3 Сучасні цифрові рішення для відстеження харчування

Сучасний розвиток цифрових технологій призвів до появи великої кількості мобільних застосунків, призначених для контролю харчування та аналізу раціону користувачів. Значна частина таких систем орієнтована на підрахунок калорійності спожитої їжі, ведення журналу прийомів їжі та відстеження основних нутріційних показників. Останніми роками особливої популярності набули рішення, які використовують методи комп’ютерного зору для автоматичного аналізу фотозображень страв. Такий підхід дозволяє суттєво спростити процес введення даних, оскільки користувачу

достатньо зробити фотографію їжі, після чого система автоматично визначає її склад та приблизну харчову цінність.

Поява подібних технологій стала можливою завдяки розвитку методів глибокого навчання та моделей розпізнавання зображень, які здатні ідентифікувати різні типи харчових продуктів та оцінювати їхні нутріційні характеристики. У результаті на ринку з’явилися спеціалізовані мобільні застосунки, що поєднують функції трекінгу харчування з автоматичним аналізом фотографій їжі. Аналіз таких рішень є важливим етапом дослідження, оскільки дозволяє визначити їхні функціональні можливості, переваги та обмеження, а також сформувати вимоги до нових систем моніторингу харчування, які можуть забезпечити більш інтелектуальний підхід до аналізу раціону користувача .

1.3.1 Огляд існуючих мобільних застосунків для контролю харчування

Одним із найбільш відомих мобільних застосунків, що використовує аналіз фотозображень для оцінювання харчування, є Cal AI. Даний застосунок дозволяє користувачу фотографувати страву, після чого система за допомогою алгоритмів комп’ютерного зору автоматично визначає калорійність та приблизний нутріційний склад їжі. Основною перевагою такого підходу є відсутність необхідності ручного введення продуктів, що значно спрощує процес ведення щоденника харчування.

Іншим прикладом подібного рішення є застосунок Foodvisor, який також використовує технології штучного інтелекту для розпізнавання їжі на фотографіях. Система аналізує зображення страви, визначає її складові компоненти та на основі бази даних харчових продуктів оцінює калорійність і співвідношення основних нутрієнтів. Крім того, застосунок дозволяє користувачеві переглядати історію харчування та отримувати базову статистику щодо споживання калорій.

Ще одним представником подібних систем є BiteSnap, який орієнтований на швидке розпізнавання продуктів за допомогою фотографії. Після створення зображення алгоритми штучного інтелекту визначають тип страви або продукту та автоматично додають відповідний запис до журналу харчування. Особливістю цього застосунку є велика база даних продуктів та відносно швидкий процес розпізнавання, що дозволяє користувачу швидко фіксувати прийоми їжі.

Також варто згадати застосунок SnapCalorie, який використовує методи глибокого навчання для аналізу фотографій їжі та визначення її калорійності. Система намагається оцінювати склад страви та приблизний об’єм продуктів на тарілці, що дозволяє отримувати більш точні результати підрахунку калорійності. Подібно до інших застосунків цієї категорії, SnapCalorie формує журнал прийомів їжі та відображає базову інформацію про нутріційні показники раціону користувача.

Таким чином, сучасні мобільні застосунки для аналізу харчування все частіше використовують методи комп’ютерного зору для автоматизації процесу фіксації прийомів їжі. Вони дозволяють швидко визначати калорійність та нутріційні характеристики страв на основі фотографій, що робить процес контролю раціону більш зручним для користувача. Однак функціональність більшості таких систем обмежується лише фіксацією та відображенням даних про спожиту їжу.

1.3.2 Порівняльний аналіз функціональних можливостей аналогів

Проведений аналіз сучасних мобільних застосунків для відстеження харчування, що використовують аналіз фотозображень їжі, дозволяє визначити основні функціональні можливості подібних систем. Більшість таких застосунків орієнтовані на автоматизацію процесу фіксації прийомів їжі шляхом використання технологій комп’ютерного зору. Користувачеві достатньо зробити фотографію страви, після чого система виконує її

розпізнавання та визначає приблизну калорійність, а також основні нутріційні показники, зокрема кількість білків, жирів і вуглеводів.

Спільною функціональною характеристикою більшості розглянутих застосунків є ведення журналу харчування, у якому зберігається інформація про всі зафіксовані прийоми їжі. У межах такого журналу користувач може переглядати історію споживання калорій та нутрієнтів, а також отримувати базову статистику щодо свого раціону. Деякі системи також надають можливість редагування результатів розпізнавання або ручного додавання продуктів у випадках, коли автоматичний аналіз працює неточно.

Крім того, окремі застосунки пропонують додаткові функції, пов’язані з візуалізацією даних, зокрема графічне відображення калорійності раціону або співвідношення нутрієнтів протягом дня. Такі інструменти дозволяють користувачеві отримувати загальне уявлення про структуру власного харчування. Водночас аналіз функціональних можливостей існуючих рішень свідчить про те, що основна увага приділяється саме збору та відображенню даних, тоді як глибокий інтелектуальний аналіз раціону користувача реалізований обмежено.

1.3.3 Визначення обмежень існуючих підходів

Аналіз сучасних мобільних застосунків для відстеження харчування, що використовують розпізнавання їжі на основі фотографій, дозволяє виявити низку суттєвих обмежень існуючих підходів. Незважаючи на використання методів комп’ютерного зору та штучного інтелекту, більшість таких систем виконують лише базові функції, пов’язані з фіксацією прийомів їжі та визначенням приблизної калорійності страв. У більшості випадків алгоритми розпізнавання визначають лише тип страви або окремі інгредієнти, після чого система підбирає відповідні значення з попередньо сформованої бази даних продуктів. Це може призводити до неточностей у

визначенні нутріційного складу, особливо у випадках складних або комбінованих страв.

Ще одним обмеженням існуючих систем є недостатній рівень аналітичних можливостей. Більшість застосунків зосереджені на відображенні статистичної інформації, такої як загальна калорійність раціону, кількість спожитих білків, жирів і вуглеводів, а також історія прийомів їжі. При цьому отримані дані переважно використовуються лише для інформування користувача, без проведення глибшого аналізу його харчових звичок або формування рекомендацій щодо оптимізації раціону.

Крім того, значна частина розглянутих застосунків не враховує індивідуальні цілі користувача, пов’язані з харчуванням. Такі цілі можуть включати зниження маси тіла, підтримання поточної ваги, набір маси або формування більш збалансованого раціону. У більшості випадків користувач отримує лише інформацію про спожиті калорії та нутрієнти, проте система не пропонує конкретних рекомендацій щодо коригування харчування.

Таким чином, існуючі рішення переважно виконують функцію цифрового журналу харчування, який фіксує та відображає інформацію про прийоми їжі, але не забезпечує повноцінного інтелектуального аналізу раціону користувача. Відсутність модулів персоналізованих рекомендацій, що враховують історію харчування та поставлені цілі користувача, визначає необхідність розроблення нових систем моніторингу харчування, здатних не лише збирати дані, але й формувати обґрунтовані рекомендації щодо оптимізації раціону.

1.4 Визначення вимог до системи моніторингу харчування

Аналіз існуючих цифрових рішень для контролю харчування показав, що більшість сучасних мобільних застосунків зосереджені переважно на фіксації прийомів їжі та відображенні статистичних даних щодо спожитих

калорій і нутрієнтів. Незважаючи на використання технологій комп’ютерного зору для автоматичного розпізнавання страв, функціональність таких систем здебільшого обмежується збором та візуалізацією інформації про харчування користувача. У результаті користувач отримує лише числові показники раціону, але не має інструментів для їх глибшого аналізу або отримання рекомендацій щодо оптимізації харчування.

З огляду на виявлені обмеження існуючих підходів виникає необхідність розроблення системи моніторингу харчування, яка поєднує автоматичний аналіз фотозображень страв із інтелектуальним аналізом історії раціону користувача. Така система повинна не лише визначати калорійність і нутріційні показники їжі, але й формувати персоналізовані рекомендації на основі накопичених даних та поставлених користувачем цілей. Для цього необхідно визначити основні вимоги до розроблюваного програмного продукту, які охоплюють цільову аудиторію, сценарії використання, а також функціональні й нефункціональні характеристики системи.

1.4.1 Визначення цільової аудиторії системи

Цільовою аудиторією системи моніторингу харчування є користувачі, які прагнуть контролювати власний раціон та отримувати інформацію про харчову цінність спожитої їжі. До цієї категорії можуть належати люди, які дотримуються певних дієт, займаються спортом або прагнуть підтримувати здоровий спосіб життя. Для таких користувачів важливо мати зручний інструмент, що дозволяє швидко фіксувати прийоми їжі та отримувати інформацію про калорійність і нутріційний склад страв.

Окрему групу цільової аудиторії становлять користувачі, які мають конкретні цілі щодо харчування, наприклад зниження маси тіла, підтримання поточної ваги або набір м’язової маси. Для них важливо не

лише відстежувати кількість спожитих калорій, але й контролювати баланс білків, жирів і вуглеводів у раціоні. У цьому випадку система повинна виступати не тільки як інструмент фіксації прийомів їжі, але й як помічник у формуванні більш збалансованого раціону.

1.4.2 Аналіз основних сценаріїв використання застосунку

Одним із основних сценаріїв використання застосунку є фіксація прийому їжі за допомогою фотографування страви. Користувач робить знімок їжі через інтерфейс мобільного застосунку, після чого система автоматично аналізує зображення та визначає основні нутріційні показники. Отримана інформація додається до журналу харчування, де користувач може переглянути дані про калорійність та співвідношення нутрієнтів.

Іншим важливим сценарієм є перегляд історії харчування та аналіз раціону за певний період часу. Користувач може переглядати дані про споживання калорій і нутрієнтів за день або за тиждень, що дозволяє оцінити загальну структуру власного харчування. На основі накопичених даних система виконує аналіз раціону користувача та формує персоналізовані рекомендації щодо його оптимізації.

Також можливим сценарієм використання є отримання рекомендацій щодо харчування з урахуванням поставлених цілей. Користувач обирає відповідну мету, наприклад зниження маси тіла або підтримання збалансованого раціону, після чого система аналізує історію харчування та надає рекомендації щодо коригування споживання калорій або співвідношення нутрієнтів.

1.4.3 Функціональні вимоги до системи

Функціональні вимоги визначають основні можливості програмної системи, які повинні бути реалізовані для забезпечення її ефективної

роботи. Однією з ключових функцій системи є можливість фіксації прийомів їжі за допомогою фотографування страви. Застосунок повинен надавати користувачу зручний інтерфейс для створення фотографій та їх подальшого аналізу.

Система повинна забезпечувати автоматичний аналіз фотозображень їжі з використанням методів комп'ютерного зору. На основі отриманого зображення алгоритми повинні визначати калорійність страви, а також кількість основних нутрієнтів, зокрема білків, жирів і вуглеводів. Результати аналізу повинні зберігатися у базі даних історії харчування користувача.

Ще однією важливою функцією системи є можливість перегляду історії прийомів їжі та аналізу раціону за визначений період часу. Система повинна надавати інструменти для перегляду статистики харчування та візуалізації основних нутріційних показників. Крім того, застосунок повинен включати модуль інтелектуального аналізу, який на основі накопичених даних формує персоналізовані рекомендації щодо оптимізації раціону користувача.

1.4.4 Нефункціональні вимоги до програмного продукту

Нефункціональні вимоги визначають характеристики програмного продукту, які забезпечують його ефективність, надійність та зручність використання. Однією з важливих вимог є зручність користувацького інтерфейсу. Застосунок повинен мати інтуїтивно зрозумілий інтерфейс, що дозволяє користувачеві швидко виконувати основні дії, такі як фотографування їжі або перегляд історії харчування.

Також важливою вимогою є швидкодія системи. Аналіз зображень та отримання результатів повинні виконуватися за мінімальний час, щоб забезпечити комфортне використання застосунку. Крім того, система

повинна забезпечувати стабільну роботу при обробці великої кількості даних, пов’язаних із історією харчування користувача.

Ще одним важливим аспектом є надійність та безпека збереження даних. Інформація про харчування користувача повинна зберігатися у захищеному середовищі та бути доступною лише для власника облікового запису. Також система повинна забезпечувати коректну роботу всіх функціональних модулів та стійкість до можливих помилок під час обробки даних.

1.5 Постановка задачі

На основі проведеного аналізу предметної галузі, сучасних технологій комп’ютерного зору та існуючих цифрових рішень для відстеження харчування можна сформулювати задачу дослідження.

Основною метою роботи є розроблення системи моніторингу харчування, яка використовує аналіз фотозображень страв для автоматичного визначення калорійності та нутріційного складу їжі, а також виконує інтелектуальний аналіз історії харчування користувача.

Система повинна забезпечувати автоматичну фіксацію прийомів їжі, накопичення даних про раціон за визначений період часу та формування персоналізованих рекомендацій щодо оптимізації харчування відповідно до поставлених користувачем цілей. Реалізація такої системи дозволить підвищити зручність контролю раціону та забезпечити більш інтелектуальний підхід до аналізу харчування.

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

2.1 Аналіз харчових зображень як задача комп’ютерного зору

Аналіз зображень є одним із ключових напрямів комп’ютерного зору, що передбачає автоматичну обробку та інтерпретацію візуальної інформації за допомогою алгоритмів штучного інтелекту. Основною метою таких систем є виявлення об’єктів на зображенні, визначення їхніх характеристик та подальша класифікація.

Розпізнавання їжі на фотозображеннях є складною задачею комп’ютерного зору, оскільки харчові продукти можуть мати різноманітну форму, текстуру, колір та спосіб подачі. Крім того, одна страва часто складається з кількох інгредієнтів, які можуть перекривати один одного або бути частково прихованими. Для вирішення таких задач використовуються алгоритми обробки зображень та методи машинного навчання, здатні виділяти характерні ознаки об’єктів і визначати їх належність до певних категорій. Результати такого аналізу можуть бути використані для подальшого визначення нутріційних показників страви та формування інформації про калорійність і склад раціону користувача.

2.1.1 Особливості аналізу зображень харчових продуктів

Аналіз зображень харчових продуктів (рисунок 2.1) є окремим напрямом комп’ютерного зору, що передбачає автоматичне визначення типу їжі або її складових елементів на основі фотозображень. Основною метою такого аналізу є ідентифікація продуктів або страв, які присутні на зображенні, а також визначення їхніх характеристик, що можуть бути використані для подальшої оцінки харчової цінності. Для цього застосовуються алгоритми обробки зображень та моделі машинного

навчання, які здатні аналізувати форму, колір, текстуру та інші візуальні ознаки об’єктів.

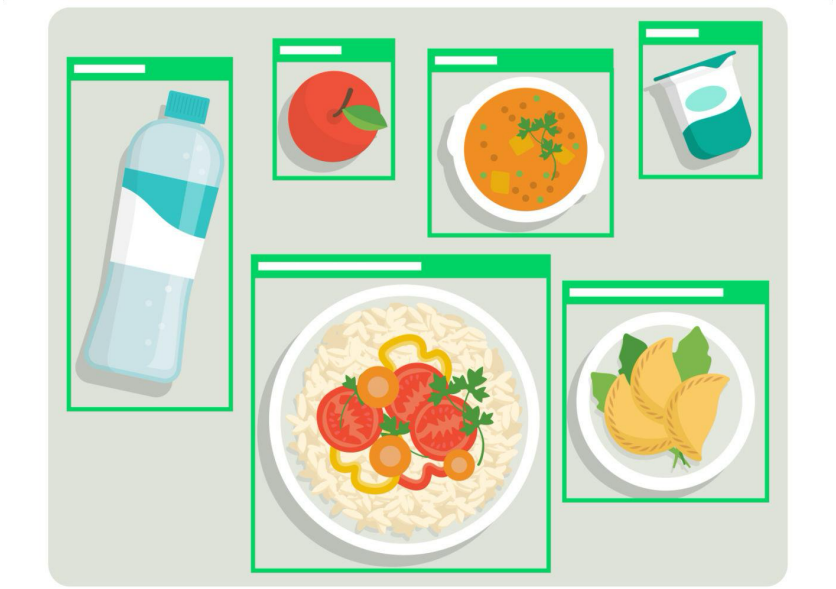


Рисунок 2.1 – Приклад виділення об’єктів на зображенні

Особливістю аналізу харчових зображень є значна варіативність вигляду страв. Один і той самий продукт може мати різний вигляд залежно від способу приготування, освітлення, кута зйомки або подачі. Крім того, страви часто складаються з кількох інгредієнтів, які можуть частково перекривати один одного. У зв’язку з цим системи комп’ютерного зору повинні виконувати не лише класифікацію зображення, але й визначати окремі об’єкти на ньому.

2.1.2 Класифікація задач комп’ютерного зору у системах розпізнавання їжі

У системах комп’ютерного зору, що використовуються для аналізу зображень їжі, можна виділити кілька основних типів задач, які визначають спосіб обробки візуальної інформації. Найпоширенішими серед них є

класифікація зображень, детекція об’єктів та сегментація (рисунок 2.2). Кожна з цих задач має різний рівень складності та дозволяє отримувати різну кількість інформації про об’єкти, присутні на зображенні. Вибір конкретного підходу залежить від вимог системи та необхідної точності аналізу.

