

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
Факультет комп'ютерно-інтегрованих технологій, мехатроніки і робототехніки
Кафедра інформаційно-вимірювальних технологій

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи
бакалавра

на тему: **Комп'ютеризована інформаційно-вимірювальна система для визначення параметрів та розпізнавання об'єктів за їх зображеннями**

Виконав: студент 2 курсу, групи ІВТК-4

Спеціальності: 175 «Інформаційно-
вимірювальні технології»

Ярошевський О. В.

(прізвище та ініціали)

Керівник: доцент кафедри ІВТ

кандидат технічних наук, доцент

Чепюк Л.О.

(науковий ступінь, вчене звання, посада, прізвище та ініціали)

Рецензент: _____

(науковий ступінь, вчене звання, посада, прізвище та ініціали)

Житомир – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
Факультет комп'ютерно-інтегрованих технологій, мехатроніки і робототехніки
Кафедра інформаційно-вимірювальних технологій

Спеціальність 175 «Інформаційно-вимірювальні технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри інформаційно-
вимірювальних технологій

д.т.н., професор Подчашинський Ю.О.

« _____ » _____ 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу бакалавра

Студент _____ Ярошевський Олександр В'ячеславович

1. Тема роботи: Комп'ютеризована інформаційно-вимірювальна система для визначення параметрів та розпізнавання об'єктів за їх зображеннями

Керівник роботи: доцент кафедри ІВТ кандидат технічних наук, доцент
Чепюк Л.О.

затверджена наказом по університету від « 04 » квітня 2025 р. № 132/с

2. Термін здачі закінченої роботи на кафедру « 10 » червня 2025р.

3. Вихідні дані до роботи Об'єкт вимірювання — мобільна система, що працює з фото у форматі HEIC

роздільною здатністю не нижче 1920×1080 пікселів.

Зображення надходять із камери або галереї для подальшого аналізу об'єктів та генерації описів.

4. Зміст пояснювальної записки (перелік питань, які підлягають розробці)

розробка структурної схеми та архітектури програмного забезпечення

розробка структурної архітектури програмного забезпечення, розробка

алгоритму роботи системи, обрахунок точності оцінки параметрів об'єктів

5. Перелік графічного матеріалу (з точним визначенням обов'язкових креслень)

структурна схема інформаційно-вимірювальної системи,

блок-схема алгоритму роботи мобільного застосунку та алгоритму роботи

системі

6. Консультанти з кваліфікаційної роботи бакалавра із зазначенням розділів роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Узгодження технічного завдання	25.02.25 – 15.03.25	Виконано
2	Огляд та аналіз методів комп'ютерного зору та обробки зображень	25.02.25 – 15.03.25	Виконано
3	Розробка структурної схеми	16.03.25 – 15.04.25	Виконано
4	Розробка архітектури програмного забезпечення мобільного додатку	16.04.25 – 24.04.25	Виконано
5	Розробка інтерфейсу користувача та UX-дизайну	25.04.25 – 09.05.25	Виконано
6	Розробка алгоритму роботи та програмної реалізації системи	10.05.25 – 24.05.25	Виконано
7	Оцінка метрологічних характеристик системи	25.05.25 – 03.06.25	Виконано
8.	Оцінка надійності роботи системи	04.06.25 – 15.06.25	Виконано
9	Оформлення пояснювальної записки та презентаційного матеріалу	03.06.25 – 17.06.25	Виконано

Студент

(підпис)

Керівник роботи

(підпис)

Ярошевський О. В.

(прізвище та ініціали)

Чепюк Л. О.

(прізвище та ініціали)

АНОТАЦІЯ

У даній кваліфікаційній роботі розроблено інформаційно-вимірювальну систему у вигляді мобільного iOS-додатку для розпізнавання об'єктів на фотографіях. Використано бібліотеки Vision для аналізу зображень та FoundationModels для генерації пояснень на основі параметрів об'єктів. Система дозволяє користувачу додавати фото, отримувати структуровані результати вимірювань та експортувати їх у форматі CSV. Розроблена система орієнтована на локальну обробку даних без залежності від хмарних сервісів, що підвищує безпеку та швидкодію.

Ключові слова: розпізнавання об'єктів, мобільна вимірювальна система, комп'ютерний зір, iOS-додаток, Vision, FoundationModels.

ABSTRACT

This qualification work presents the development of an information-measuring system implemented as a mobile iOS application for object recognition in photographs. The system uses Vision framework for image analysis and FoundationModels for generating explanations based on detected object parameters. Users can add photos, receive structured measurement results, and export data in CSV format. The system performs local data processing, enhancing security and performance by avoiding cloud dependencies.

Keywords: object recognition, mobile measuring system, computer vision, iOS application, Vision, FoundationModels.

РЕФЕРАТ

Кваліфікаційна робота представлена пояснювальною запискою обсягом близько 80 сторінок формату А4, яка містить вступ, основну частину, що складається з восьми розділів, висновки, список використаних джерел та додатки. Темою роботи є комп'ютеризована інформаційно-вимірювальна система для визначення параметрів та розпізнавання об'єктів за їх зображеннями.

Перший розділ визначає основні завдання та вимоги до системи, зокрема умови експлуатації, функціональні характеристики, надійність і вимоги до інтерфейсу користувача. Другий розділ містить огляд сучасних методів комп'ютерного зору, аналіз бібліотек Vision та FoundationModels, а також обґрунтування вибору технологій для реалізації проєкту. У третьому розділі наведено аналіз існуючих мобільних систем розпізнавання об'єктів, а також розроблено структурну схему інформаційно-вимірювальної системи.

Четвертий розділ описує архітектуру програмного забезпечення та основні компоненти мобільного додатку.

П'ятий розділ присвячений розробці інтерфейсу користувача та UX-дизайну.

У шостому розділі викладено алгоритм роботи системи: від додавання фотографії до отримання результатів аналізу та їх експорту у формат CSV.

Сьомий розділ містить оцінку метрологічних характеристик системи, зокрема точності розпізнавання об'єктів.

Восьмий розділ присвячений оцінці надійності роботи застосунку та забезпеченню безпеки користувача.

Завершує роботу комплексне оформлення пояснювальної записки та презентаційних матеріалів, що ілюструють результати розробки.

ЗМІСТ

	стор
ВСТУП	8
1. ТЕХНІЧНЕ ЗАВДАННЯ	9
2. ОГЛЯД ТА АНАЛІЗ МЕТОДІВ ТА СИСТЕМ ОБРОБКИ ЗОБРАЖЕНЬ.....	10
2.1. Застосування комп'ютерного зору у мобільних вимірювальних системах.....	10
2.2. Технології розпізнавання об'єктів на основі машинного навчання.....	11
2.3. Висновки до розділу	20
3. РОЗРОБКА СТРУКТУРНОЇ СХЕМИ	21
3.1. Структурна схема інформаційно-вимірювальної системи.....	24
3.2. Взаємозв'язок підсистем додатку.....	25
4. РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ....	26
4.1. Вибір технологічного стека.....	26
4.2. Вибір архітектури застосунку.....	28
4.3. Опис основних модулів.....	32
4.4. Організація передачі даних між модулями.....	37
4.5. Інтеграція Core ML / FoundationModels.....	38
4.6. Розрахунок необхідних ресурсів пристрою.....	40