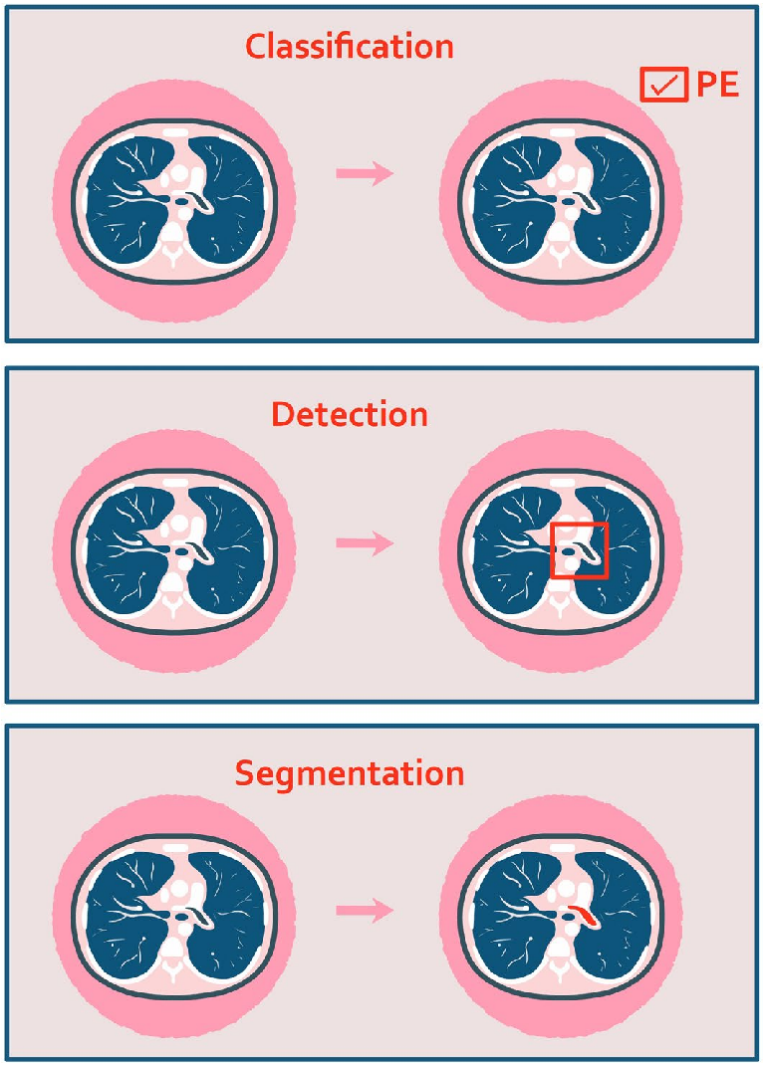


Рисунок 2.2 – Основні типи задач комп’ютерного зору

Задача класифікації передбачає визначення загального класу об’єкта або сцени на зображенні. У випадку систем розпізнавання їжі це може бути визначення типу страви, наприклад салат, суп або паста. Детекція об’єктів є більш складною задачею, оскільки вона передбачає не лише визначення

типу об’єкта, але й локалізацію його положення на зображенні за допомогою спеціальних рамок. Такий підхід дозволяє ідентифікувати кілька різних продуктів на одній фотографії. Найбільш деталізованим методом є сегментація, яка передбачає точне виділення меж кожного об’єкта на зображенні. Це дає можливість визначити форму та площу кожного інгредієнта, що особливо важливо для задач оцінювання порції та калорійності страв.

2.1.3 Основні етапи аналізу фотозображень страв

Аналіз фотозображень страв у системах комп’ютерного зору зазвичай складається з кількох послідовних етапів обробки даних. Сучасні системи аналізу їжі застосовують комплекс алгоритмів обробки зображень, машинного навчання та глибоких нейронних мереж для поступового виділення необхідної інформації з фотозображення (рисунок 2.3).

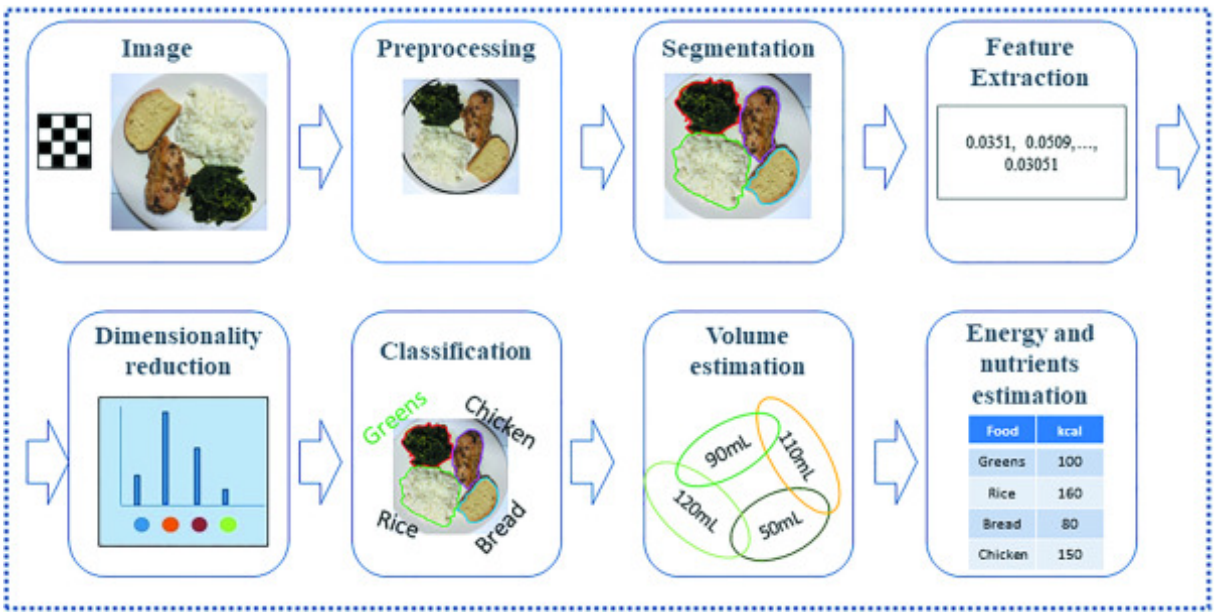


Рисунок 2.3 – Архітектура системи розпізнавання харчових продуктів для нутриційного аналізу

На першому етапі виконується попередня обробка зображення, яка включає нормалізацію розміру, корекцію освітлення та підвищення якості зображення. Це дозволяє підготувати фотографію до подальшого аналізу. Після цього виконується сегментація, під час якої система визначає області зображення, що відповідають окремим продуктам або інгредієнтам страви. На основі виділених областей здійснюється виділення ознак, тобто визначення характеристик зображення, які можуть бути використані для подальшої класифікації об’єктів.

Наступним етапом є класифікація продуктів, де алгоритми машинного навчання визначають тип кожного інгредієнта або страви на зображенні. Після цього система може виконувати оцінювання об’єму або порції продуктів, що дозволяє більш точно визначити їхню харчову цінність. Завершальним етапом є розрахунок калорійності та нутріційних показників, зокрема кількості білків, жирів і вуглеводів.

2.2 Методи розпізнавання об’єктів на зображеннях

Розпізнавання об’єктів на зображеннях є однією з основних задач комп’ютерного зору, яка передбачає автоматичне визначення та класифікацію об’єктів, присутніх на зображенні. У межах цієї задачі система повинна аналізувати візуальну інформацію та визначати, які саме об’єкти зображені на фотографії. Для цього використовуються різні методи обробки зображень та алгоритми машинного навчання, здатні виділяти характерні ознаки об’єктів і визначати їх належність до певних категорій. У системах аналізу харчових продуктів ці методи дозволяють визначати окремі інгредієнти страви, а також класифікувати типи продуктів на фотографії.

Історично перші підходи до розпізнавання зображень базувалися на класичних алгоритмах комп’ютерного зору, які використовували ручне виділення ознак, таких як контури, текстури або ключові точки. Ці методи

дозволяли аналізувати структуру зображення та виконувати його подальшу класифікацію за допомогою статистичних алгоритмів. Однак ефективність таких підходів значною мірою залежала від якості зображення та складності сцени, що обмежувало їх застосування у задачах, пов’язаних із розпізнаванням складних об’єктів.

2.2.1 Класичні підходи комп’ютерного зору

Класичні підходи комп’ютерного зору ґрунтуються на використанні алгоритмів обробки зображень, які дозволяють виділяти характерні ознаки об’єктів. Основною ідеєю таких методів є аналіз окремих візуальних характеристик зображення, зокрема контурів, текстур, границь та геометричних форм. На основі цих характеристик система формує опис зображення, який може бути використаний для подальшого розпізнавання або класифікації об’єктів. Подібні методи активно використовувалися на ранніх етапах розвитку комп’ютерного зору та стали основою для створення перших систем автоматичного аналізу зображень.

Одним із найпоширеніших підходів у класичному комп’ютерному зорі є використання алгоритмів виділення контурів. Такі алгоритми дозволяють визначати межі об’єктів на зображенні шляхом аналізу різких змін яскравості пікселів. Найвідомішими прикладами є оператори Sobel, Prewitt та Canny, які застосовуються для виявлення границь об’єктів. Виділення контурів дозволяє спростити структуру зображення та зосередитися на ключових елементах сцени, що може бути корисним для подальшого аналізу. Приклад роботи алгоритмів виділення контурів представлено нижче (рисунок 2.4). На ньому показано, як початкове зображення може бути перетворене у карту границь, що відображає структуру об’єктів на зображенні.

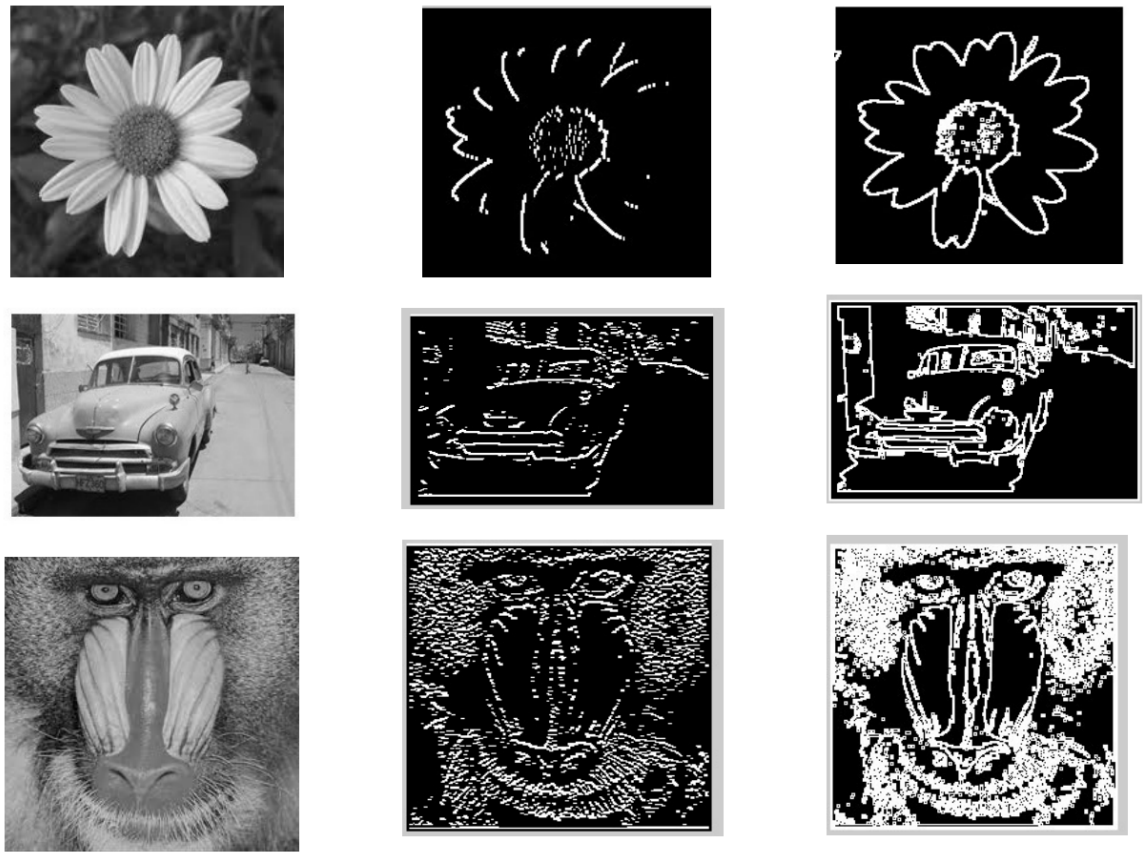


Рисунок 2.4 – Приклади виділення контурів об’єктів на зображенні за допомогою алгоритмів обробки зображень

2.2.2 Методи машинного навчання для аналізу зображень

Методи машинного навчання стали важливим етапом розвитку систем комп’ютерного зору, оскільки дозволили автоматизувати процес розпізнавання об’єктів на зображеннях. На відміну від класичних алгоритмів, які базуються на ручному виділенні ознак, підходи машинного навчання використовують навчальні дані для побудови моделей, здатних самостійно визначати характерні особливості об’єктів. У межах такого підходу система аналізує велику кількість прикладів зображень та формує модель, яка може використовувати отримані знання для класифікації нових зображень.

У традиційних методах машинного навчання процес розпізнавання зображень складається з двох основних етапів: виділення ознак та класифікації. На першому етапі зображення перетворюється у набір числових характеристик, що описують форму, текстуру або інші властивості об’єктів. На другому етапі ці ознаки використовуються для навчання класифікатора, який визначає, до якого класу належить об’єкт. Для цього можуть застосовуватися різні алгоритми машинного навчання, зокрема метод опорних векторів, дерева рішень або байєсівські класифікатори.

Порівняння класичних підходів машинного навчання та сучасних методів глибокого навчання зображено нижче (рисунок 2.5). Як показано на схемі, у традиційних підходах виділення ознак виконується вручну або за допомогою спеціальних алгоритмів, тоді як у моделях глибокого навчання цей процес автоматизований і виконується безпосередньо нейронною мережею під час навчання. Такий підхід дозволяє підвищити точність розпізнавання та значно розширити можливості систем аналізу зображень.

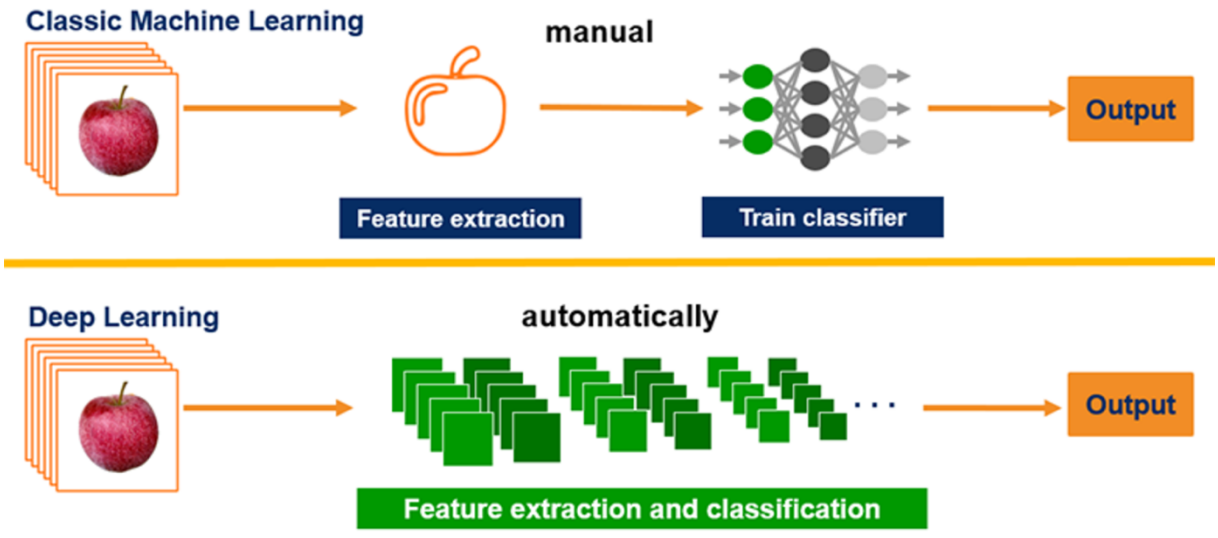


Рисунок 2.5 – Порівняння класичних методів машинного навчання та підходів глибокого навчання у задачах аналізу зображень

2.2.3 Переваги та обмеження традиційних методів

Традиційні методи комп’ютерного зору та машинного навчання мають низку переваг, зокрема відносно просту реалізацію, невисокі обчислювальні вимоги та зрозумілу структуру алгоритмів. Такі підходи дозволяють ефективно вирішувати задачі розпізнавання у випадках, коли об’єкти мають чітко визначені ознаки та зображення отримані у контрольованих умовах. Крім того, традиційні методи часто потребують менших обсягів навчальних даних порівняно з сучасними моделями глибокого навчання.

Водночас ці підходи мають суттєві обмеження. Основною проблемою є необхідність ручного виділення ознак, що ускладнює побудову універсальних моделей для складних задач комп’ютерного зору. Крім того, ефективність таких методів значною мірою залежить від якості зображення, освітлення та інших умов зйомки.

Традиційні підходи (рисунок 2.6) можуть мати проблеми з масштабованістю, гнучкістю алгоритмів, швидкістю обробки та вартістю розробки складних систем. У зв’язку з цим сучасні системи аналізу зображень все частіше використовують методи глибокого навчання, які здатні автоматично визначати ознаки об’єктів та забезпечувати більш високу точність розпізнавання.



Рисунок 2.6 – Обмеження традиційних методів комп’ютерного зору

2.3 Глибоке навчання у задачах аналізу харчових зображень

З розвитком технологій штучного інтелекту значного поширення у задачах комп’ютерного зору набули методи глибокого навчання. На відміну від традиційних алгоритмів машинного навчання, які потребують ручного виділення ознак, моделі глибокого навчання здатні автоматично визначати важливі характеристики зображень у процесі навчання. Завдяки цьому такі моделі можуть ефективно аналізувати складні візуальні сцени та досягати високої точності розпізнавання об’єктів. У задачах аналізу харчових зображень глибоке навчання дозволяє визначати типи продуктів, класифікувати страви та ідентифікувати окремі інгредієнти на фотографіях.

Основою більшості сучасних систем аналізу зображень є нейронні мережі, здатні обробляти великі обсяги візуальних даних. Такі моделі навчаються на великих наборах зображень і поступово формують внутрішні представлення об’єктів, що дозволяє їм розпізнавати нові зображення з високою точністю. Особливо ефективними у задачах комп’ютерного зору є згорткові нейронні мережі, які використовують спеціальні згорткові шари для виділення локальних ознак зображення. Ці ознаки можуть описувати форму, текстуру або структуру об’єктів, що дозволяє мережі формувати ієрархічне представлення візуальної інформації.

У сучасних дослідженнях також активно використовуються нові архітектури нейронних мереж, які забезпечують більш ефективну обробку зображень. До таких моделей належать глибокі згорткові мережі та трансформерні архітектури, здатні аналізувати глобальні залежності між елементами зображення. Використання таких підходів дозволяє значно підвищити точність розпізнавання харчових продуктів, що робить їх перспективними для застосування у мобільних системах моніторингу харчування.

2.3.1 Згорткові нейронні мережі у задачах розпізнавання їжі

Згорткові нейронні мережі (рисунок 2.7) є одним із найпоширеніших типів глибоких нейронних мереж, що використовуються у задачах комп'ютерного зору. Основною особливістю таких мереж є використання згорткових шарів, які дозволяють автоматично виділяти важливі ознаки зображення. На відміну від традиційних методів машинного навчання, де ознаки визначаються вручну, згорткові нейронні мережі здатні самостійно формувати ієрархію ознак у процесі навчання. Це дозволяє ефективно аналізувати складні візуальні структури та підвищувати точність розпізнавання об'єктів.

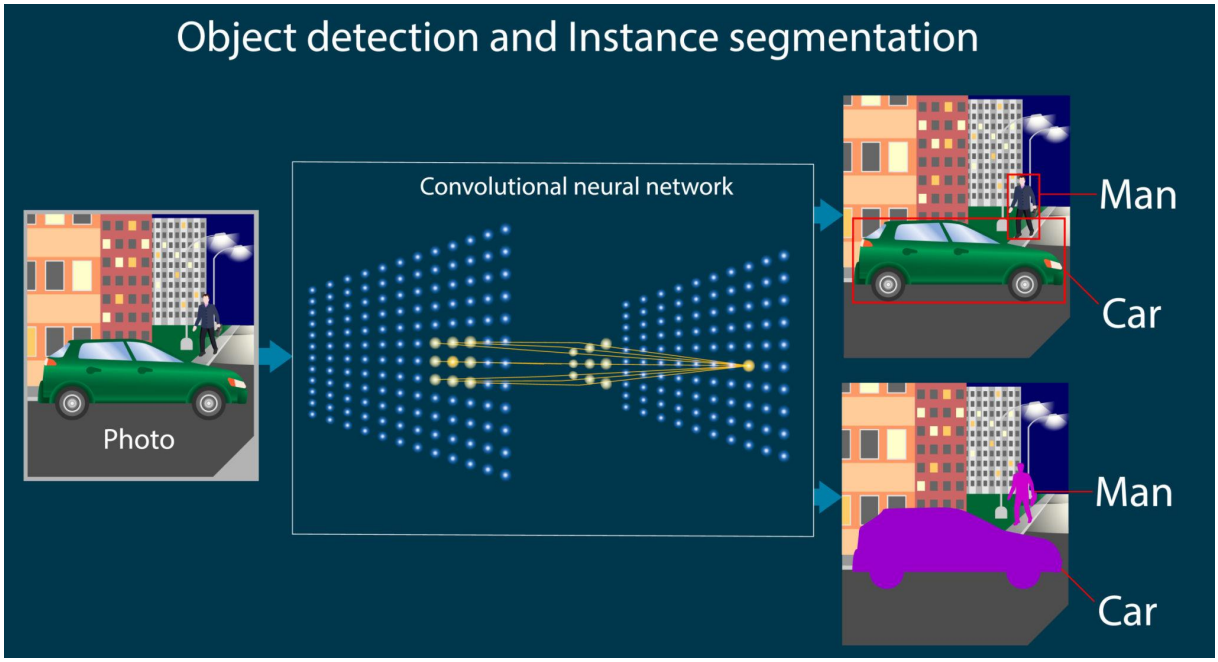


Рисунок 2.7 – Приклад використання згорткової нейронної мережі для задач детекції та сегментації об'єктів на зображенні

У задачах розпізнавання їжі згорткові нейронні мережі застосовуються для класифікації страв, визначення інгредієнтів та локалізації об'єктів на зображенні. Мережа аналізує зображення у кілька етапів, починаючи з виділення простих ознак, таких як контури та текстури,

і завершуючи формуванням більш складних характеристик, що дозволяють ідентифікувати конкретні продукти або типи страв. Такий підхід дає змогу обробляти складні сцени, де на одному зображенні можуть бути присутні кілька різних інгредієнтів.

2.3.2 Архітектури нейронних мереж для класифікації зображень

У задачах комп'ютерного зору для класифікації зображень широко використовуються різні архітектури глибоких нейронних мереж, які відрізняються структурою та принципами обробки візуальної інформації. Одними з найбільш відомих архітектур є VGGNet, ResNet, Inception (GoogLeNet) та EfficientNet. Ці моделі були розроблені для підвищення точності класифікації зображень і стали основою багатьох сучасних систем комп'ютерного зору.

Наприклад, архітектура VGGNet використовує послідовність згорткових шарів невеликого розміру, що дозволяє ефективно виділяти ознаки зображення, тоді як ResNet (рисунок 2.8) застосовує так звані залишкові з'єднання, які допомагають навчати дуже глибокі нейронні мережі без втрати інформації.

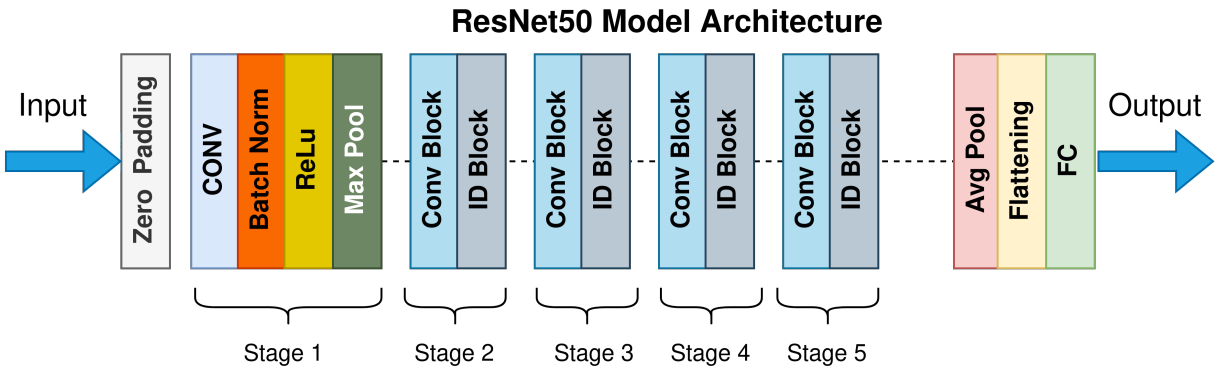


Рисунок 2.8 – Архітектура ResNet50

Архітектура Inception використовує спеціальні модулі, які дозволяють одночасно виконувати згортки різного розміру, що дає змогу ефективно аналізувати об’єкти різних масштабів на зображенні. У свою чергу, EfficientNet застосовує оптимізований підхід до масштабування мережі, що дозволяє досягати високої точності при відносно невеликій кількості параметрів. Завдяки таким архітектурним рішенням сучасні нейронні мережі здатні ефективно аналізувати складні зображення та розпізнавати велику кількість різних об’єктів.

2.3.3 Сучасні моделі комп’ютерного зору

У сучасних дослідженнях комп’ютерного зору все більшого поширення набувають моделі, що використовують архітектуру трансформерів. Однією з таких моделей є Vision Transformer (ViT), яка була запропонована як альтернатива традиційним згортковим нейронним мережам для задач аналізу зображень. На відміну від CNN, де обробка виконується за допомогою згорткових шарів, Vision Transformer використовує механізм самоуваги, що дозволяє моделі враховувати глобальні залежності між різними частинами зображення. Завдяки цьому модель здатна ефективно аналізувати складні сцени та покращувати точність класифікації.

Принцип роботи Vision Transformer полягає у розбитті зображення на невеликі фрагменти – так звані patches, які перетворюються у векторні представлення. До цих представлень додається позиційна інформація, після чого вони подаються на вхід трансформерного енкодера. Усередині енкодера використовується механізм multi-head attention, який дозволяє моделі визначати взаємозв’язки між різними частинами зображення. Такий підхід дає можливість ефективно виділяти важливі ознаки та формувати більш повне представлення візуальної інформації.

Схематично архітектура моделі Vision Transformer для задач комп’ютерного зору наведена нижче (рисунок 2.9). На схемі показано процес поділу зображення на фрагменти, їх перетворення у векторні представлення та подальшу обробку за допомогою трансформерного енкодера. Сучасні моделі комп’ютерного зору, засновані на архітектурі трансформерів, активно застосовуються у задачах класифікації зображень, детекції об’єктів та аналізу складних візуальних сцен.

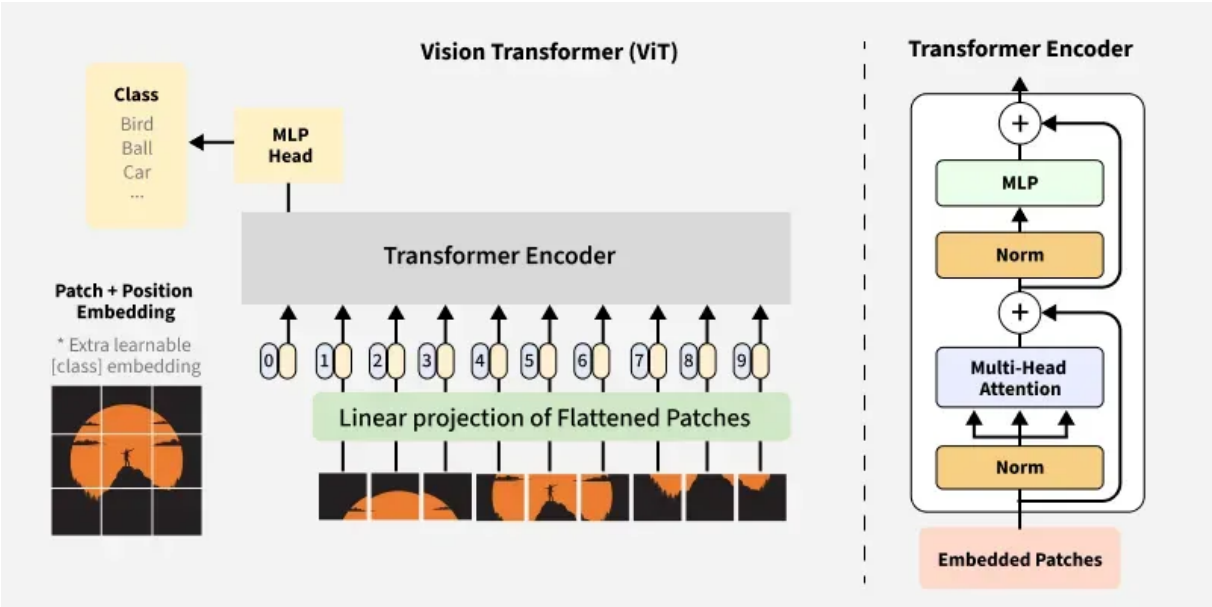


Рисунок 2.9 – Архітектура Vision Transformer

2.4 Використання OpenAI Vision API для аналізу зображень їжі

Одним із сучасних підходів до аналізу зображень є використання великих мультимодальних моделей штучного інтелекту, які здатні одночасно працювати з текстовою та візуальною інформацією. До таких рішень належить OpenAI Vision API, що дозволяє виконувати розпізнавання об’єктів, опис сцени та інтерпретацію змісту зображень без необхідності створення власної моделі комп’ютерного зору. У контексті систем моніторингу харчування така технологія може використовуватися для

автоматичного визначення типу страви, її інгредієнтів та приблизних нутріційних характеристик на основі фотографії.

Особливістю використання Vision API є те, що аналіз зображення виконується за допомогою попередньо навченої великої моделі, яка має значний обсяг знань про об’єкти реального світу. Користувачеві достатньо зробити фотографію прийому їжі, після чого система передає зображення до моделі, яка формує структурований опис страви. На основі отриманої інформації можуть бути визначені складові інгредієнти, приблизна калорійність та співвідношення макронутрієнтів. Застосування такого підходу дозволяє значно спростити процес ведення харчового щоденника, оскільки користувачеві не потрібно вручну вводити назву продукту або його харчову цінність.

2.4.1 Принцип роботи мультимодальних моделей

Мультимодальні моделі штучного інтелекту призначені для одночасної обробки різних типів даних, зокрема тексту, зображень та інших форм інформації. Основна ідея таких моделей полягає у формуванні єдиного представлення даних, що дозволяє системі встановлювати зв’язки між візуальними об’єктами та їх текстовими описами. Завдяки цьому модель може інтерпретувати зміст зображення та генерувати пояснення або структуровані дані.

У випадку аналізу фотографій їжі мультимодальна модель спочатку виконує обробку візуальної інформації, визначаючи ключові об’єкти на зображенні. Після цього модель формує текстовий опис знайдених елементів, наприклад назви інгредієнтів або типи страв. Такий підхід дозволяє інтегрувати результати аналізу зображення у подальші етапи обробки, наприклад у систему підрахунку калорій або формування рекомендацій.

2.4.2 Можливості Vision API для аналізу харчових зображень

У системах моніторингу харчування Vision API використовується для автоматичного розпізнавання продуктів на зображеннях. Після отримання фотографії модель аналізує її зміст та визначає, які об’єкти присутні на зображенні. Зокрема, вона може ідентифікувати різні типи страв, інгредієнтів або напоїв, що дозволяє отримати базову інформацію про прийом їжі.

Після розпізнавання елементів страви система може виконувати подальшу обробку отриманих даних. Наприклад, на основі визначених інгредієнтів можна оцінити приблизну калорійність страви та вміст основних макронутрієнтів. Отримана інформація зберігається у базі даних застосунку та використовується для подальшого аналізу харчових звичок користувача.

2.4.3 Переваги використання хмарних AI-сервісів для задач розпізнавання їжі

Використання готових AI-сервісів, таких як Vision API, має низку важливих переваг. По-перше, розробнику не потрібно створювати власні моделі комп’ютерного зору або збирати великі набори даних для їх навчання. Моделі вже навчені на великих обсягах інформації та здатні ефективно розпізнавати різноманітні об’єкти на зображеннях.

По-друге, використання готових сервісів значно скорочує час розробки програмного продукту. Інтеграція такого API дозволяє швидко реалізувати функціональність аналізу зображень, що особливо важливо для мобільних застосунків. Крім того, AI-сервіси регулярно оновлюються та вдосконалюються, що дозволяє системі автоматично отримувати покращення алгоритмів.

2.5 Використання Foundation Models та on-device AI у мобільних застосунках

Розвиток мобільних платформ останніми роками призвів до активного впровадження технологій штучного інтелекту безпосередньо на пристроях користувачів. Одним із ключових напрямів такого розвитку є використання Foundation Models – великих універсальних моделей, здатних виконувати широкий спектр задач аналізу даних. На відміну від класичних моделей машинного навчання, які створюються для конкретної задачі, foundation-моделі мають більш узагальнений характер і можуть використовуватися для обробки тексту, зображень, аудіо та інших типів інформації.