					МІВТ.480.024.036 – ПЗ			
<i>Вим</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Розроб.		Ярошевський О.В.			Комп'ютеризована інформаційно-вимірювальна система для визначення параметрів та розпізнавання об'єктів за їх зображеннями Пояснювальна записка	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
Перевір.		Чепюк Л.О.					6	67
Реценз.						Державний університет «Житомирська політехніка», гр. ІВТК-4		
Н. Контр.		Чепюк Л.О.						
Затверд.		Подчашинський Ю						

	стор
4.7. Опис взаємодії користувача з додатком	
5. КОНСТРУКТОРСЬКА ЧАСТИНА	43
5.1. Дизайн та компонування головного екрану.....	43
5.2. UX-логіка переходів між екранами.....	47
6. РОЗРОБКА АЛГОРИТМУ ТА ПРОГРАМИ РОБОТИ СИСТЕМИ...	49
7. ОЦІНКА МЕТРОЛОГІЧНИХ ХАРАКТЕРИСТИК.....	54
8. ОЦІНКА НАДІЙНОСТІ РОБОТИ СИСТЕМИ.....	57
9. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОХОРОНА ПРАЦІ.....	61
СПИСОК ВИКОРИСТАНИХ.....	65
ДОДАТКИ	67

ВСТУП

Сучасний розвиток мобільних технологій, штучного інтелекту та комп'ютерного зору відкриває дедалі ширші можливості для створення інтелектуальних інформаційно-вимірювальних систем нового покоління. Поява високопродуктивних процесорів у мобільних пристроях, підтримка апаратного прискорення обчислень, а також розширення екосистеми інструментів розробки дозволяють реалізовувати складні обчислювальні задачі прямо на смартфоні або планшеті. Зокрема, одним із найбільш перспективних напрямів у цьому контексті є розробка застосунків, здатних виконувати автономний аналіз візуальної інформації без потреби в зовнішньому сервері чи хмарному сервісі. Такий підхід значно підвищує конфіденційність, прискорює обробку та розширює сферу застосування подібних рішень у реальному часі.

У межах цієї парадигми особливої актуальності набуває розробка систем, що інтегрують можливості комп'ютерного зору, локального машинного навчання та генеративного моделювання тексту. Застосування таких підходів дозволяє створити повноцінну мобільну платформу, яка не лише розпізнає об'єкти на зображенні, а й формує осмислене пояснення щодо виявлених даних у зручному для користувача форматі. Це відкриває нові можливості в таких сферах, як мобільна діагностика, освіта, цифрова експертиза, логістика, дослідницькі задачі та побутове використання. Крім того, локальна обробка є ключовою перевагою з точки зору безпеки та захисту персональних даних, що відповідає сучасним вимогам до конфіденційності.

Метою даної кваліфікаційної роботи є розробка мобільної інформаційно-вимірювальної системи у вигляді iOS-застосунку, здатного здійснювати повноцінне локальне розпізнавання об'єктів на зображеннях, аналіз отриманих параметрів та генерацію пояснень за допомогою технологій штучного інтелекту. Основними інструментами є фреймворк Vision, що використовується для виявлення та класифікації вмісту зображення, а також

FoundationModels — для побудови природномовного опису побаченого. Уся обробка виконується на пристрої користувача, без підключення до інтернету, що забезпечує стабільність, високу продуктивність і незалежність від зовнішньої інфраструктури.

У межах реалізації було створено застосунок, який дозволяє користувачу завантажити або зробити фото, отримати результати розпізнавання, переглянути пояснення до зображення у текстовій формі, а також експортувати дані у форматі CSV для подальшої аналітики. Такий підхід робить систему не лише технічно ефективною, а й зручною у використанні, адаптованою до різних сценаріїв практичного застосування. Результати роботи можуть використовуватись у навчальних цілях, у польових вимірюваннях, у побуті чи в проєктах, що потребують автономного аналізу зображень без залучення серверної інфраструктури.

У межах реалізації було створено мобільний застосунок, який забезпечує повний цикл роботи з візуальними даними: від завантаження або зйомки фото до отримання структурованих результатів. Користувач має змогу додати зображення високої роздільної здатності у форматі HEIC, які далі обробляються локально засобами комп'ютерного зору та великих мовних моделей. Система здійснює детекцію об'єктів, визначає їх характеристики, а також формує текстові пояснення результатів у зрозумілій людині формі. Для збереження або подальшої обробки передбачена можливість експорту даних у форматі CSV.

Такий підхід не лише підвищує продуктивність та автономність, а й забезпечує зручність використання, роблячи систему гнучкою до різних практичних сценаріїв: польових досліджень, навчального аналізу, побутового застосування чи інтеграції у більші інформаційно-вимірювальні комплекси. Відсутність залежності від серверної частини дозволяє використовувати систему у середовищах з обмеженим доступом до інтернету, що робить її ефективною і безпечною для широкого кола користувачів.

1. ТЕХНІЧНЕ ЗАВДАННЯ

1.1. Постановка задачі, вимоги до функціональності, інтерфейсу, продуктивності та безпеки мобільної інформаційно-вимірювальної системи.

Підставою є завдання на дипломне проєктування. Система — iOS-додаток для розпізнавання об'єктів на зображеннях і генерації пояснень. Застосовується в мобільних інформаційно-вимірювальних системах, зокрема для аналізу візуальної інформації. Розробка: лютий – червень 2025 р.

1.2. Умови експлуатації

- Температура: 0...+45 °C
- Вологість: до 80 %
- Тиск: 600–800 мм рт. ст.
- Хімічно активні речовини — відсутні
- Режим: автономна періодична робота

1.3. Вимоги до технічних характеристик

- Вхідні дані: фото HEIC, не нижче 1080p
- Параметри: об'єкти, форма, опис
- Похибка: до 10 %
- Час обробки: до 3 с
- Результат: екран + експорт CSV
- Платформа: iOS 26+, рекомендовано iPhone 15+

1.4. Вимоги до надійності

- Стабільність: >99,5 %
- Обробка помилок та самовідновлення
- Робота без інтернету

1.5. Вимоги до оформлення

- Сучасний інтерфейс SwiftUI/UIKit

2. ОГЛЯД ТА АНАЛІЗ МЕТОДІВ ТА СИСТЕМ ОБРОБКИ ЗОБРАЖЕНЬ

2.1. Функціональність, інтерфейсу, продуктивність та безпека мобільної інформаційно-вимірювальної системи.

Виконання цієї кваліфікаційної роботи є завдання на дипломне проектування, спрямоване на розробку інноваційного мобільного засобу збору, обробки та інтерпретації візуальних даних у рамках інформаційно-вимірювальної системи. Мета розробки полягає у створенні сучасного, технологічно автономного рішення для локального розпізнавання об'єктів на зображеннях без використання зовнішніх серверів або інтернет-з'єднання [1].