У мобільних застосунках використання таких моделей дозволяє реалізувати складні інтелектуальні функції, зокрема аналіз поведінки користувача, інтерпретацію накопичених даних та формування персоналізованих рекомендацій. Важливою особливістю сучасних підходів є використання on-device AI, тобто виконання моделей безпосередньо на пристрої користувача. Це забезпечує швидку обробку інформації, зменшує залежність від мережевого з’єднання та підвищує рівень конфіденційності персональних даних.

2.5.1 Концепція Apple Intelligence та Foundation Models

Компанія Apple активно розвиває концепцію інтеграції штучного інтелекту у свої операційні системи через платформу Apple Intelligence, яка поєднує можливості локальних моделей та хмарних обчислень. Основна ідея цієї концепції полягає у використанні Foundation Models, які можуть виконувати складні задачі аналізу інформації та генерації контенту безпосередньо на пристроях користувачів.

Архітектура Apple Intelligence (рисунок 2.10) передбачає багаторівневу систему обробки даних, у якій частина обчислень

виконується локально на пристрої, а складніші задачі можуть передаватися до хмарної інфраструктури. Такий підхід дозволяє поєднати високу продуктивність локальних моделей з масштабованістю серверних рішень.

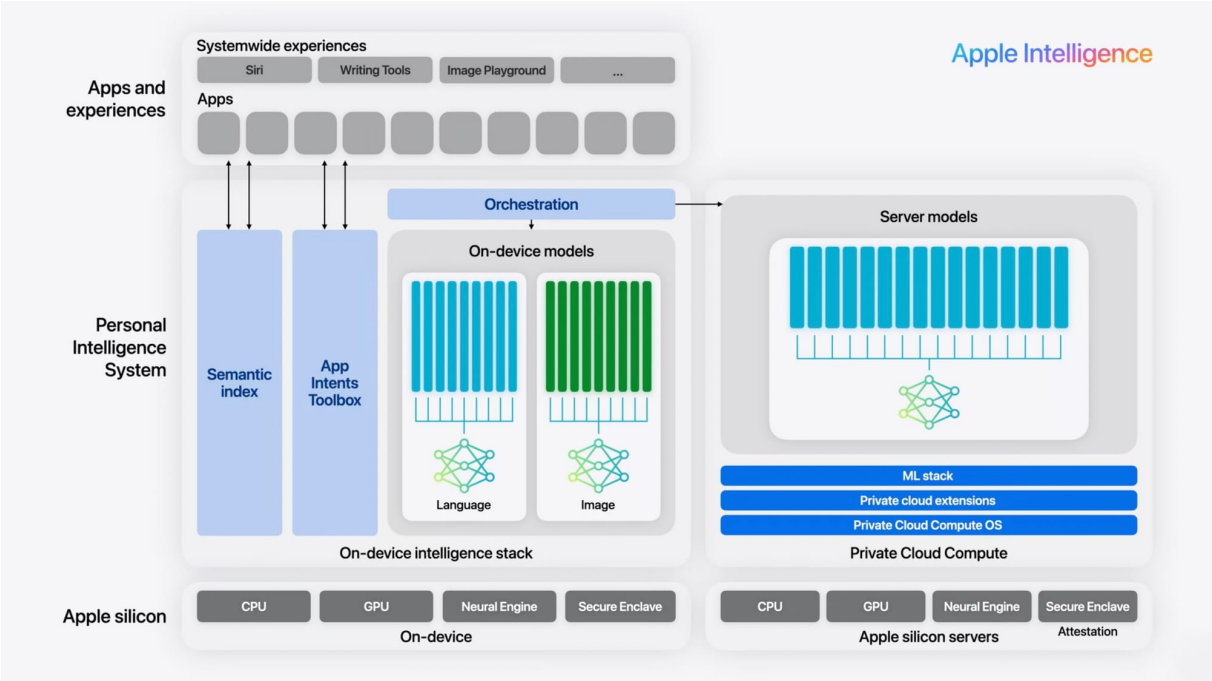


Рисунок 2.10 – Архітектура системи Apple Intelligence

Згідно зі схемою система Apple Intelligence включає декілька ключових компонентів: модулі обробки даних на пристрої, інструменти взаємодії застосунків із моделями, а також серверні моделі для виконання більш складних обчислень. Завдяки такій архітектурі розробники мобільних застосунків можуть інтегрувати можливості штучного інтелекту без необхідності створення власних великих моделей.

2.5.2 Особливості використання on-device AI у мобільних системах

Технологія on-device AI передбачає виконання алгоритмів штучного інтелекту безпосередньо на мобільному пристрої користувача. Такий підхід має низку важливих переваг, серед яких висока швидкість обробки

інформації, зменшення затримок та незалежність від стабільності інтернет-з’єднання. Це особливо важливо для мобільних застосунків, які повинні працювати швидко та забезпечувати миттєвий відгук системи.

Ще однією важливою перевагою on-device AI є підвищений рівень конфіденційності даних. Оскільки обробка інформації виконується локально, персональні дані користувача не передаються на зовнішні сервери. Це є критично важливим для систем, що працюють із чутливою інформацією, зокрема даними про стан здоров’я або харчові звички користувачів. Крім того, сучасні мобільні процесори оснащені спеціалізованими обчислювальними модулями, такими як Neural Engine, які оптимізовані для виконання задач машинного навчання. Це дозволяє ефективно використовувати моделі штучного інтелекту навіть на портативних пристроях із обмеженими обчислювальними ресурсами.

2.5.3 Застосування генеративних моделей для формування персоналізованих рекомендацій

Одним із перспективних напрямів використання foundation-моделей у мобільних застосунках є формування персоналізованих рекомендацій на основі накопичених даних користувача. Генеративні моделі здатні аналізувати історію дій користувача, виявляти закономірності та формувати рекомендації, адаптовані до індивідуальних потреб.

У системах моніторингу харчування такі моделі можуть аналізувати історію спожитих продуктів, співвідношення макронутрієнтів, частоту прийомів їжі та інше. На основі цього аналізу модель може формувати рекомендації щодо оптимізації харчування. Таким чином, використання генеративних моделей дозволяє перейти від простого фіксування даних до більш інтелектуального аналізу поведінки користувача.

3 ПРОЕКТУВАННЯ СИСТЕМИ

3.1 Аналіз вимог до системи моніторингу харчування

Проєктування програмної системи є важливим етапом розробки мобільного застосунку, оскільки саме на цьому етапі визначаються основні вимоги до майбутнього програмного продукту, його функціональні можливості та технічні характеристики. Аналіз вимог дозволяє сформувати чітке уявлення про те, які задачі повинна виконувати система, які обмеження необхідно врахувати під час її розробки та яким чином користувач буде взаємодіяти із застосунком. Для систем моніторингу харчування особливо важливим є врахування специфіки обробки даних про раціон користувача, а також використання технологій комп’ютерного зору та штучного інтелекту для аналізу фотографій їжі.

У цьому розділі буде проведено аналіз основних вимог до системи моніторингу харчування, що розробляється. Зокрема, буде визначено функціональні можливості мобільного застосунку, сформульовано нефункціональні вимоги до програмного продукту, а також розглянуто технічні обмеження мобільної платформи та вимоги до користувацького інтерфейсу. Такий підхід дозволяє систематизувати основні характеристики майбутньої системи та сформувати основу для подальшого проєктування її архітектури, функціональних модулів і логіки роботи.

3.1.1 Функціональні вимоги до мобільного застосунку

Мобільний застосунок системи моніторингу харчування повинен забезпечувати можливість автоматичного аналізу прийомів їжі на основі фотографій. Користувач має мати змогу зробити фотографію страви безпосередньо у застосунку або завантажити вже існуюче зображення з галереї пристрою. Після отримання зображення система повинна виконати

його аналіз за допомогою технологій комп’ютерного зору та визначити тип страви або основні інгредієнти, що присутні на фотографії. На основі отриманих результатів застосунок має формувати інформацію про приблизну калорійність страви та вміст основних макронутрієнтів.

Другим важливим функціональним елементом системи є можливість збереження та аналізу історії харчування користувача. Застосунок повинен забезпечувати ведення харчового щоденника, у якому зберігається інформація про всі прийоми їжі, що були зафіксовані користувачем. Користувач повинен мати можливість переглядати історію спожитих страв за певний період часу, а також отримувати статистичну інформацію про калорійність раціону та співвідношення основних нутрієнтів. Такий функціонал дозволяє користувачу краще розуміти свої харчові звички та контролювати структуру власного раціону.

Ще однією важливою функціональною можливістю застосунку є формування персоналізованих рекомендацій щодо оптимізації харчування. Система повинна аналізувати накопичені дані про раціон користувача за визначений період часу та на основі цього аналізу генерувати рекомендації щодо покращення балансу нутрієнтів або зміни структури харчування. При формуванні таких рекомендацій застосунок має враховувати поставлені користувачем цілі, наприклад підтримку здорового способу життя, контроль калорійності раціону або збалансування співвідношення макронутрієнтів. Такий підхід дозволяє перетворити застосунок із простого інструменту обліку харчування на інтелектуальну систему підтримки прийняття рішень щодо раціону.

3.1.2 Нефункціональні вимоги до програмного продукту

Нефункціональні вимоги визначають загальні характеристики програмного продукту, які забезпечують його стабільність, ефективність та зручність використання. Для мобільного застосунку моніторингу

харчування важливо забезпечити високу продуктивність системи під час обробки фотографій та аналізу даних користувача. Час виконання основних операцій, таких як обробка зображення та отримання результатів аналізу, повинен бути мінімальним, щоб користувач міг швидко отримувати необхідну інформацію без значних затримок.

Важливою нефункціональною характеристикою системи є надійність та стабільність роботи застосунку. Програмний продукт повинен коректно працювати на різних моделях мобільних пристроїв, забезпечуючи безперервний доступ до основних функцій навіть при нестабільному мережевому з’єднанні. Крім того, система повинна забезпечувати збереження даних користувача та їх відновлення у разі виникнення технічних збоїв або перезапуску застосунку.

Окрему увагу необхідно приділити питанням безпеки та конфіденційності даних. Оскільки застосунок працює з персональною інформацією користувача, зокрема історією харчування, система повинна забезпечувати належний рівень захисту цих даних. Обробка частини інформації може виконуватися безпосередньо на мобільному пристрої за допомогою технологій on-device штучного інтелекту, що дозволяє мінімізувати передачу персональних даних на зовнішні сервери та підвищити рівень конфіденційності користувача.

3.1.3 Технічні вимоги до мобільної платформи

Мобільний застосунок системи моніторингу харчування повинен бути реалізований з урахуванням технічних можливостей сучасних мобільних пристроїв. Основною платформою для розробки обрано операційну систему iOS, що забезпечує доступ до сучасних інструментів розробки, фреймворків штучного інтелекту та засобів обробки мультимедійних даних. Використання нативних технологій дозволяє забезпечити високу продуктивність застосунку, ефективне використання ресурсів пристрою та

стабільну роботу програмного продукту на різних моделях мобільних пристроїв.

З технічної точки зору система повинна підтримувати роботу з камерою мобільного пристрою для отримання фотографій страв, а також можливість доступу до галереї користувача для завантаження вже існуючих зображень. Крім того, застосунок повинен забезпечувати передачу зображень до сервісів штучного інтелекту для їх подальшого аналізу, зокрема до API комп'ютерного зору. Для цього необхідно забезпечити стабільну роботу мережевих запитів, обробку отриманих результатів та їх інтеграцію у внутрішню логіку застосунку.

Також важливим технічним аспектом є підтримка сучасних обчислювальних можливостей мобільних пристроїв, зокрема використання локальних модулів машинного навчання та оптимізованих апаратних компонентів. Сучасні мобільні процесори містять спеціалізовані блоки для виконання задач штучного інтелекту, що дозволяє реалізувати частину обробки даних безпосередньо на пристрої користувача. Використання таких можливостей сприяє підвищенню швидкості роботи застосунку та зменшенню залежності від віддалених серверів.

3.1.4 Вимоги до користувацького інтерфейсу

Користувацький інтерфейс мобільного застосунку повинен бути інтуїтивно зрозумілим та зручним для щоденного використання. Оскільки система призначена для регулярного фіксування прийомів їжі, інтерфейс повинен забезпечувати швидкий доступ до основних функцій застосунку, зокрема до створення фотографії страви, перегляду історії харчування та отримання рекомендацій. Важливим аспектом є мінімізація кількості дій, необхідних для виконання основних операцій, щоб користувач міг швидко зафіксувати прийом їжі без складних налаштувань або введення великої кількості даних.

Інтерфейс застосунку повинен бути структурованим та логічно організованим, щоб користувач міг легко орієнтуватися у функціональних можливостях системи. Основні елементи інтерфейсу повинні включати екран додавання прийому їжі, сторінку перегляду історії харчування та розділ із персоналізованими рекомендаціями. Важливу роль відіграє візуальне представлення інформації, зокрема використання графіків, діаграм або інших елементів візуалізації для відображення калорійності раціону та співвідношення макронутрієнтів.

Окрему увагу необхідно приділити адаптації інтерфейсу до особливостей мобільних пристроїв. Інтерфейс повинен коректно відображатися на екранах різних розмірів, забезпечувати плавну навігацію між екранами та підтримувати сучасні принципи дизайну мобільних застосунків. Використання стандартних елементів інтерфейсу мобільної платформи дозволяє забезпечити зрозумілу взаємодію користувача із системою та підвищити загальну зручність використання застосунку.

3.2 Функціональна модель застосунку

Функціональна модель мобільного застосунку (рисунок 3.1) NutriGuide відображає основні програмні компоненти системи та їх взаємодію у процесі аналізу харчування користувача. Така модель дозволяє формалізувати внутрішню структуру застосунку, визначити основні функціональні модулі та встановити зв’язки між ними. Основною метою функціональної моделі є демонстрація того, як система обробляє вхідні дані, виконує їх аналіз та формує результати у вигляді аналітичної інформації та персоналізованих рекомендацій.

Центральним елементом системи є модуль аналізу фотографій їжі, який відповідає за обробку зображень, отриманих від користувача. Після отримання фотографії вона передається до зовнішнього сервісу OpenAI Vision API, що виконує розпізнавання страви, визначення її складу та

оцінювання харчової цінності. У результаті обробки формується структурована інформація, яка може містити назву страви, приблизну калорійність, а також вміст основних макронутрієнтів – білків, жирів та вуглеводів. Отримані дані повертаються до мобільного застосунку для подальшого використання.

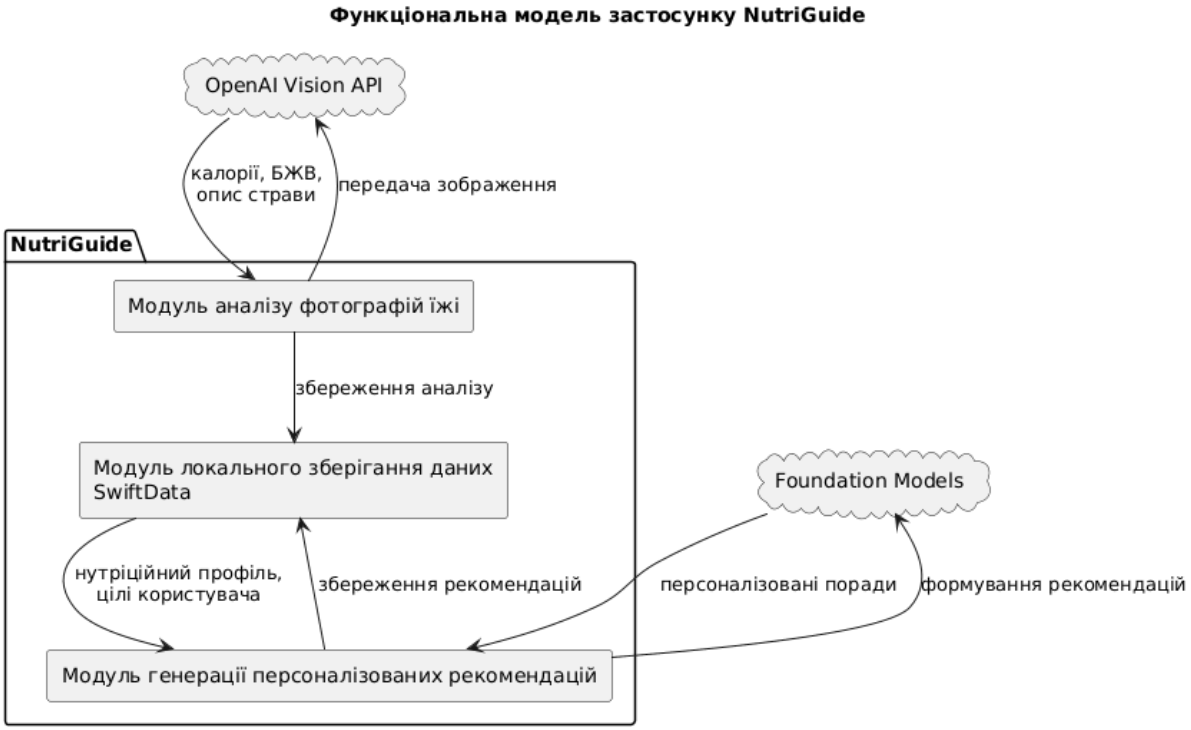


Рисунок 3.1 – Функціональна модель застосунку

Після завершення аналізу результати зберігаються у модулі локального зберігання даних, реалізованому за допомогою технології SwiftData. Цей модуль виконує роль внутрішнього сховища інформації про всі прийоми їжі користувача, забезпечуючи накопичення історії харчування. Наявність такої бази даних дозволяє системі виконувати подальший аналіз раціону, оцінювати харчові звички користувача та формувати індивідуальний нутріційний профіль.