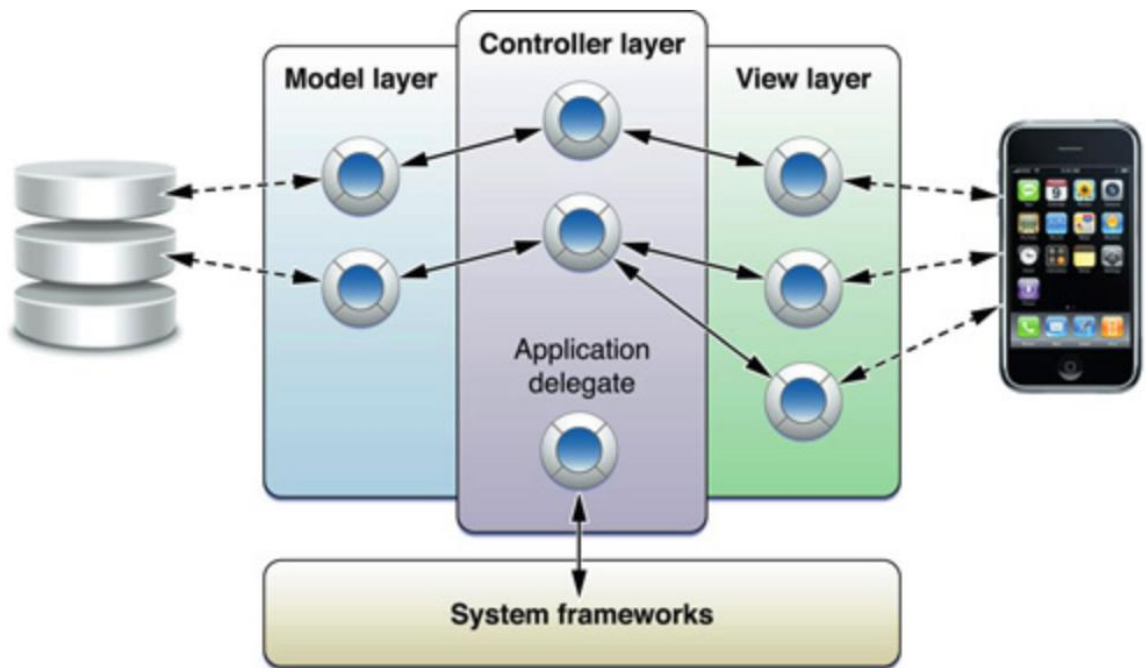


Рис. 2.1. Схіма зв'язків компонентів додатку

Проектована система реалізується у вигляді мобільного iOS-застосунку з використанням фреймворків Vision та FoundationModels. Модуль Vision відповідає за виявлення та класифікацію об'єктів, присутніх на зображенні, на основі попередньо навчених моделей. У свою чергу, FoundationModels

забезпечує генерацію природномовних пояснень до виявлених об'єктів, формуючи інтерпретаційний рівень системи. Вся обробка виконується **на пристрої користувача**, що забезпечує високий рівень приватності, швидкодії та незалежності від зовнішніх сервісів.

До основних функціональних вимог належить реалізація повного циклу роботи з даними: завантаження зображень, їх обробка, збереження результатів та перегляд детальних пояснень. Також передбачено можливість експорту проаналізованої інформації у **форматі CSV**, що дозволяє легко інтегрувати результати в інші системи або використовувати їх у наукових чи технічних звітах.

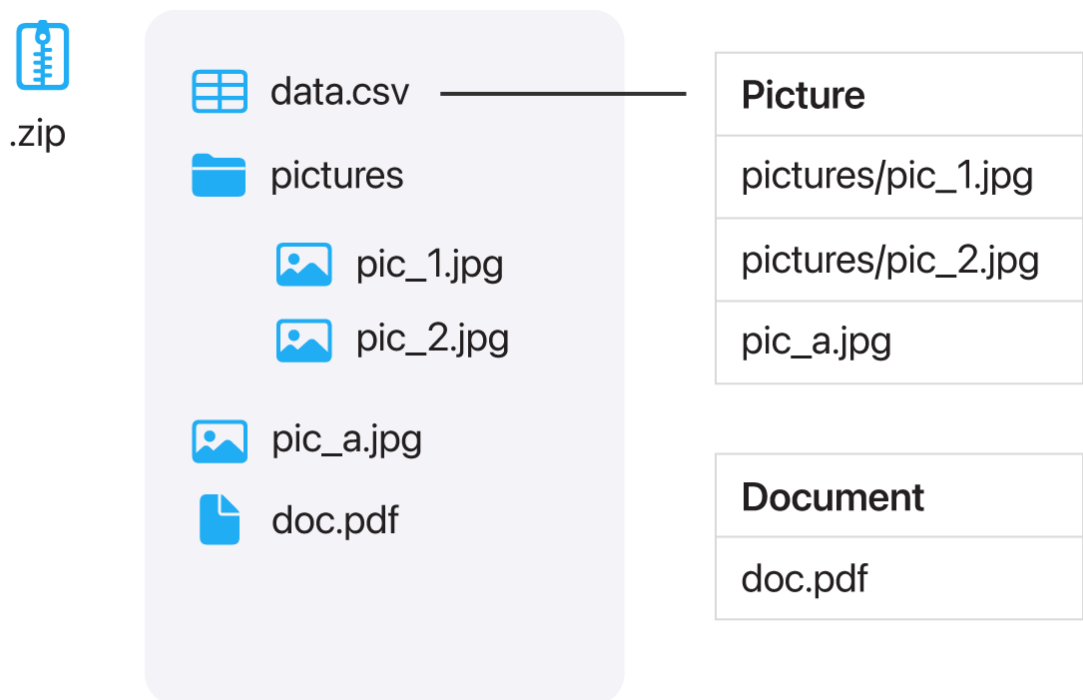
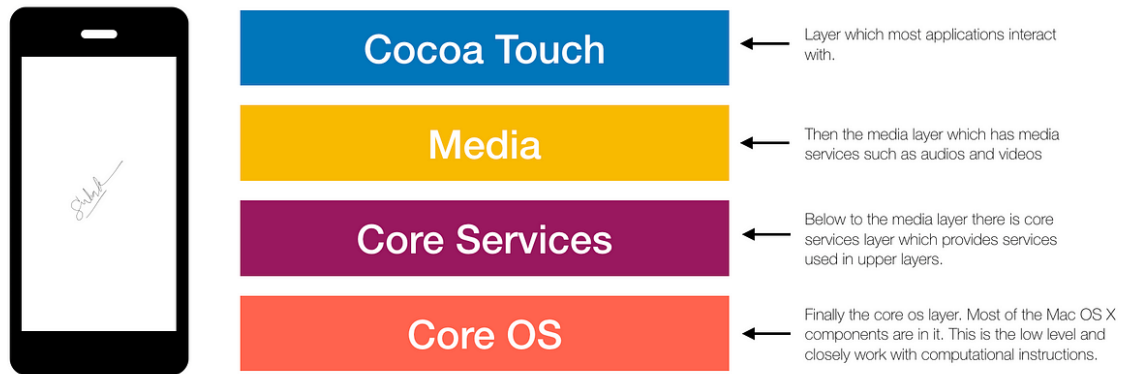


Рис. 2.2. CSV приклад використання

Інтерфейс застосунку повинен бути інтуїтивно зрозумілим, підтримувати роботу з кількома зображеннями, забезпечувати швидкий доступ до результатів обробки та дозволяти користувачеві отримувати опис

об'єктів за допомогою зручної взаємодії (таби, списки, popover тощо). Важливою є також реалізація навігаційної логіки, що базується на принципах UX-дизайну для мобільних платформ, з акцентом на простоту та ефективність.



All Rights Reserved To Anuradh Caldera

Рис. 2.3. Схема iOS input леєр

До вимог продуктивності мобільної інформаційно-вимірювальної системи висуваються підвищені критерії, зумовлені необхідністю оперативної обробки візуальних даних безпосередньо на пристрої користувача. Насамперед, система повинна забезпечувати коректне розпізнавання об'єктів на зображеннях з роздільною здатністю не нижче **1920×1080 пікселів (1080p)** із затримкою, що не перевищує **2–3 секунд** з моменту вибору або фіксації зображення до отримання результатів аналізу. Це дозволяє підтримувати взаємодію в реальному часі та гарантує комфортну роботу користувача без тривалого очікування.