На основі накопичених даних активується модуль генерації персоналізованих рекомендацій. Він використовує інформацію про історію

харчування, цілі користувача та нутріційний баланс для формування корисних порад щодо оптимізації раціону. Для цього застосовуються сучасні генеративні моделі штучного інтелекту Foundation Models, які здатні аналізувати харчові патерни та генерувати рекомендації у зрозумілій текстовій формі.

Таким чином, функціональна модель демонструє логіку роботи застосунку, у межах якої відбувається поєднання декількох технологічних компонентів – аналізу зображень, локального зберігання даних та генерації інтелектуальних рекомендацій.

3.2.1 Модуль аналізу фотографій їжі

Модуль аналізу фотографій їжі є одним із ключових компонентів мобільного застосунку NutriGuide, оскільки саме він забезпечує автоматичне визначення складу та харчової цінності страв на основі їх зображень. Основним завданням цього модуля є отримання фотографії від користувача, її попередня обробка та передача до системи аналізу зображень. Такий підхід дозволяє значно спростити процес ведення харчового щоденника, оскільки користувачеві достатньо зробити фотографію страви замість ручного введення даних.

Після отримання зображення система виконує базову попередню обробку, яка може включати оптимізацію розміру зображення, корекцію формату та підготовку даних до подальшого аналізу. Підготовлене зображення передається до зовнішнього сервісу OpenAI Vision API, який використовує сучасні моделі комп’ютерного зору для розпізнавання об’єктів на зображенні. У результаті аналізу система отримує структуровану інформацію про страву, що може включати її назву, складові інгредієнти, приблизну калорійність та вміст основних макронутрієнтів.

Отримані результати обробки передаються назад до мобільного застосунку та використовуються для створення запису про прийом їжі. Ці

дані зберігаються у внутрішньому сховищі застосунку та можуть бути використані для подальшого аналізу харчування користувача. Таким чином, модуль аналізу фотографій виконує роль інтелектуального інтерфейсу між користувачем та системою обліку харчування, автоматизуючи процес збору інформації про раціон.

3.2.2 Модуль збереження та аналізу історії харчування

Модуль збереження та аналізу історії харчування відповідає за накопичення та структуроване зберігання інформації про всі прийоми їжі користувача. Основною функцією цього модуля є формування централізованого сховища даних, у якому зберігаються результати аналізу фотографій, інформація про харчову цінність страв, а також часові параметри прийомів їжі. Для реалізації локального зберігання даних у застосунку використовується технологія SwiftData, яка забезпечує ефективну роботу з моделями даних у середовищі iOS.

Накопичена інформація дозволяє системі формувати повну історію харчування користувача за різні часові періоди, зокрема за день, тиждень або місяць. На основі цих даних можуть виконуватися різні види аналітики, включаючи підрахунок сумарної калорійності раціону, визначення співвідношення макронутрієнтів та виявлення харчових патернів. Такий аналіз дозволяє користувачу отримувати об’єктивне уявлення про власні харчові звички та рівень дотримання встановлених цілей.

Крім збереження даних, модуль також забезпечує їх подальшу обробку для використання іншими компонентами системи. Зокрема, інформація про історію харчування передається до модуля генерації рекомендацій, який використовує ці дані для формування індивідуальних порад щодо оптимізації раціону. Таким чином, модуль історії харчування виконує роль основи для інтелектуальних функцій застосунку.

3.2.3 Модуль генерації персоналізованих рекомендацій

Модуль генерації персоналізованих рекомендацій призначений для формування індивідуальних порад щодо оптимізації харчування користувача на основі аналізу його раціону. Основним завданням цього модуля є інтерпретація накопичених даних про прийоми їжі та перетворення їх на зрозумілі рекомендації, які можуть допомогти користувачу досягти поставлених цілей харчування. До таких цілей можуть належати зниження або збільшення маси тіла, підтримання балансу макронутрієнтів чи формування більш здорових харчових звичок.

Для реалізації цього функціоналу використовуються генеративні моделі штучного інтелекту Foundation Models, які здатні аналізувати великі обсяги текстової та структурованої інформації. На основі даних про історію харчування, калорійність раціону та співвідношення макронутрієнтів модель формує текстові рекомендації щодо покращення раціону. Такі рекомендації можуть включати поради щодо збільшення або зменшення споживання певних груп продуктів, корекції калорійності раціону або оптимізації балансу поживних речовин.

Згенеровані рекомендації передаються до інтерфейсу мобільного застосунку, де вони відображаються користувачу у зручному та зрозумілому вигляді. Це дозволяє не лише відстежувати власне харчування, але й отримувати інтелектуальну підтримку у формуванні здорового раціону. Таким чином, модуль генерації рекомендацій є ключовим елементом системи, який перетворює звичайний трекер харчування на інтелектуальний інструмент аналізу та покращення харчових звичок.

3.3 Модель взаємодії користувача із системою

Модель взаємодії користувача із системою (рисунок 3.2) NutriGuide відображає основні способи використання функціональних можливостей

мобільного застосунку та демонструє ключові сценарії взаємодії між користувачем і програмною системою. Така модель дозволяє формалізувати основні дії користувача в межах застосунку, а також визначити, які функції системи активуються під час виконання цих дій. Вона є важливим етапом проєктування, оскільки допомагає зрозуміти логіку роботи застосунку з точки зору кінцевого користувача.



Рисунок 3.2 – Модель взаємодії користувача із системою

Основним актором у системі є користувач, який взаємодіє із застосунком через інтерфейс мобільного пристрою. Найбільш важливим сценарієм використання є фотографування страви, після чого система виконує аналіз отриманого зображення за допомогою модуля розпізнавання їжі. Після завершення аналізу інформація про страву, включаючи її назву, калорійність та вміст основних макронутрієнтів, автоматично зберігається у системі як запис про прийом їжі.

Крім додавання нових записів, користувач може переглядати історію харчування, що містить усі зафіксовані прийоми їжі за певний період часу. На основі накопичених даних система формує статистичну інформацію про харчування, яка дозволяє оцінити калорійність раціону, баланс макронутрієнтів та загальні харчові звички. Така функціональність

допомагає користувачеві отримувати більш повне уявлення про власний раціон.

Додатково система забезпечує можливість отримання персоналізованих рекомендацій щодо оптимізації харчування. Для цього використовується аналіз історії прийомів їжі та генеративні моделі штучного інтелекту, які формують поради з урахуванням індивідуальних цілей користувача.

3.3.1 Основні ролі користувачів системи

У мобільному застосунку NutriGuide передбачено використання однієї основної ролі користувача, якою є кінцевий користувач системи. Саме ця роль відповідає за взаємодію із функціональними можливостями застосунку та використання інструментів моніторингу харчування. На відміну від складних інформаційних систем, що передбачають декілька ролей з різними рівнями доступу, у даному застосунку всі функції орієнтовані безпосередньо на особисте використання користувачем.

Користувач взаємодіє із системою через інтерфейс мобільного застосунку та виконує основні операції, пов’язані з фіксацією та аналізом власного харчування. До таких операцій належать фотографування страв, перегляд історії прийомів їжі, аналіз статистики харчування та отримання персоналізованих рекомендацій щодо оптимізації раціону. Усі ці дії виконуються безпосередньо на мобільному пристрої користувача та не потребують додаткових ролей або адміністративного управління.

Крім безпосереднього введення даних, користувач також отримує результати роботи інтелектуальних модулів системи. До них належать результати аналізу фотографій їжі, аналітична інформація про структуру раціону та рекомендації щодо покращення харчових звичок.

3.3.2 Основні сценарії використання застосунку

Основні сценарії (рисунок 3.3) використання мобільного застосунку NutriGuide описують послідовність дій, які виконує користувач під час взаємодії із системою для аналізу та моніторингу власного харчування. Такі сценарії дозволяють формалізувати процеси роботи застосунку та визначити, які функціональні модулі системи беруть участь у виконанні окремих операцій. Моделювання сценаріїв використання є важливим етапом проєктування програмної системи, оскільки воно демонструє логіку обміну даними між компонентами застосунку.

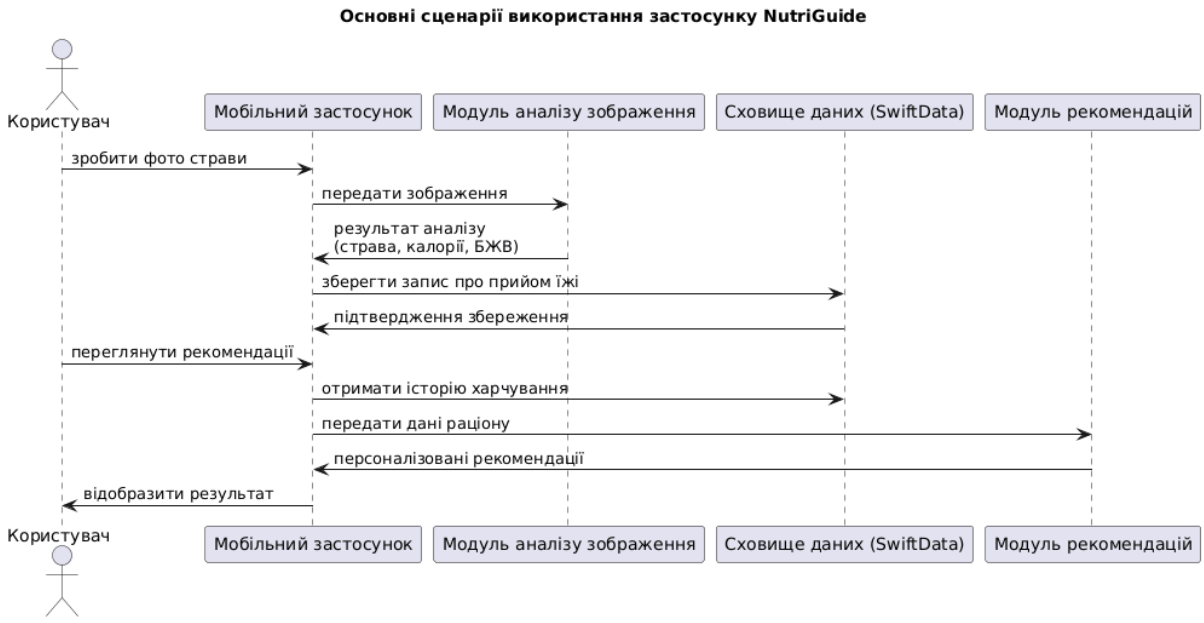


Рисунок 3.3 – Основні сценарії використання застосунку

Одним із основних сценаріїв роботи системи є додавання нового прийому їжі за допомогою фотографування страви. У цьому випадку користувач робить фотографію страви за допомогою камери мобільного пристрою, після чого зображення передається до модуля аналізу зображень. Модуль аналізу виконує розпізнавання страви та визначає її основні харчові характеристики, зокрема приблизну калорійність та вміст макронутрієнтів.

Отримані результати аналізу повертаються до мобільного застосунку, після чого створюється запис про прийом їжі, який зберігається у локальному сховищі даних SwiftData.

Іншим важливим сценарієм є отримання користувачем персоналізованих рекомендацій щодо харчування. Для цього система використовує накопичену історію прийомів їжі, що зберігається у базі даних застосунку. Дані про раціон користувача передаються до модуля рекомендацій, який виконує аналіз харчових звичок та формує індивідуальні поради щодо оптимізації раціону. Після завершення обробки рекомендації передаються назад до мобільного застосунку та відображаються користувачу у зрозумілому текстовому вигляді.

Таким чином, основні сценарії використання застосунку демонструють повний цикл роботи системи – від отримання вхідних даних у вигляді фотографії страви до формування аналітичної інформації та рекомендацій.

3.3.3 Сценарій аналізу фотографії страви

Сценарій аналізу фотографії страви (рисунок 3.4) є одним із ключових процесів функціонування мобільного застосунку NutriGuide, оскільки саме він забезпечує автоматичне визначення харчових характеристик спожитої їжі. Основна мета цього сценарію полягає у спрощенні процесу ведення харчового щоденника шляхом використання технологій комп’ютерного зору. Завдяки цьому користувач може швидко додавати інформацію про прийоми їжі без необхідності ручного введення даних.

На початковому етапі користувач робить фотографію страви за допомогою камери мобільного пристрою через інтерфейс застосунку. Отримане зображення передається до мобільного застосунку, який виконує його підготовку та відправляє до сервісу Vision API для подальшого аналізу.

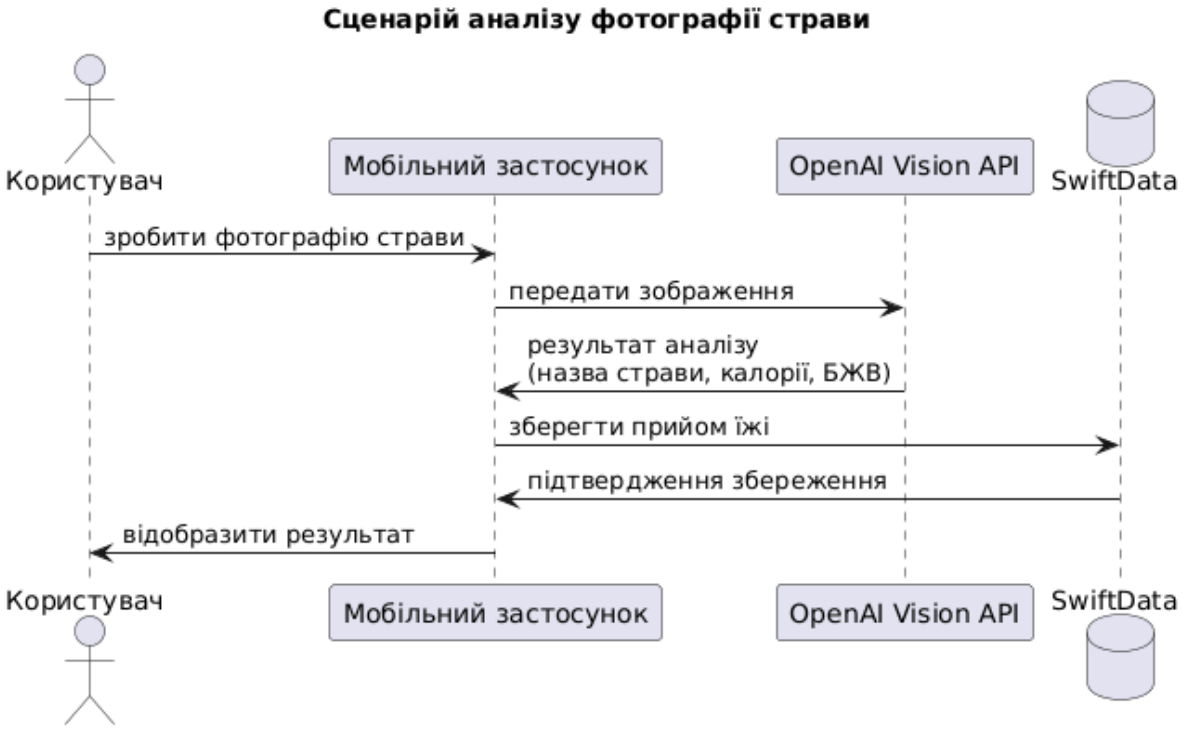


Рисунок 3.4 – Сценарій аналізу фотографії страви

У процесі обробки зображення система розпізнає тип страви, визначає її основні складові та оцінює приблизну калорійність і вміст макронутрієнтів, зокрема білків, жирів і вуглеводів. Після завершення аналізу отримані результати повертаються до мобільного застосунку у структурованому вигляді. На основі цих даних формується запис про прийом їжі. Далі цей запис зберігається у локальному сховищі даних, реалізованому за допомогою технології SwiftData, що дозволяє накопичувати історію харчування користувача.

На завершальному етапі система підтверджує успішне збереження даних та відображає результати аналізу у користувацькому інтерфейсі мобільного застосунку. Користувач отримує інформацію про розпізнану страву та її поживні характеристики, після чого ці дані можуть бути використані для подальшого аналізу раціону та формування персоналізованих рекомендацій.

3.3.4 Сценарій отримання рекомендацій щодо харчування

Сценарій отримання рекомендацій (рисунок 3.5) щодо харчування описує процес формування індивідуальних порад на основі історії харчування користувача, що накопичується у мобільному застосунку NutriGuide. Основною метою цього сценарію є надання користувачу корисної аналітичної інформації, яка допоможе оптимізувати раціон відповідно до поставлених цілей харчування. Такий підхід дозволяє перетворити застосунок із простого інструменту обліку їжі на інтелектуальну систему підтримки прийняття рішень.