Система повинна функціонувати стабільно на пристроях середнього та вище середнього рівня (зокрема iPhone 12, iPhone SE 3-го покоління, iPad з чіпами A14 і новішими), при цьому не допускаючи суттєвих навантажень на **CPU** та **GPU**, які можуть спричинити перегрів, зниження швидкодії інших

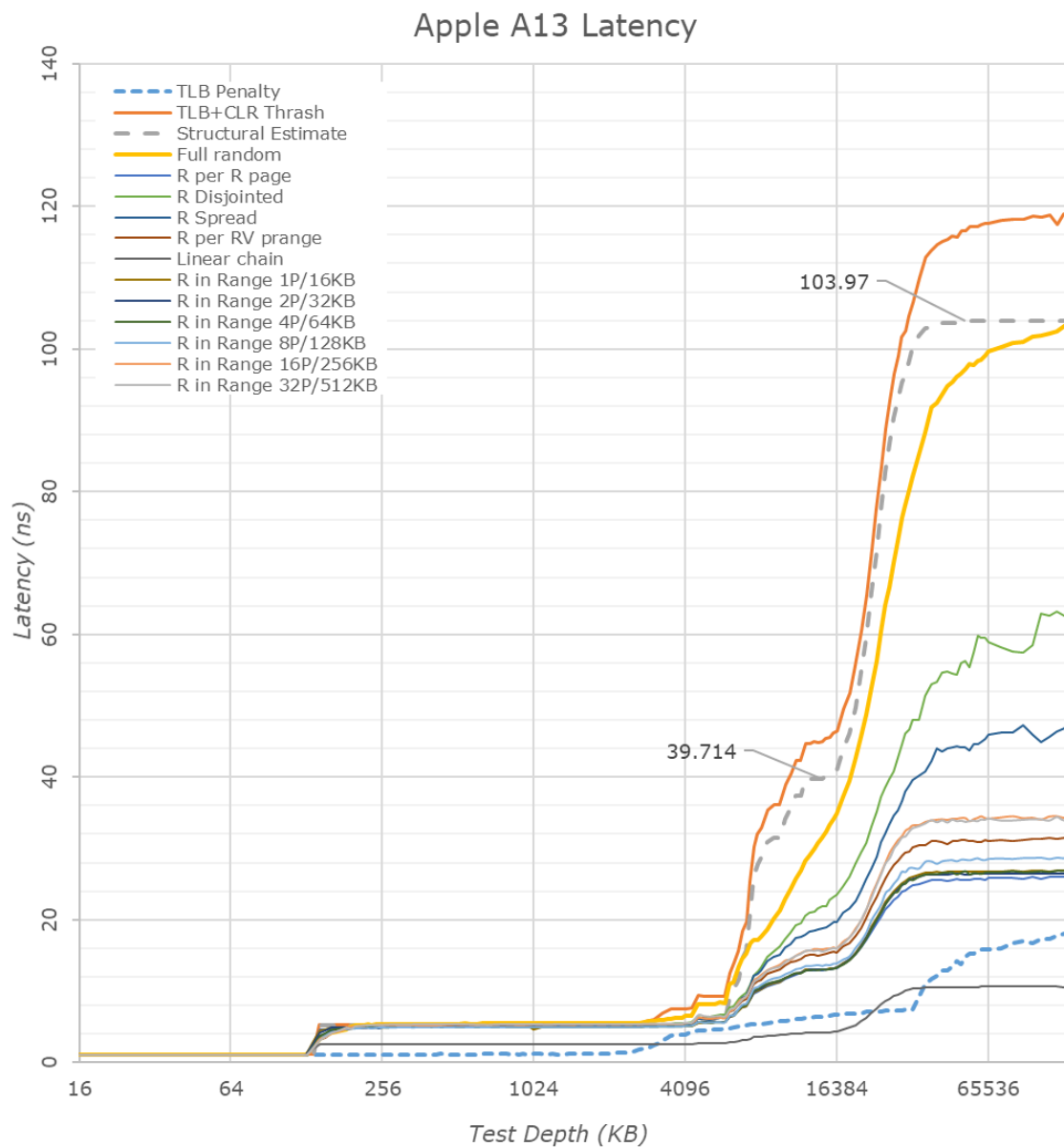
застосунків або швидкий розряд акумулятора. У зв'язку з цим реалізація модулів обробки зображень та генерації тексту повинна враховувати особливості мобільної апаратної платформи, використовуючи апаратне прискорення (Metal, Neural Engine) лише там, де це виправдано, та забезпечуючи баланс між швидкістю, якістю та енергоспоживанням.

Крім того, важливими є **оптимізація пам'яті (RAM)**, що передбачає мінімізацію тимчасових буферів, уникнення дублювання даних та ефективне управління життєвим циклом зображень, проміжних результатів та виводу. Особливо це актуально в контексті **обробки кількох зображень у черзі**, коли користувач може додати одразу серію фото, і система повинна гарантувати стабільність без аварійного завершення чи зниження швидкодії.

Також до продуктивності належить **швидкість запуску застосунку**, що має не перевищувати 1 секунди при холодному старті, та **миттєва реакція інтерфейсу** на дії користувача. Анімації, перехід між екранами, запуск обробки та відображення результатів повинні бути узгоджені з очікуваннями користувача, не викликати зависань чи візуальних лагів.[2]

Загалом система повинна забезпечувати **стабільну, безпечну та передбачувану роботу в мобільних умовах**, враховуючи особливості мобільної платформи iOS та обмеженість ресурсів пристрою. Вона має підтримувати **послідовну обробку серії зображень** — не менше 10–15 знімків протягом однієї сесії без необхідності ручного очищення пам'яті чи перезапуску. Це означає, що механізми управління кешем, пам'яттю та внутрішнім станом мають бути оптимізовані для безперебійної роботи в реальних умовах використання.