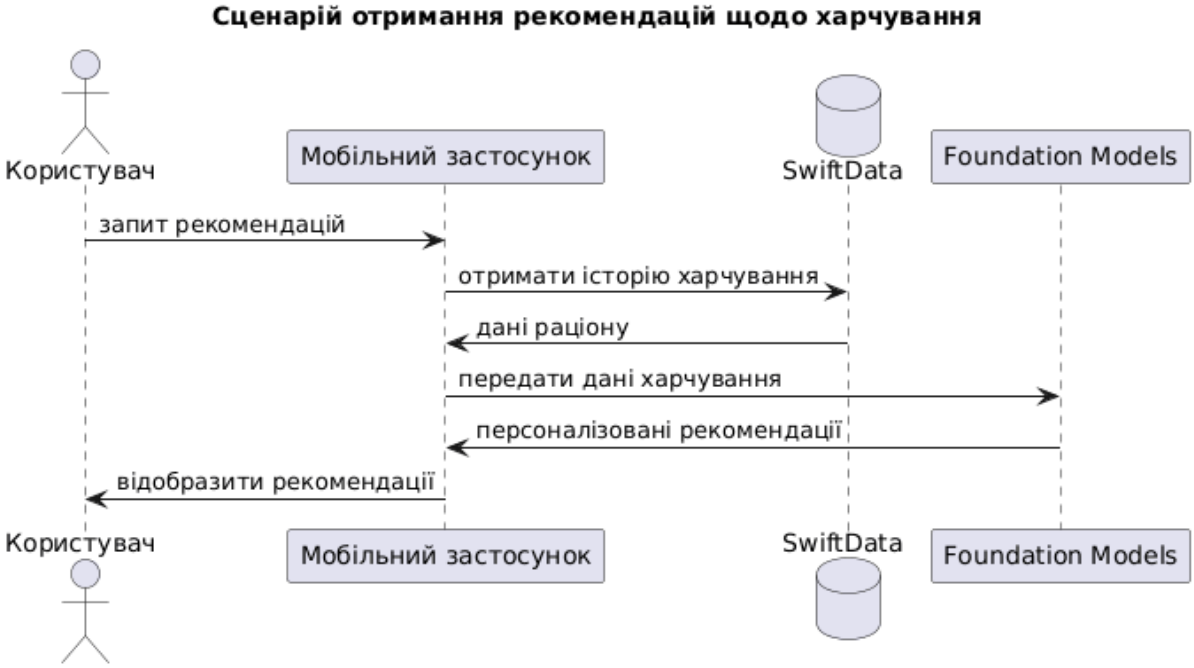


Рисунок 3.5 – Сценарій отримання рекомендацій щодо харчування

На початковому етапі користувач ініціює запит на отримання рекомендацій через інтерфейс мобільного застосунку. Після цього система звертається до локального сховища даних SwiftData для отримання інформації про історію харчування користувача. Зі сховища витягуються дані про попередні прийоми їжі, їх калорійність та структуру

макронутрієнтів, що дозволяє сформувати узагальнений профіль харчування.

Далі отримані дані передаються до модуля рекомендацій, який використовує сучасні генеративні моделі штучного інтелекту Foundation Models для виконання аналізу раціону. На основі отриманої інформації система визначає можливі дисбаланси у харчуванні та формує текстові персоналізовані рекомендації. Такі рекомендації можуть стосуватися зміни калорійності раціону, корекції співвідношення білків, жирів і вуглеводів або збільшення споживання певних груп продуктів.

На завершальному етапі сформовані рекомендації передаються назад до мобільного застосунку та відображаються користувачу у зрозумілому вигляді. Це дозволяє користувачу отримувати практичні поради щодо покращення власного харчування на основі об’єктивного аналізу даних.

3.4 Архітектура мобільного застосунку

Архітектура мобільного застосунку (рисунок 3.6) NutriGuide визначає структуру програмної системи, основні програмні компоненти та взаємодію між ними. Основною метою архітектурного проєктування є забезпечення чіткої організації коду, розділення відповідальностей між модулями та підвищення масштабованості й підтримуваності застосунку.

Користувацький інтерфейс застосунку реалізовано за допомогою фреймворку SwiftUI, який забезпечує сучасний декларативний підхід до розробки інтерфейсів мобільних застосунків для платформи iOS. Основна бізнес-логіка системи реалізована через компоненти ViewModel та сервісні модулі, які відповідають за обробку даних, взаємодію з зовнішніми сервісами та координацію роботи різних функціональних частин застосунку. Для локального зберігання інформації про прийоми їжі використовується технологія SwiftData.

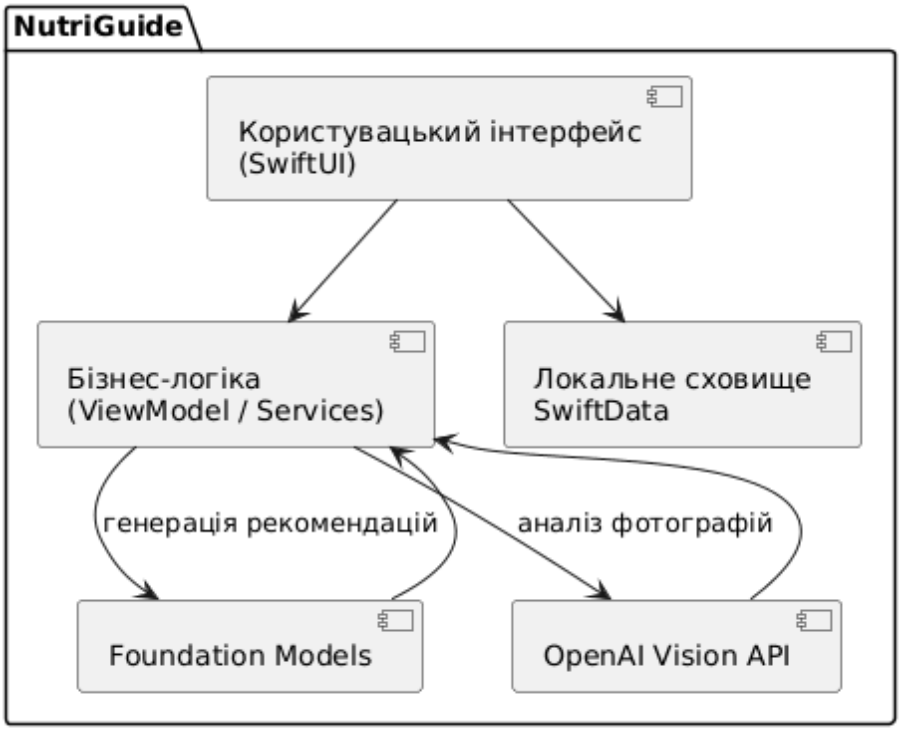


Рисунок 3.6 – Архітектура мобільного застосунку

Окрему роль у архітектурі системи відіграє інтеграція із зовнішніми інтелектуальними сервісами. Для аналізу фотографій їжі застосовується OpenAI Vision API, який виконує розпізнавання страв та визначення їх харчових характеристик. Формування персоналізованих рекомендацій здійснюється за допомогою генеративних моделей Foundation Models, які аналізують історію харчування користувача та генерують індивідуальні поради.

3.4.1 Вибір технологічного стеку та його обґрунтування

Під час проєктування мобільного застосунку NutriGuide важливим етапом є вибір технологічного стеку, який забезпечує реалізацію функціональних можливостей системи та ефективну інтеграцію сучасних інструментів штучного інтелекту. Технологічний стек визначає набір мов програмування, фреймворків і сервісів, що використовуються для розробки

застосунку, а також забезпечує узгоджену взаємодію між усіма компонентами програмної системи.

Для реалізації клієнтської частини застосунку (рисунок 3.7) використовується мова програмування Swift. Інтерфейс користувача реалізовано за допомогою фреймворку SwiftUI, що забезпечує декларативний підхід до побудови інтерфейсів. Для локального зберігання інформації про прийоми їжі використовується технологія SwiftData, яка дозволяє ефективно працювати зі структурованими даними без необхідності використання зовнішніх серверних баз даних.

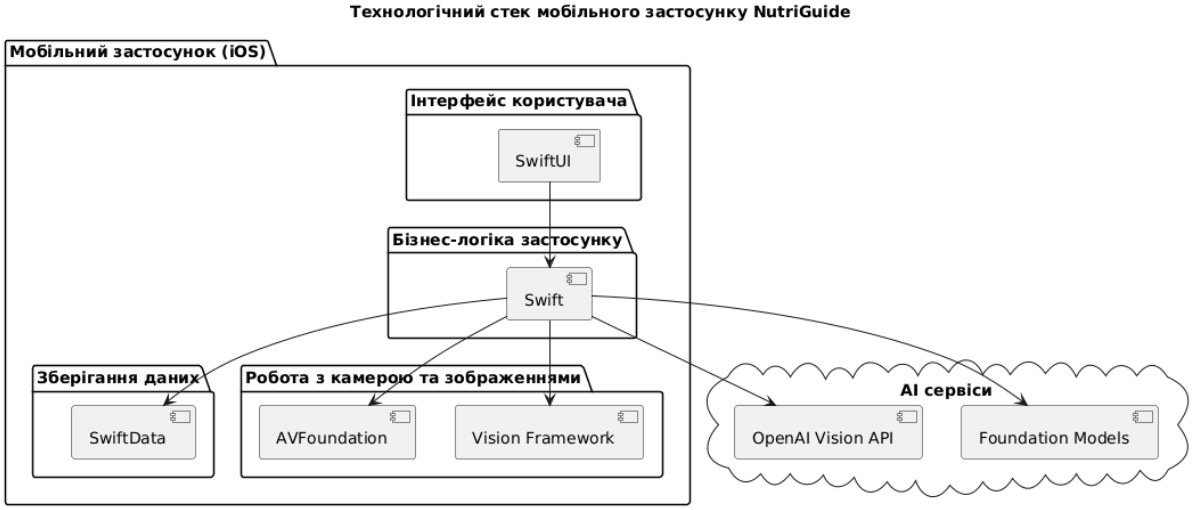


Рисунок 3.7 – Технологічний стек

Окрему частину технологічного стеку становлять інструменти роботи з камерою та обробкою зображень, зокрема фреймворки AVFoundation та Vision Framework. Вони забезпечують отримання фотографій страв та їх підготовку до подальшого аналізу. Для виконання інтелектуальних функцій системи застосовується інтеграція із сучасними AI-сервісами, серед яких OpenAI Vision API для розпізнавання зображень та Foundation Models для генерації персоналізованих рекомендацій щодо харчування.

3.4.2 Архітектурний шаблон

Для побудови внутрішньої структури мобільного застосунку NutriGuide використано архітектурний шаблон MVVM (рисунок 3.8), який є одним із найбільш поширених підходів у сучасній розробці застосунків для платформи iOS.

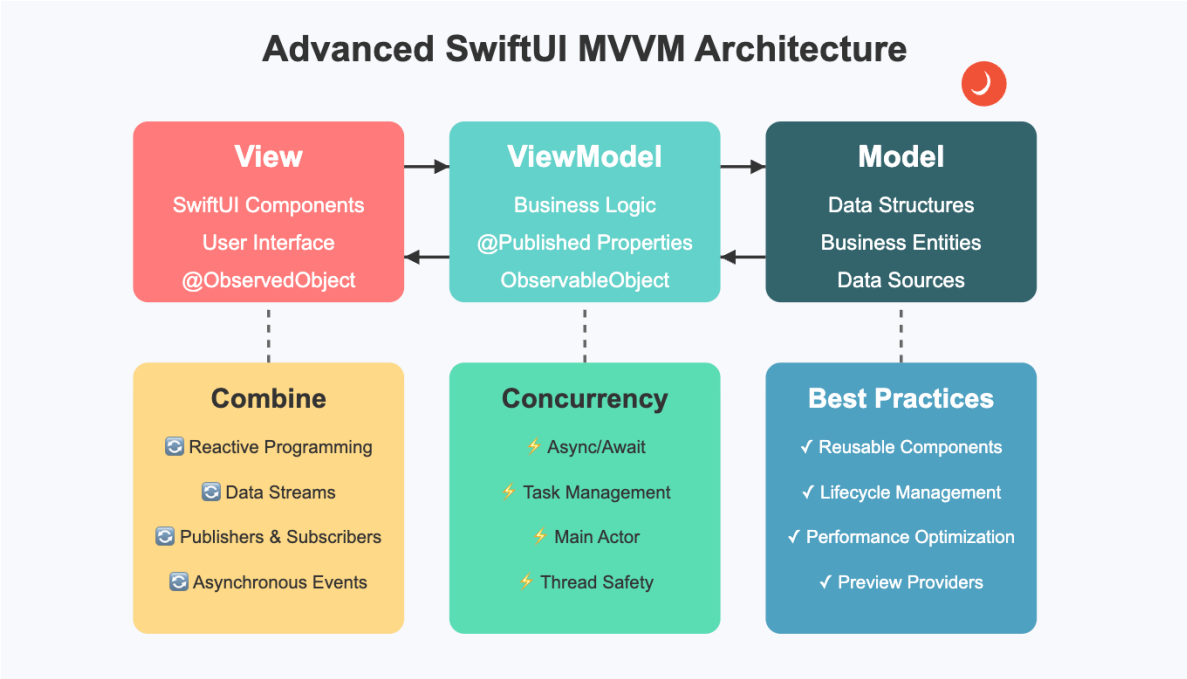


Рисунок 3.8 – MVVM архітектура

Даний шаблон передбачає чітке розділення компонентів інтерфейсу користувача, бізнес-логіки та моделей даних, що дозволяє підвищити модульність системи та спростити подальшу підтримку програмного коду. У межах цього підходу компонент View відповідає за відображення інформації у користувацькому інтерфейсі, Model представляє структури даних застосунку, а ViewModel забезпечує обробку даних та координацію взаємодії між інтерфейсом і бізнес-логікою.

3.4.3 Інтеграція Vision API для аналізу зображень

Однією з ключових функціональних можливостей застосунку NutriGuide є автоматичний аналіз фотографій страв, який реалізується шляхом інтеграції із сервісом OpenAI Vision API (рисунок 3.9). Даний сервіс використовує сучасні моделі комп'ютерного зору для розпізнавання об'єктів на зображенні та визначення їх характеристик.

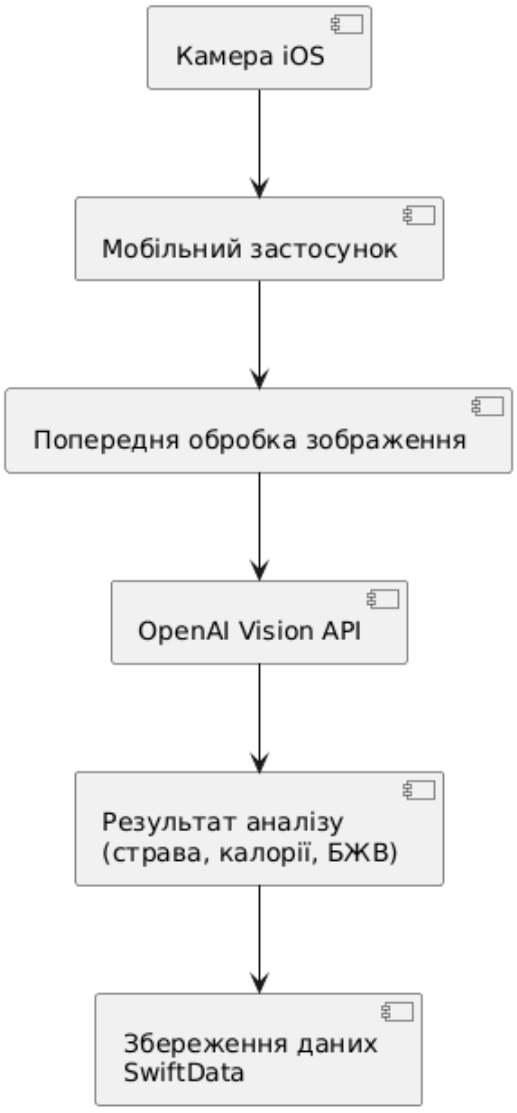


Рисунок 3.9 – Схематичне зображення інтеграції Vision API для аналізу зображень

Завдяки використанню цього інструменту мобільний застосунок може автоматично визначати тип страви, оцінювати її приблизну калорійність та визначати вміст основних макронутрієнтів. Процес інтеграції реалізується у вигляді послідовності етапів, що починається з отримання фотографії страви за допомогою камери мобільного пристрою. Після цього зображення проходить попередню обробку у мобільному застосунку, що може включати оптимізацію розміру та формату зображення.

Далі підготовлене зображення передається до сервісу OpenAI Vision API для аналізу, після чого система отримує результат у вигляді структурованих даних. Отримані результати використовуються для формування запису про прийом їжі та зберігаються у локальному сховищі SwiftData для подальшого аналізу та використання у модулі рекомендацій.