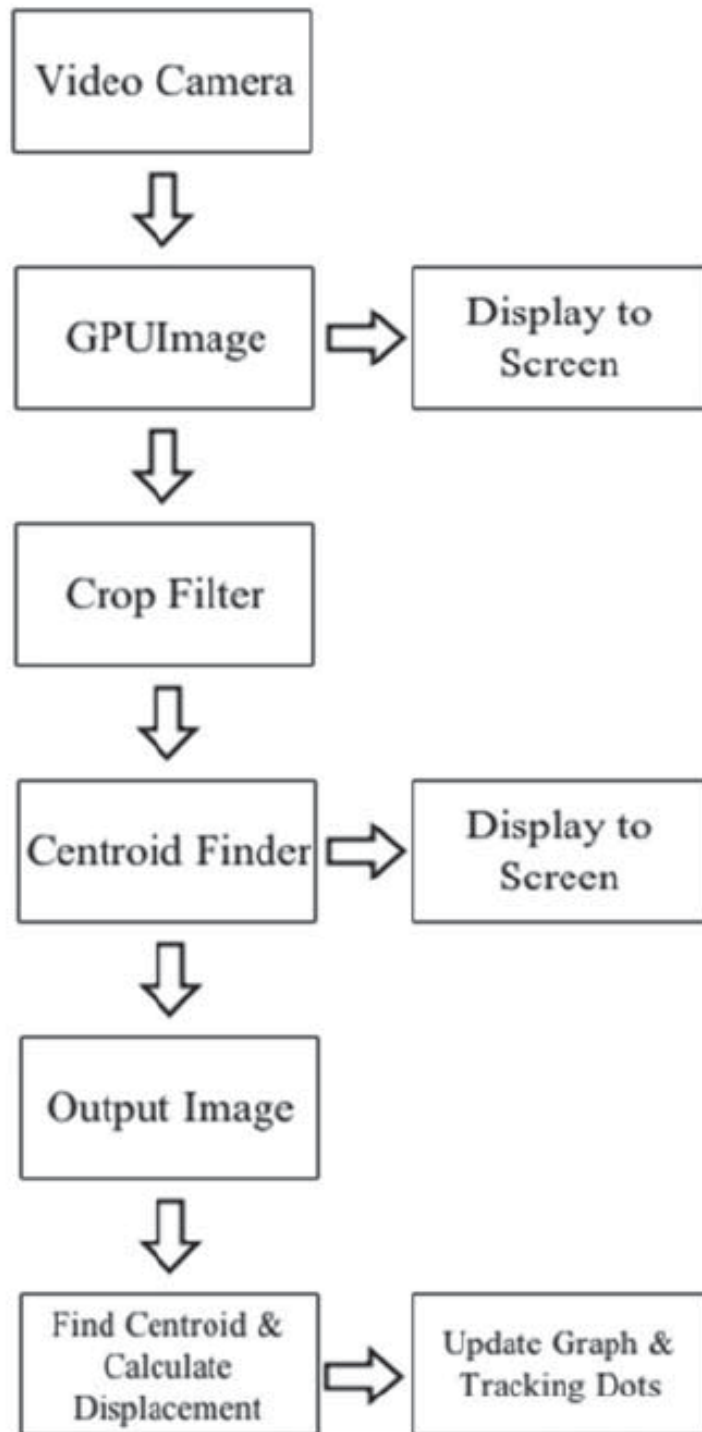
Крім того, застосунок повинен коректно функціонувати у **фоновому режимі**, в межах дозволеного iOS-таймера (до ~3 хвилин), з можливістю завершення активної обробки або збереження проміжного стану. У разі примусового згортання або перезавантаження системи додаток має **відновити попередній сеанс роботи**, включно з обробленими зображеннями, згенерованими описами та ще неекспортованими даними.



2.4. Діаграма затримки найпопулярнішого процесору Apple

З точки зору безпеки, додаток повинен гарантувати локальне зберігання даних, виключення несанкціонованого доступу до зображень або результатів аналізу, а також відсутність відправки персональної інформації через мережу. Вся логіка реалізована з урахуванням рекомендацій Apple щодо конфіденційності та обробки персональних даних.

Таким чином, завданням проекту є створення повноцінної автономної вимірювальної системи нового покоління, яка поєднає потужність мобільного машинного зору, природномовного інтерфейсу та сучасного архітектурного підходу до проектування додатків. Початок розробки — лютий 2025 року. Завершення — червень 2025 року.



2.5. Блочна діаграма додатку

2.2. Застосування комп'ютерного зору у мобільних вимірювальних системах

У мобільних інформаційно-вимірювальних системах комп'ютерний зір дозволяє автоматично ідентифікувати, класифікувати та вимірювати об'єкти без використання додаткового обладнання, базуючись виключно на зображеннях, отриманих з камери пристрою. Такий підхід істотно зменшує вартість та розміри системи, а також забезпечує зручність її використання в польових умовах.

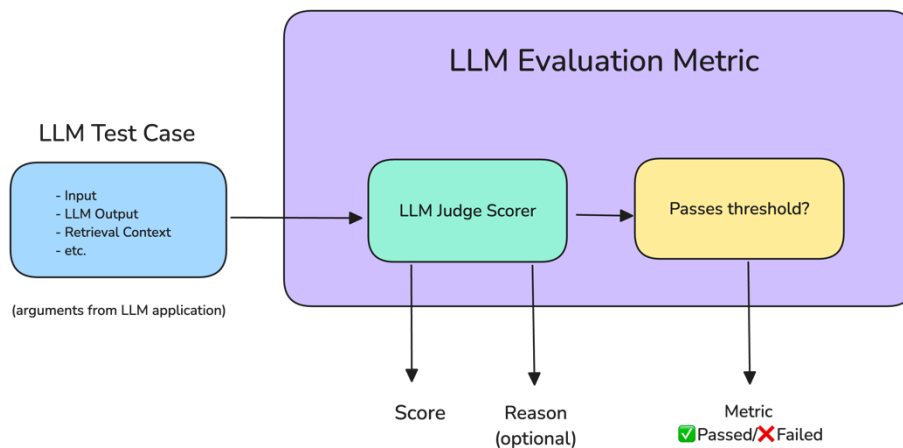


Рис. 2.6. Основні метрики LLM

Мобільні додатки, що використовують комп'ютерний зір, знаходять широке застосування в інженерних, медичних, освітніх та наукових задачах, включаючи аналіз матеріалів, визначення дефектів, вимірювання геометричних параметрів та побудову контекстних пояснень. У межах даної роботи комп'ютерний зір реалізується як складова вимірювальної системи, яка дозволяє отримувати параметри об'єктів на зображеннях для подальшого аналізу та інтерпретації з використанням моделей штучного інтелекту.

2.3. Технології розпізнавання об'єктів на основі машинного навчання

Розпізнавання об'єктів є одним із найважливіших застосувань машинного навчання в комп'ютерному зорі. Метою цього процесу є виявлення, класифікація та локалізація об'єктів на зображенні, що дозволяє системі здійснювати подальший аналіз сцен або вимірювання параметрів виявлених об'єктів.

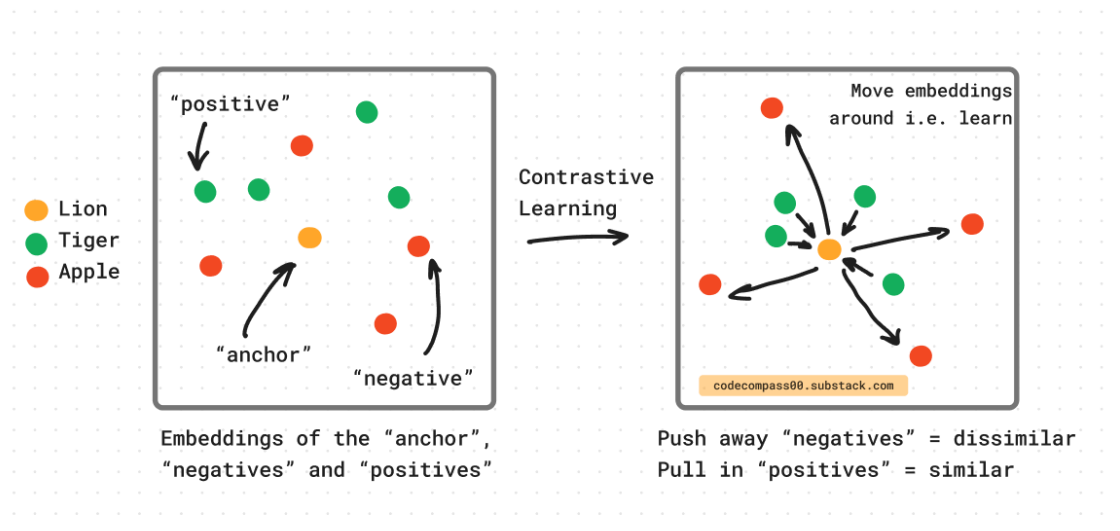


Рис. 2.7. Схема розпізнавання об'єктів на фото

Застосування машинного навчання, зокрема глибоких нейронних мереж (Deep Neural Networks, DNN), дозволяє досягти високої точності розпізнавання завдяки здатності моделей автоматично виділяти суттєві ознаки з великої кількості навчальних даних. Однією з найпоширеніших архітектур у задачах розпізнавання об'єктів є згорткові нейронні мережі (Convolutional Neural Networks, CNN), які успішно застосовуються в мобільних пристроях завдяки своїй ефективності та здатності до оптимізації

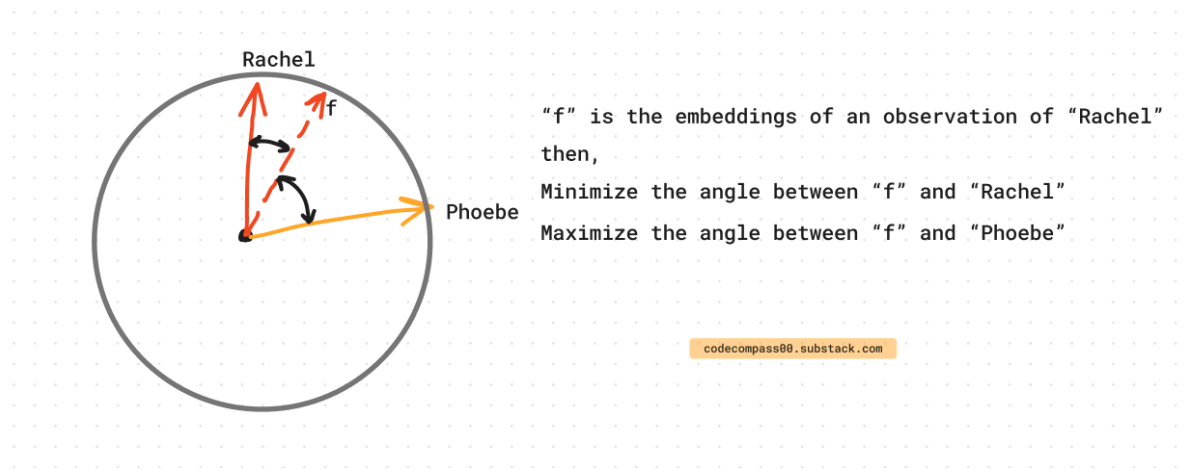


Рис. 2.8. Embedding models

Також варто зазначити тенденцію до використання трансформерних моделей (наприклад, CLIP, OWL-ViT), які поєднують візуальну та текстову інформацію, дозволяючи не лише виявити об'єкт, але й сформуванати його текстовий опис. У контексті даної роботи такі моделі реалізуються за допомогою бібліотеки FoundationModels, яка забезпечує генерацію пояснень на основі вхідних параметрів.

Класифікація моделей для розпізнавання об'єктів - це поділ алгоритмів комп'ютерного зору на категорії за принципом їхньої архітектури, підходу до обробки зображень і типу результату, який вони генерують. Основна мета — виділити об'єкти на зображенні, визначити їх положення (bounding box) і класи (label). У мобільному середовищі важливим є також баланс між якістю розпізнавання та часом реакції інтерфейсу. Тому вибір моделей повинен враховувати не лише точність, але й час інференсу, використання пам'яті та стабільність в умовах багатозадачності. У цьому контексті поєднання оптимізованих CNN-моделей для первинного виявлення з легкими LLM-модулями для пояснення формує основу гнучкої, надійної та ефективної інформаційно-вимірювальної системи нового покоління.

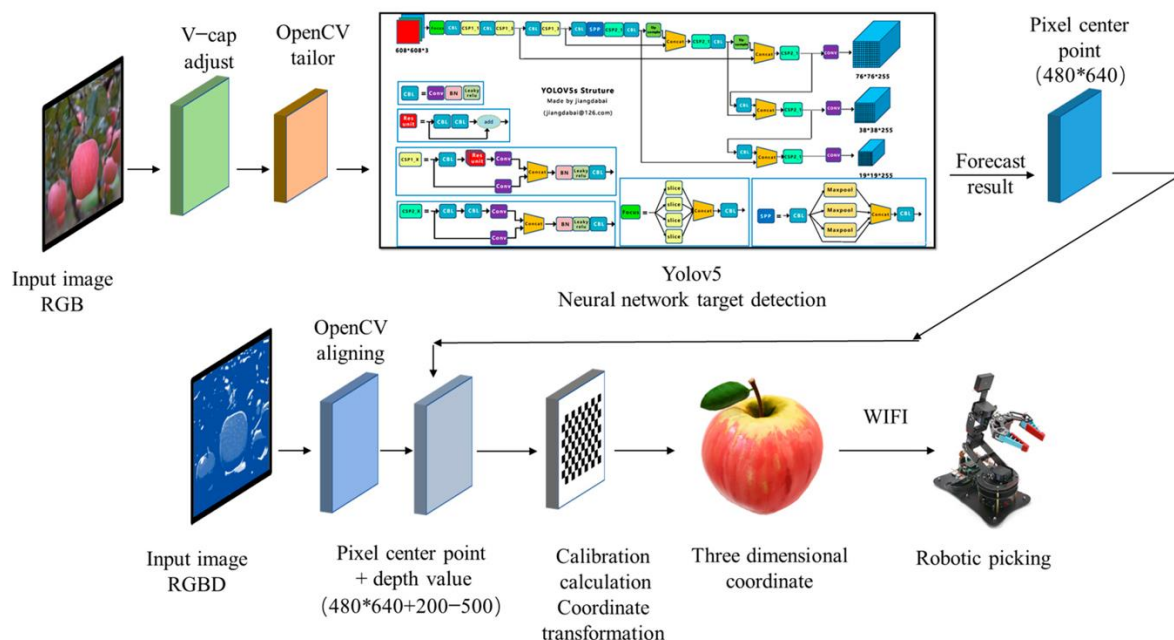


Рис. 2.9. Схема прикладу класифікації

Моделі для розпізнавання об'єктів поділяються на кілька основних типів залежно від архітектури, принципу роботи та призначення. Найбільш поширеним підходом є використання згорткових нейронних мереж (CNN), які обробляють вхідні зображення шляхом послідовного застосування фільтрів, що дозволяє виявляти локальні ознаки об'єктів — краї, текстури, контури та інші характеристики.

У загальному випадку моделі для розпізнавання об'єктів поділяють на два великі класи: **моделі класифікації зображень** (image classification) та **моделі детекції об'єктів** (object detection). Перші визначають, що зображено на всій сцені, без локалізації конкретних елементів. Прикладами таких архітектур є ResNet, MobileNet, EfficientNet. Натомість моделі детекції не лише класифікують, але й знаходять положення об'єктів на зображенні у вигляді обмежувальних прямокутників (bounding boxes). Серед них найвідомішими є YOLO (You Only Look Once), SSD (Single Shot Detector) та Faster R-CNN [3].