3.4.4 Використання Foundation Models для формування персоналізованих рекомендацій

Важливим елементом інтелектуальної функціональності мобільного застосунку NutriGuide є модуль формування персоналізованих рекомендацій щодо харчування (рисунок 3.10). Для реалізації цієї можливості використовується підхід, заснований на застосуванні сучасних генеративних моделей штучного інтелекту, зокрема Foundation Models. Використання подібних технологій дозволяє перетворити застосунок із простого інструменту фіксації прийомів їжі на систему інтелектуального аналізу раціону.

Основою для формування рекомендацій є історія харчування користувача, яка накопичується у локальному сховищі даних SwiftData. На основі цих даних система виконує аналіз структури раціону, зокрема визначає загальну калорійність харчування, співвідношення макронутрієнтів та частоту споживання певних типів продуктів.

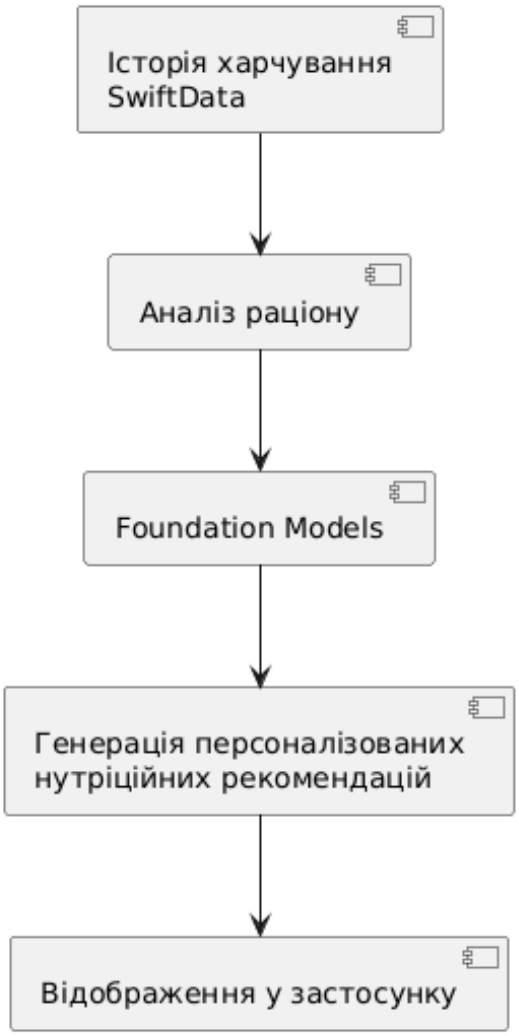


Рисунок 3.10 – Схематичне зображення інтеграції Foundation Models для формування персоналізованих рекомендацій

Після цього узагальнена інформація передається до моделі Foundation Models, яка виконує інтерпретацію отриманих даних та формує рекомендації щодо покращення харчових звичок. Згенеровані рекомендації можуть включати поради щодо оптимізації калорійності раціону, зміни співвідношення білків, жирів і вуглеводів, а також пропозиції щодо включення або обмеження певних груп продуктів. Після завершення генерації рекомендації передаються до мобільного застосунку, де вони відображаються користувачу у зрозумілому текстовому форматі.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Опис технологічного стеку

Для реалізації системи моніторингу харчування на основі аналізу фотозображень страв було обрано сучасний стек технологій, орієнтований на мобільну платформу iOS та інтеграцію інструментів штучного інтелекту. Використані технології забезпечують можливість локального збереження даних, аналізу зображень, формування персоналізованих рекомендацій та побудови сучасного користувацького інтерфейсу.

Основою реалізації мобільного застосунку було обрано мову програмування Swift та фреймворк SwiftUI. Використання SwiftUI дозволяє створювати адаптивний інтерфейс користувача з підтримкою декларативного підходу до побудови екранів. Даний фреймворк забезпечує тісну інтеграцію з екосистемою Apple, підтримує анімації, реактивне оновлення інтерфейсу та оптимізовану роботу на мобільних пристроях.

Для організації архітектури застосунку використовується підхід MVVM. Дана архітектура забезпечує розділення логіки інтерфейсу, бізнес-логіки та моделей даних. Завдяки цьому підвищується підтримуваність програмного коду, спрощується масштабування системи та повторне використання окремих компонентів.

Локальне зберігання інформації про історію харчування реалізовано за допомогою SwiftData. Дана технологія використовується для створення моделей даних, збереження записів про прийоми їжі та швидкого доступу до накопиченої інформації без використання зовнішнього сервера. Застосування SwiftData дозволяє організувати роботу з локальною базою даних із мінімальною кількістю допоміжного коду.

Для аналізу фотозображень страв використовується OpenAI Vision API. Дана технологія забезпечує можливість визначення складу страви, оцінювання калорійності та розрахунку основних нутріційних показників на

основі фотографії. Використання моделі комп’ютерного зору дозволяє автоматизувати процес аналізу харчування та усунути необхідність ручного введення даних користувачем.

Формування персоналізованих рекомендацій реалізовано із застосуванням Foundation Models. Модель аналізує історію харчування користувача за визначений період та формує текстові рекомендації щодо оптимізації раціону відповідно до поставлених цілей харчування. Використання генеративних моделей дозволяє адаптувати рекомендації до індивідуальних особливостей користувача та забезпечити більш гнучкий підхід до аналізу харчування.

Таким чином, обраний технологічний стек забезпечує реалізацію всіх основних функціональних можливостей системи, включаючи аналіз фотозображень, локальне зберігання історії харчування, генерацію рекомендацій та побудову сучасного мобільного інтерфейсу.

4.2 Архітектура програмної системи

Архітектура програмної системи є одним із ключових елементів реалізації мобільного застосунку, оскільки саме вона визначає структуру взаємодії між окремими компонентами системи, організацію обробки даних та принципи роботи функціональних модулів. Правильно спроєктована архітектура дозволяє забезпечити масштабованість застосунку, спрощує підтримку програмного коду та підвищує стабільність роботи системи.

У межах розробленого застосунку архітектура побудована з урахуванням необхідності обробки фотозображень, роботи з моделями штучного інтелекту, локального зберігання інформації та формування персоналізованих рекомендацій. Система включає окремі модулі для роботи з користувацьким інтерфейсом, аналізом зображень, збереженням даних та генерацією результатів нутріційного аналізу.

Для побудови програмної системи використовується архітектурний підхід MVVM, який дозволяє розділити логіку відображення інтерфейсу, бізнес-логіку та моделі даних. Такий підхід спрощує організацію взаємодії між екранами застосунку та забезпечує більш ефективне керування станом інтерфейсу.

Архітектура також враховує необхідність асинхронної взаємодії з зовнішніми сервісами штучного інтелекту, зокрема OpenAI Vision API та Foundation Models. Завдяки цьому система здатна виконувати аналіз фотозображень та генерацію рекомендацій без блокування основного інтерфейсу користувача.

У наступних підрозділах буде детально розглянуто загальну структуру застосунку, організацію архітектури MVVM та взаємодію основних модулів програмної системи.

4.2.1 Загальна структура мобільного застосунку

Структура мобільного застосунку побудована за модульним принципом, що забезпечує розділення функціональності на окремі незалежні компоненти. Такий підхід дозволяє спростити підтримку програмного коду, підвищити масштабованість системи та забезпечити повторне використання окремих елементів застосунку.

Основна структура проєкту включає окремі директорії для функціональних модулів, інфраструктурного рівня, спільних компонентів та ресурсів застосунку (рисунок 4.1). Кожен функціональний модуль відповідає за окрему частину логіки системи, наприклад ведення історії харчування, аналітику або роботу з нутріційними даними.

Директорія Features містить основні функціональні модулі застосунку. Усередині кожного модуля реалізовано окремі моделі даних, ViewModel-компоненти та SwiftUI-представлення.

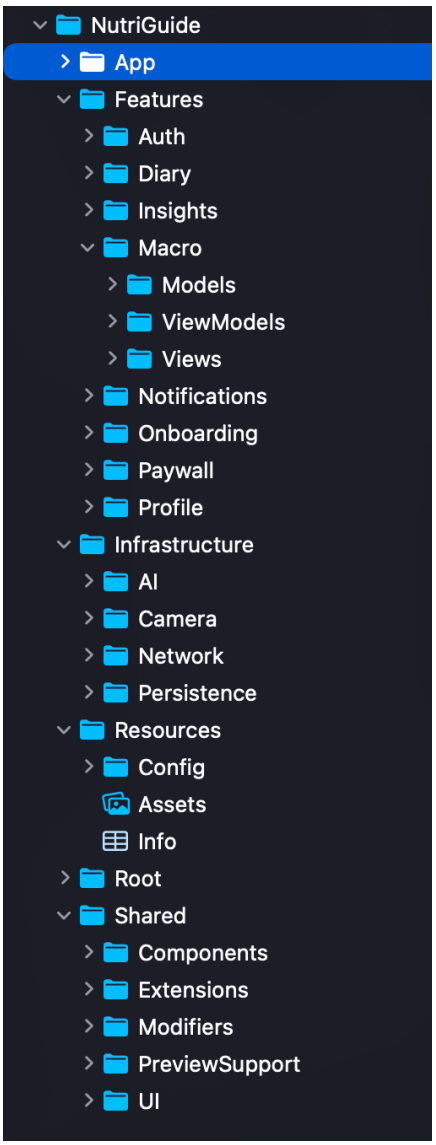


Рисунок 4.1 – Загальна структура мобільного застосунку NutriGuide

Інфраструктурний рівень системи реалізовано у директорії Infrastructure, яка містить модулі роботи зі штучним інтелектом, камерою, мережевими запитами та локальним збереженням даних. Це забезпечує ізоляцію низькорівневої логіки від компонентів користувацького інтерфейсу. Спільні компоненти інтерфейсу, розширення та допоміжні елементи розміщені у директорії Shared. Така організація дозволяє уникнути дублювання програмного коду та забезпечує єдиний стиль побудови інтерфейсу у всіх частинах застосунку.

4.2.2 Організація архітектури MVVM

Для побудови програмної системи було використано архітектурний підхід MVVM, який забезпечує розділення логіки інтерфейсу користувача, бізнес-логіки та моделей даних. Використання даної архітектури дозволяє спростити підтримку застосунку, покращити повторне використання компонентів та забезпечити більш структуровану організацію програмного коду.

У межах архітектури MVVM кожен функціональний модуль містить окремі директорії для моделей даних, ViewModel-компонентів та SwiftUI-представлень (рисунок 4.2). Такий підхід дозволяє ізолювати окремі рівні застосунку та мінімізувати залежності між ними.

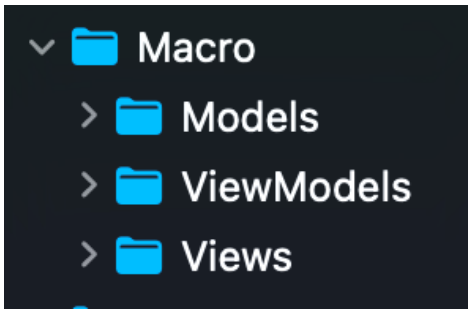


Рисунок 4.2 – Організація структури модуля відповідно до архітектури MVVM

Рівень Model відповідає за опис структури даних та взаємодію із локальним сховищем. У даному шарі реалізовано моделі прийомів їжі, нутріційних показників та історії харчування користувача. Моделі використовуються для збереження та обробки інформації всередині системи.

Рівень View містить SwiftUI-компоненти, що відповідають за відображення інтерфейсу користувача. Даний шар реалізує екрани аналізу страв, перегляду історії харчування та відображення статистики й

рекомендацій. Усі елементи інтерфейсу оновлюються відповідно до змін стану даних.

Рівень ViewModel забезпечує зв’язок між інтерфейсом користувача та моделями даних. Саме у ViewModel реалізовано логіку аналізу фотозображень, виконання мережевих запитів до OpenAI Vision API, обробку результатів нутріційного аналізу та підготовку даних для відображення в інтерфейсі.

Використання архітектури MVVM дозволяє забезпечити більш гнучку організацію програмної системи, а також спрощує подальше масштабування функціональності застосунку та підтримку програмного коду.

4.2.3 Взаємодія основних модулів системи

Програмна система NutriGuide побудована за принципом взаємодії окремих функціональних модулів, кожен із яких відповідає за виконання визначеної частини логіки застосунку.

Взаємодія між основними компонентами системи організована відповідно до архітектури MVVM та передбачає послідовну передачу даних між рівнями View, ViewModel та Infrastructure (рисунок 4.3). Користувацький інтерфейс відповідає за взаємодію з користувачем та відображення результатів аналізу, тоді як ViewModel виконує обробку стану застосунку та координує виконання основної бізнес-логіки.

Після фотографування страви інтерфейс користувача передає зображення до ViewModel, де виконується підготовка даних та запуск процесу аналізу.

Далі ViewModel взаємодіє з інфраструктурними компонентами системи, що відповідають за обробку зображень, мережеві запити та взаємодію з моделями штучного інтелекту.

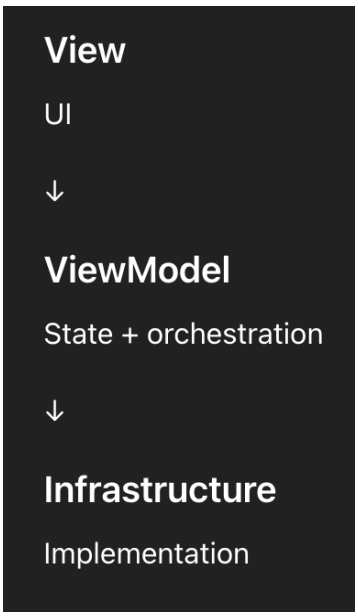


Рисунок 4.3 – Схема взаємодії основних модулів програмної системи

Інфраструктурний рівень реалізує низькорівневу логіку роботи системи, включаючи інтеграцію з OpenAI Vision API, обробку результатів аналізу та локальне збереження інформації у SwiftData. Після завершення аналізу результати повертаються до ViewModel, де виконуються додаткові обчислення та підготовка даних для подальшого відображення у користувацькому інтерфейсі.

Подібна організація взаємодії між модулями дозволяє ізолювати бізнес-логіку від інтерфейсного рівня, забезпечує більш зрозумілу структуру застосунку та спрощує подальше розширення функціональності системи.

4.3 Реалізація моделей даних та локального зберігання

Для забезпечення роботи системи моніторингу харчування необхідним є збереження інформації про прийоми їжі, результати нутріційного аналізу та історію харчування користувача. З цією метою у застосунку реалізовано локальне сховище даних на технології SwiftData.

Основною моделлю системи є модель прийому їжі, яка містить інформацію про назву страви, калорійність, білки, жири, вуглеводи, дату створення запису та фотографію страви. Реалізацію моделі даних наведено у лістингу 4.1.

Лістинг 4.1 – Програмний код реалізації моделі даних для збереження інформації про прийом їжі

```
@Model
final class MealRecord {
    var id: UUID
    var date: Date
    var title: String
    var calories: Double
    var proteins: Double
    var fats: Double
    var carbohydrates: Double
    var imageData: Data?
    init(
        title: String,
        calories: Double,
        proteins: Double,
        fats: Double,
        carbohydrates: Double,
        imageData: Data? ) {
        self.id = UUID()
        self.date = Date()
        self.title = title
        self.calories = calories
        self.proteins = proteins
        self.fats = fats
        self.carbohydrates = carbohydrates
        self.imageData = imageData }
}
```

Використання анотації @Model дозволяє інтегрувати модель безпосередньо у механізм локального збереження SwiftData та автоматизувати роботу з базою даних.

Модель містить основні нутріційні показники, що використовуються для подальшого аналізу харчування користувача. Також у структурі передбачено збереження фотозображення прийому їжі у форматі Data, що дозволяє відображати історію харчування разом із фотографіями страв.

Для додавання нових записів до локальної бази даних реалізовано окрему функцію збереження результатів аналізу, приклад якої наведено у лістингу 4.2.

Лістинг 4.2 – Програмний код збереження результату нутріційного аналізу у локальну базу даних

```
func saveMeal(
    title: String,
    calories: Double,
    proteins: Double,
    fats: Double,
    carbohydrates: Double,
    imageData: Data?
) {
    let meal = MealRecord(
        title: title,
        calories: calories,
        proteins: proteins,
        fats: fats,
        carbohydrates: carbohydrates,
        imageData: imageData
    )
    modelContext.insert(meal)
    try? modelContext.save()
}
```

Після отримання результатів аналізу фотозображення створюється новий об’єкт типу MealRecord, який додається до локального контексту даних та зберігається у SwiftData.

Використання локального збереження забезпечує швидкий доступ до історії харчування користувача та дозволяє виконувати аналіз накопичених даних без необхідності постійного підключення до мережі. Крім того, локальна обробка та зберігання інформації підвищують рівень конфіденційності персональних даних користувача.

Таким чином, реалізована система моделей даних та локального зберігання забезпечує основу для роботи всіх основних функціональних модулів застосунку, включаючи аналіз харчування, формування статистики та генерацію персоналізованих рекомендацій.