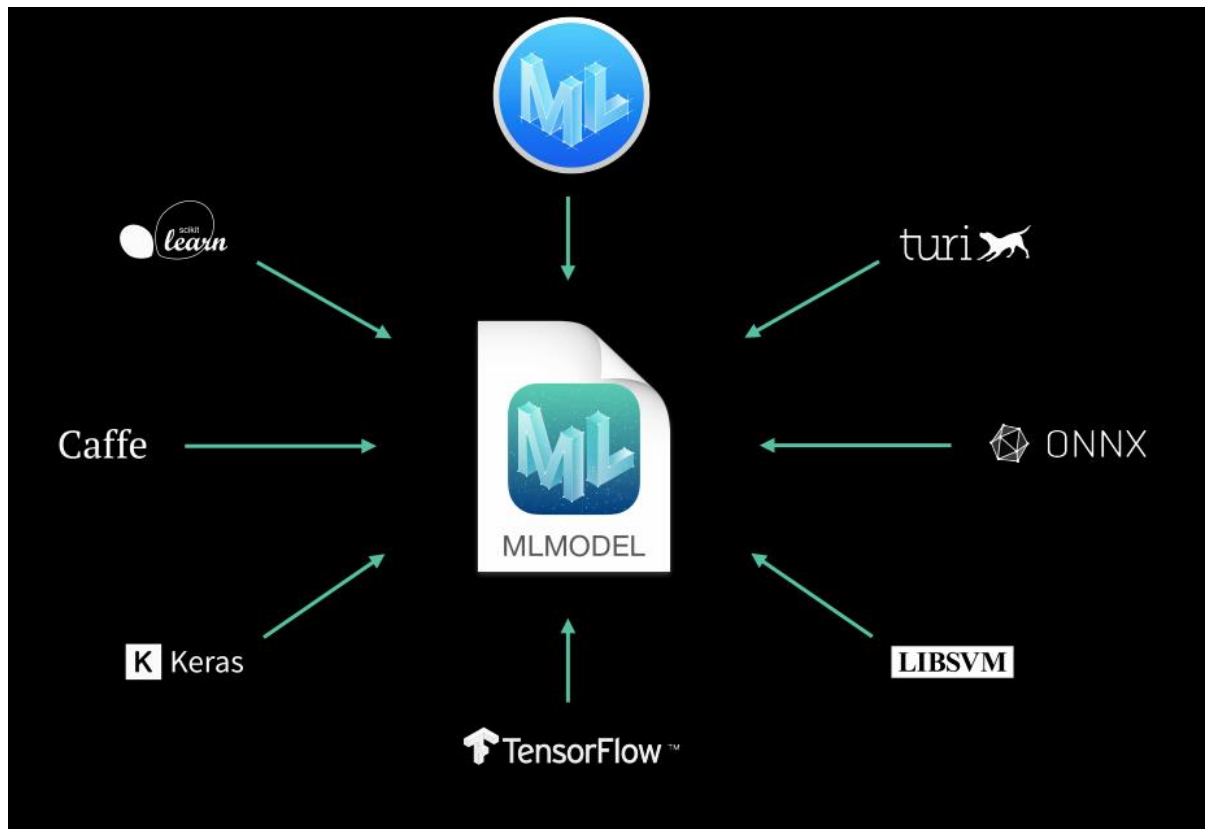
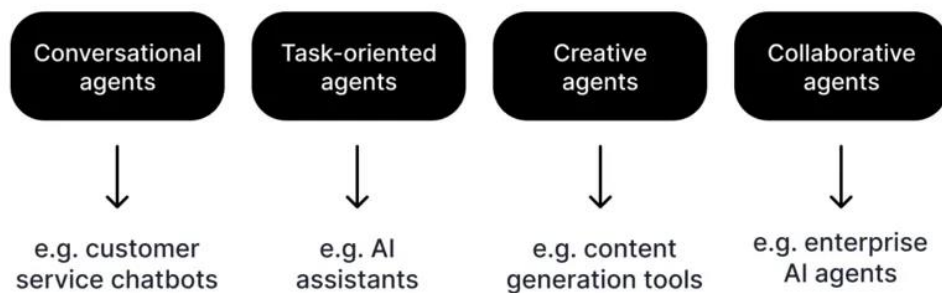


Рис. 2.10. Схема файлу моделі

Залежно від потреб застосунку обирається баланс між точністю, швидкістю та вимогами до ресурсів. Для мобільних пристроїв найчастіше використовуються легковагові модифікації моделей, наприклад Tiny YOLO аб MobileNet-SSD, які дозволяють проводити обробку без значного

Types of LLM Agents



навантаження на систему.

2.4. Висновки до розділу

У цьому розділі було здійснено всебічний аналіз сучасних методів обробки зображень, що лежать в основі реалізації комп'ютерного зору в мобільних інформаційно-вимірювальних системах. Особливу увагу приділено не лише технічній реалізації процесу виявлення об'єктів, а й семантичному рівню аналізу, що має критичне значення для вимірювання інформації, її класифікації та подальшої інтерпретації. Такий підхід виходить за межі класичного "вимірювання фізичних величин", розширюючи парадигму на семантичні та контекстні ознаки, які фіксуються та аналізуються через візуальний канал.

Технології машинного навчання, зокрема згорткові нейронні мережі (CNN), одноетапні детектори (YOLO, SSD) та трансформерні архітектури, розглядаються як ключові інструменти, що дозволяють забезпечити точне виявлення об'єктів, а також вимірювання ступеня впевненості класифікації. Це надає змогу інтерпретувати зображення як джерело багатовимірної інформації, де кожна ознака (label) має свій рівень значущості — у вигляді числової оцінки довіри (confidence score), що, по суті, є вимірюваною величиною в рамках даної системи.

Використання бібліотеки **Vision** дозволяє ефективно реалізовувати перший етап — детекцію та попередню класифікацію об'єктів, тоді як інтеграція **FoundationModels** відкриває можливість досягнення глибшого рівня інтерпретації даних через генерацію природномовних пояснень. Це означає, що система не лише "вимірює", а й "осмислює" візуальну інформацію, що робить її значно ближчою до інтелектуальних вимірювальних платформ нового покоління.

Таким чином, застосування комп'ютерного зору в межах мобільного iOS-застосунку перетворює зображення на повноцінне джерело інформаційно-вимірювальних даних. Це включає не лише детекцію об'єктів, а

й аналіз їх типів, взаєморозташування, контексту та семантичного змісту. Такий підхід є принципово новим кроком у розвитку ІВТ, де "вимірювання" набуває не лише кількісного, а й якісного — змістовного характеру.