4.4 Реалізація модуля аналізу фотозображень страв

Одним із ключових компонентів системи NutriGuide є модуль аналізу фотозображень страв, який забезпечує автоматичне визначення нутріційних характеристик їжі на основі фотографії.

Робота модуля включає отримання фотозображення від користувача, попередню обробку зображення, передавання даних до моделі комп’ютерного зору та формування результатів нутріційного аналізу.

Для реалізації даного функціоналу використовується OpenAI Vision API, що дозволяє визначати склад страви, калорійність та основні показники білків, жирів і вуглеводів.

Отримані результати надалі використовуються для формування історії харчування користувача, побудови статистики та генерації персоналізованих рекомендацій щодо оптимізації раціону, що є фундаментом для мого застосунку.

4.4.1 Отримання фотозображення прийому їжі

Першим етапом роботи модуля аналізу харчування є отримання фотозображення прийому їжі від користувача. Для реалізації даного функціоналу у застосунку використовується компонент PhotosPicker, який забезпечує доступ до галереї пристрою та дозволяє обрати необхідне зображення для подальшого аналізу. Реалізацію механізму вибору фотозображення наведено у лістингу 4.3. Після вибору фотографії застосунок асинхронно завантажує дані зображення у форматі Data, після чого створюється об’єкт типу UIImage, який використовується для подальшої обробки та аналізу.

Лістинг 4.3 – Програмний код отримання фотозображення прийому їжі з інтерфейсу користувача

```
PhotosPicker(
    selection: $selectedPhoto,
    matching: .images
) {
    Label("Додати фото страви", systemImage: "camera.fill")
}
.onChange(of: selectedPhoto) {
    Task {
        if let data = try? await
selectedPhoto?.loadTransferable(type: Data.self) {
            selectedImageData = data
            selectedImage = UIImage(data: data)
        }
    }
}
```

Використання асинхронного підходу дозволяє уникнути блокування користувацького інтерфейсу під час завантаження зображення. Крім того,

така реалізація забезпечує швидку взаємодію користувача із системою та створює основу для подальшого передавання фотографії до модуля комп’ютерного зору.

Отримане фотозображення надалі використовується для виконання попередньої обробки, оптимізації розміру та формування запиту до OpenAI Vision API.

4.4.2 Попередня обробка зображення

Після отримання фотозображення виконується етап попередньої обробки, необхідний для оптимізації роботи моделі комп’ютерного зору та зменшення обсягу даних, що передаються до OpenAI Vision API. Даний етап дозволяє підвищити стабільність аналізу зображень, а також зменшити час обробки та мережеве навантаження.

У процесі попередньої обробки виконується зміна розміру зображення до фіксованого формату та його подальше стиснення у формат JPEG. Такий підхід дозволяє забезпечити однакові параметри вхідних даних незалежно від початкової роздільної здатності фотографії. Програмний код попередньої обробки фотозображення перед передаванням до моделі наведено у лістингу 4.4.

Лістинг 4.4 – Програмний код попередньої обробки фотозображення перед передаванням до моделі

```
func prepareImageData(from image: UIImage) -> Data? {
    let targetSize = CGSize(width: 768, height: 768)
    let renderer = UIGraphicsImageRenderer(size: targetSize)
    let resizedImage = renderer.image { _ in image.draw(in: CGRect(origin: .zero, size: targetSize)) }
    return resizedImage.jpegData(compressionQuality: 0.8) }
```

У наведеній реалізації використовується UIGraphicsImageRenderer, який створює нове зображення із заданими параметрами розміру. Після цього виконується конвертація результату у формат JPEG із використанням коефіцієнта стиснення. Отримані дані надалі використовуються для формування запиту до моделі комп'ютерного зору.

4.4.3 Інтеграція OpenAI Vision API для визначення складу страви

Після виконання попередньої обробки фотозображення здійснюється передавання даних до моделі комп'ютерного зору OpenAI Vision API. Даний етап є основою роботи всієї системи, оскільки саме модель штучного інтелекту виконує визначення складу страви та оцінювання її нутріційних характеристик. Для інтеграції з OpenAI Vision API у застосунку реалізовано окремий сервіс, який формує мережевий запит, передає фотозображення у форматі Base64 та отримує структурований результат аналізу. Програмний код формування запиту до OpenAI Vision API для аналізу страви наведено у лістингу 4.5.

Лістинг 4.5 – Програмний код формування запиту до OpenAI Vision API для аналізу страви

```
    func analyzeMealImage(_ imageData: Data) async throws -> 
NutritionResult {
        let base64Image = imageData.base64EncodedString()
        let request = VisionMealRequest(
            model: "gpt-4o-mini",
            image: base64Image,
            prompt: "")
        let response = try await openAIService.send(request)
        return try JSONDecoder().decode(
            NutritionResult.self,
            from: response.data)}
```

У наведеній реалізації фотозображення спочатку конвертується у формат Base64, після чого формується запит до моделі gpt-4o-mini. У тексті запиту задається інструкція для моделі щодо визначення назви страви, калорійності та показників білків, жирів і вуглеводів. Після отримання відповіді від OpenAI Vision API результат декодується, щоб потім отримати відображення даних у користувацькому інтерфейсі та збереження інформації в історії харчування користувача.

4.4.4 Формування результатів нутріційного аналізу

Після отримання відповіді від OpenAI Vision API виконується формування результатів нутріційного аналізу для подальшого використання у застосунку. На даному етапі система перетворює отримані дані у структурований формат, який використовується для відображення інформації в інтерфейсі користувача, збереження історії харчування та подальшого аналізу раціону. Для представлення результатів аналізу використовується окрема модель даних NutritionResult, що містить основні нутріційні показники страви. Програмний код моделі результату нутріційного аналізу фотозображення наведено у лістингу 4.6.

Лістинг 4.6 – Програмний код моделі результату нутріційного аналізу фотозображення

```
struct NutritionResult: Decodable {
    let title: String
    let calories: Double
    let proteins: Double
    let fats: Double
    let carbohydrates: Double
    let confidence: Double
}
```

У структурі передбачено збереження назви страви, калорійності, показників білків, жирів і вуглеводів, а також коефіцієнта достовірності результату аналізу.

Використання протоколу Decodable дозволяє автоматично декодувати JSON-відповідь, отриману від OpenAI Vision API.

Після декодування результатів виконується підготовка даних для подальшого відображення в інтерфейсі користувача. Програмний код формування результатів аналізу наведено у лістингу 4.7.

Лістинг 4.7 – Програмний код формування результатів аналізу для подальшого відображення

```
func applyAnalysisResult(_ result: NutritionResult) {
    mealTitle = result.title
    calories = String(format: "%.0f", result.calories)
    proteins = String(format: "%.1f", result.proteins)
    fats = String(format: "%.1f", result.fats)
    carbohydrates = String(format: "%.1f",
result.carbohydrates)
    confidence = result.confidence
    isResultPresented = true
}
```

У наведеній реалізації отримані числові значення додатково форматуються для коректного відображення в інтерфейсі застосунку. Після завершення обробки змінюється стан isResultPresented, що сигналізує системі про готовність результатів до відображення користувачу.

Таким чином, реалізований механізм формування результатів забезпечує перетворення відповіді моделі штучного інтелекту у зручний для використання формат та створює основу для подальшої роботи модулів аналітики й персоналізованих рекомендацій.

4.5 Реалізація користувацького інтерфейсу

Користувацький інтерфейс мобільного застосунку реалізовано за допомогою фреймворку SwiftUI, який забезпечує побудову адаптивних та реактивних екранів для взаємодії користувача із системою. Основною задачею інтерфейсу є забезпечення швидкого доступу до функцій аналізу страв, перегляду історії харчування та відображення статистики нутріційних показників.

Після завершення аналізу фотозображення користувач отримує інформацію про калорійність страви та показники білків, жирів і вуглеводів. Для відображення результатів аналізу використовується окремий інтерфейсний компонент, реалізацію якого наведено у лістингу 4.8.

Лістинг 4.8 – Програмний код відображення результату аналізу прийому їжі в інтерфей

```
VStack(alignment: .leading, spacing: 12) {
    Text(mealTitle)
        .font(.title2)
    Text("Калорійність: \(calories) ккал")
    Text("Білки: \(proteins) г")
    Text("Жири: \(fats) г")
    Text("Вуглеводи: \(carbohydrates) г")
    Button("Зберегти прийом їжі") {
        viewModel.saveCurrentMeal()
    }
}
.padding()
.background(.thinMaterial)
.clipShape(RoundedRectangle(cornerRadius: 20))
```

У наведеному компоненті реалізовано відображення основних нутріційних показників та кнопку збереження результату аналізу у локальну

базу даних. Для оформлення інтерфейсу використовується ефект Material, що забезпечує сучасний вигляд елементів інтерфейсу відповідно до стилістики iOS-застосунків.

Однією з основних функцій системи є ведення історії харчування користувача. Для реалізації даного функціоналу використовується компонент List, який дозволяє динамічно відображати список збережених прийомів їжі. Програмний код виведення історії харчування наведено у лістингу 4.9.

Лістинг 4.9 – Програмний код виведення історії харчування користувача

```
List(meals) { meal in
    VStack(alignment: .leading, spacing: 6) {
        Text(meal.title)
            .font(.headline)
        Text("\(meal.calories, specifier: "%.0f") ккал")
            .font(.subheadline)
        Text("Б: \(meal.proteins, specifier: "%.1f") г  Ж: \(meal.fats, specifier: "%.1f") г  В: \(meal.carbohydrates, specifier: "%.1f") г")
            .font(.caption)
    }
}
```

У кожному елементі списку відображається назва страви, калорійність та співвідношення білків, жирів і вуглеводів. Такий підхід дозволяє користувачу швидко переглядати накопичену історію харчування та аналізувати власний раціон.

Крім перегляду окремих прийомів їжі, застосунок також забезпечує відображення загальної статистики харчування за день. Для цього реалізовано окремий блок, приклад якого наведено у лістингу 4.10.

Лістинг 4.10 – Програмний код відображення добової статистики харчування

```
    VStack(alignment: .leading, spacing: 10) {
        Text("Статистика за день")
            .font(.headline)
        Text("Калорії: \(dailyCalories, specifier: "%.0f") ккал")
        Text("Білки: \(dailyProteins, specifier: "%.1f") г")
        Text("Жири: \(dailyFats, specifier: "%.1f") г")
        Text("Вуглеводи: \(dailyCarbohydrates, specifier: "%.1f") г")
    }
    .frame(maxWidth: .infinity, alignment: .leading)
    .padding()
    .background(.regularMaterial)
    .clipShape(RoundedRectangle(cornerRadius: 18))
```

Реалізований компонент дозволяє відображати сумарні показники калорійності та нутрієнтів за поточний день. Подібна візуалізація забезпечує зручний моніторинг харчування та створює основу для подальшого аналізу раціону користувача і формування персоналізованих рекомендацій.

У межах реалізації користувацького інтерфейсу було розроблено набір основних екранів мобільного застосунку, які забезпечують взаємодію користувача із системою моніторингу харчування. Інтерфейс побудований відповідно до сучасних рекомендацій Apple Human Interface Guidelines та орієнтований на забезпечення простоти використання, швидкого доступу до функцій аналізу харчування та зручного перегляду статистики.

Основний екран застосунку містить інформацію про добову калорійність, коротку статистику нутрієнтів та список останніх прийомів їжі. Екран історії харчування реалізовано у форматі календарного перегляду, що дозволяє користувачу переглядати дані за окремі дні та

аналізувати зміни раціону з часом. Екран аналітики містить візуалізацію співвідношення білків, жирів і вуглеводів, а також персоналізовані рекомендації щодо оптимізації харчування. Крім того, у застосунку реалізовано окремий екран профілю користувача, де зберігаються персональні параметри та цілі харчування (рисунок 4.4).



Рисунок 4.4 – Основні екрани мобільного застосунку NutriGuide

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було досягнуто поставленої мети – розроблено систему моніторингу харчування на основі аналізу фотозображень страв та формування персоналізованих рекомендацій. У межах роботи було проведено аналіз предметної галузі, досліджено сучасні підходи до автоматизованого визначення нутріційних характеристик їжі, а також розглянуто особливості використання методів комп’ютерного зору та генеративного штучного інтелекту у задачах аналізу харчування.

У теоретичній частині роботи було досліджено сучасні системи моніторингу харчування, методи розпізнавання об’єктів на фотозображеннях та підходи до формування персоналізованих рекомендацій. Проведений аналіз дозволив визначити основні недоліки існуючих рішень, зокрема необхідність ручного введення даних та обмежені можливості персоналізації рекомендацій.

У процесі проєктування системи було сформовано архітектуру мобільного застосунку, визначено структуру основних модулів та реалізовано взаємодію між компонентами системи. Для побудови застосунку використано архітектуру MVVM, що забезпечило розділення бізнес-логіки, інтерфейсу користувача та моделей даних. Також було реалізовано локальне збереження історії харчування за допомогою SwiftData.

У практичній частині роботи реалізовано мобільний застосунок NutriGuide для платформи iOS із використанням SwiftUI. Система забезпечує фотографування прийомів їжі, автоматичний аналіз фотозображень за допомогою OpenAI Vision API, визначення калорійності та показників білків, жирів і вуглеводів, ведення історії харчування та формування персоналізованих рекомендацій на основі накопичених даних користувача.

У результаті реалізації системи було досягнуто автоматизації процесу аналізу харчування без необхідності ручного введення інформації про склад страв. Використання моделей комп’ютерного зору дозволило забезпечити швидке отримання нутріційних показників на основі фотозображення, а інтеграція генеративних моделей – формування рекомендацій щодо оптимізації раціону відповідно до цілей користувача.

Розроблений застосунок може бути використаний як персональний інструмент контролю харчування, а також як основа для подальшого розвитку систем підтримки здорового способу життя. Практична цінність роботи полягає у поєднанні методів комп’ютерного зору, генеративного штучного інтелекту та мобільних технологій у межах єдиної програмної системи.

Подальший розвиток роботи може бути пов’язаний із підвищенням точності аналізу складних багатокомпонентних страв, інтеграцією моделей локального виконання на мобільних пристроях, підтримкою розпізнавання ваги порцій та розширенням системи рекомендацій із використанням більш глибокого аналізу поведінки користувача. Крім того, перспективним напрямком є інтеграція застосунку з фітнес-трекерами та медичними системами моніторингу здоров’я.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2016. P. 770–778.

2. Krizhevsky A., Sutskever I., Hinton G. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems.* 2012. P. 1097–1105.

3. Goodfellow I., Bengio Y., Courville A. Deep Learning. *Cambridge: MIT Press*, 2016. 775 p.

4. Chollet F. Deep Learning with Python. *Shelter Island: Manning Publications*, 2021. 504 p.

5. Szeliski R. Computer Vision: Algorithms and Applications. *Cham: Springer*, 2022. 979 p.

6. Apple Inc. Vision. *Apple Developer Documentation.* URL: https://developer.apple.com/documentation/vision (date of access: 01.02.2026).

7. Apple Inc. Core ML. *Apple Developer Documentation.* URL: https://developer.apple.com/documentation/coreml (date of access: 03.02.2026).

8. Apple Inc. Create ML. *Apple Developer Documentation.* URL: https://developer.apple.com/documentation/createml (date of access: 06.02.2026).

9. Apple Inc. SwiftUI. *Apple Developer Documentation.* URL: https://developer.apple.com/documentation/swiftui (date of access: 10.02.2026).

10. Apple Inc. Core ML Models. *Apple Developer.* URL: https://developer.apple.com/machine-learning/models/ (date of access: 12.02.2026).

11. Apple Inc. Apple Intelligence. *Apple.* URL: https://www.apple.com/apple-intelligence/ (date of access: 15.02.2026).

12. OpenAI. Images and Vision. *OpenAI Platform Documentation.* URL: https://platform.openai.com/docs/guides/vision (date of access: 18.02.2026).

13. OpenAI. Models. *OpenAI Platform Documentation.* URL: https://platform.openai.com/docs/models (date of access: 20.02.2026).

14. Howard A., Sandler M., Chu G. et al. Searching for MobileNetV3. *Proceedings of the IEEE International Conference on Computer Vision.* 2019. P. 1314–1324.

15. Tan M., Le Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the International Conference on Machine Learning.* 2019. P. 6105–6114.

16. Dosovitskiy A., Beyer L., Kolesnikov A. et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations.* 2021. P. 1–21.

17. Kaggle Inc. Datasets. *Kaggle.* URL: https://www.kaggle.com/datasets (date of access: 25.02.2026).

18. ETH Zurich. Food-101. *Computer Vision Lab.* URL: https://data.vision.ee.ethz.ch/cvl/datasets_extra/food-101/ (date of access: 03.03.2026).

19. PyTorch Foundation. Tutorials. *PyTorch Documentation.* URL: https://pytorch.org/tutorials/ (date of access: 10.03.2026).

20. TensorFlow. Image Classification. *TensorFlow Core.* URL: https://www.tensorflow.org/tutorials/images/classification (date of access: 18.03.2026).