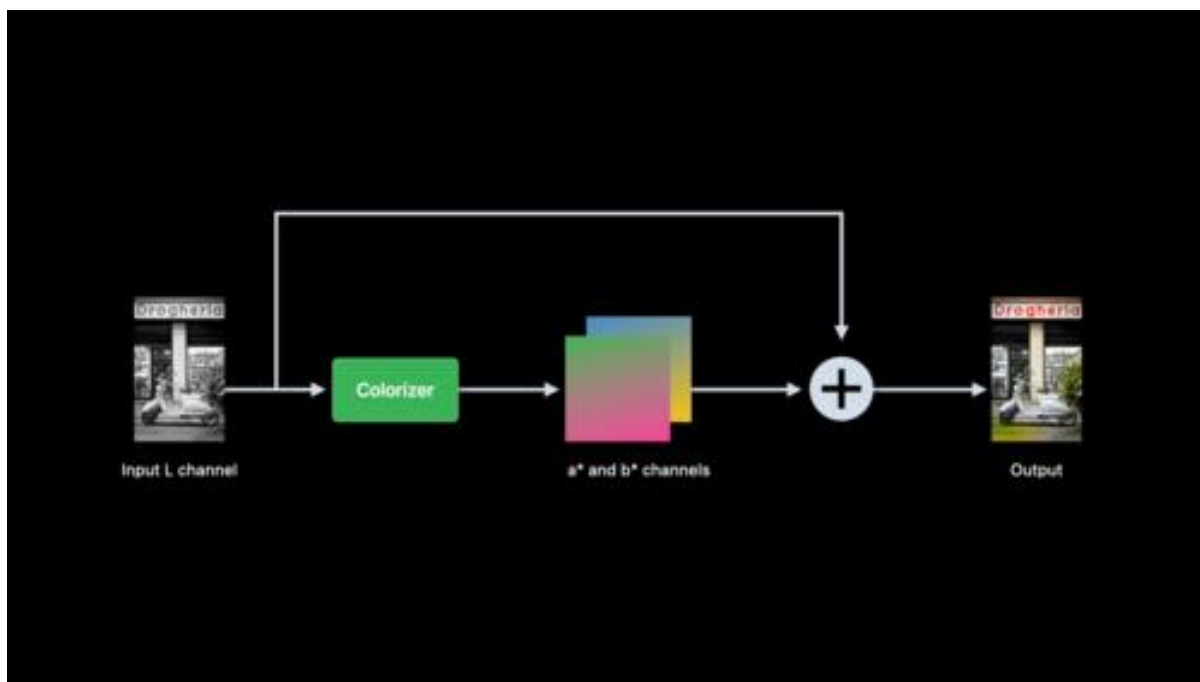


Рис. 2.12. Схема підходу input output

Також важливою перевагою запропонованого рішення є можливість масштабування. Завдяки модульній архітектурі та використанню сучасних фреймворків, таких як Vision та FoundationModels, система може бути адаптована для вирішення нових вимірювальних задач, включаючи визначення геометричних параметрів, контроль якості візуальних об'єктів або автоматизоване сортування за вмістом. Розширення функціоналу можливе без зміни фундаментальної структури додатку, що забезпечує його довготривалу актуальність і зручність підтримки.

Крім того, реалізація локальної обробки даних гарантує відповідність сучасним стандартам конфіденційності та кібербезпеки, особливо актуальним в умовах зростаючих вимог до захисту персональної інформації. Відсутність потреби передавати дані на сервери або в хмару мінімізує ризики витоку або несанкціонованого доступу, що є критично важливим для застосунків, які можуть використовуватись у медичних, освітніх або правових сферах.

3. РОЗРОБКА СТРУКТУРНОЇ СХЕМИ

3.1. Структурна схема інформаційно-вимірювальної системи

Структурна схема, зображена на діаграмі, відображає багаторівневу архітектуру мобільного застосунку для інформаційно-вимірювальної системи, що виконує розпізнавання об'єктів на зображеннях. Вона поділена на **три логічні рівні**

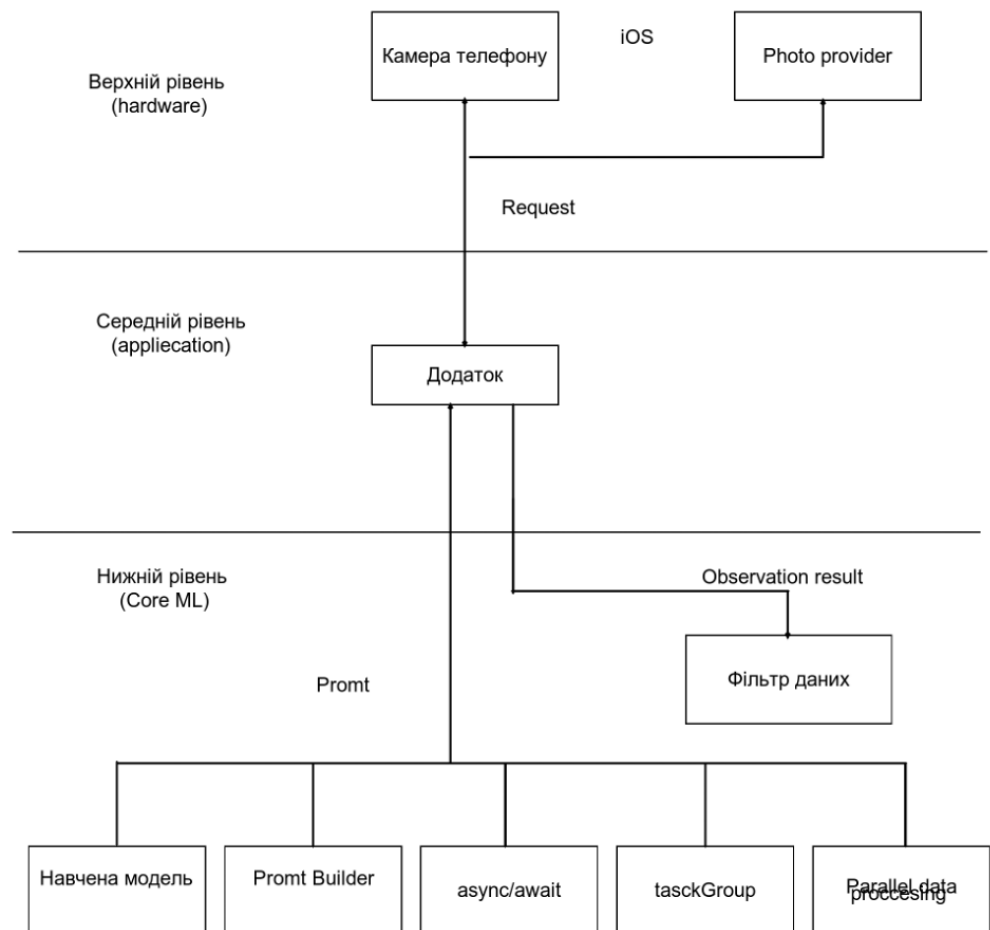


Рис. 3.1. Структурна схема

Верхній рівень — Апаратний рівень (Hardware)

Цей рівень відповідає за отримання вхідних даних:

- **Камера телефону** — модуль, який робить фотографії в реальному часі.
- **Photo provider** — доступ до галереї/фотоплівки користувача.
- Взаємодія між цими компонентами та додатком здійснюється через **Request** — запит на отримання зображення.

Середній рівень — Прикладний рівень (Application)

Цей рівень виконує основну логіку обробки:

- **Додаток** — центральний компонент, що обробляє запити користувача, ініціює запуск моделей, виконує навігацію та управляє інтерфейсом.
- Він отримує дані із верхнього рівня, передає їх на нижній (Core ML) рівень, а також отримує **Observation result** — результати обробки зображення.
- **Фільтр даних** — виконує попередню фільтрацію об'єктів, наприклад, за порогом довіри або типом розпізнаного об'єкта.

Нижній рівень — Рівень машинного навчання (Core ML)

Цей рівень відповідає за інтелектуальну обробку зображень:

- **Навчена модель** — модель машинного навчання, інтегрована через Core ML (наприклад, Vision або CLIP).
- **Prompt Builder** — компонент, який формує запит (prompt) для мовної моделі на основі виявлених об'єктів.
- **async/await** — використовується для асинхронної обробки зображень без блокування основного потоку.
- **taskGroup** — дозволяє запускати кілька паралельних задач (наприклад, розпізнавання кількох об'єктів).
- **Parallel data processing** — модуль для паралельної обробки даних, що пришвидшує роботу при великій кількості об'єктів.

Усі ці блоки передають результати назад у **додаток**, формуючи повний цикл: від зчитування фото — до пояснення, фільтрації та відображення користувачеві.

Інформаційно-вимірювальна система, що стала об'єктом розробки у межах цієї кваліфікаційної роботи, реалізована у вигляді повнофункціонального автономного **iOS-додатку**, який забезпечує вхід,

обробку, інтерпретацію та візуалізацію візуальної інформації. Архітектура системи ґрунтується на модульному підході з чітким розмежуванням обов'язків, що дозволяє досягти високої масштабованості, повторного використання коду та надійності під час виконання обчислень без підключення до інтернету. Вся логіка роботи додатку базується на обробці зображень та вимірюванні інформації, отриманої із візуальних джерел.

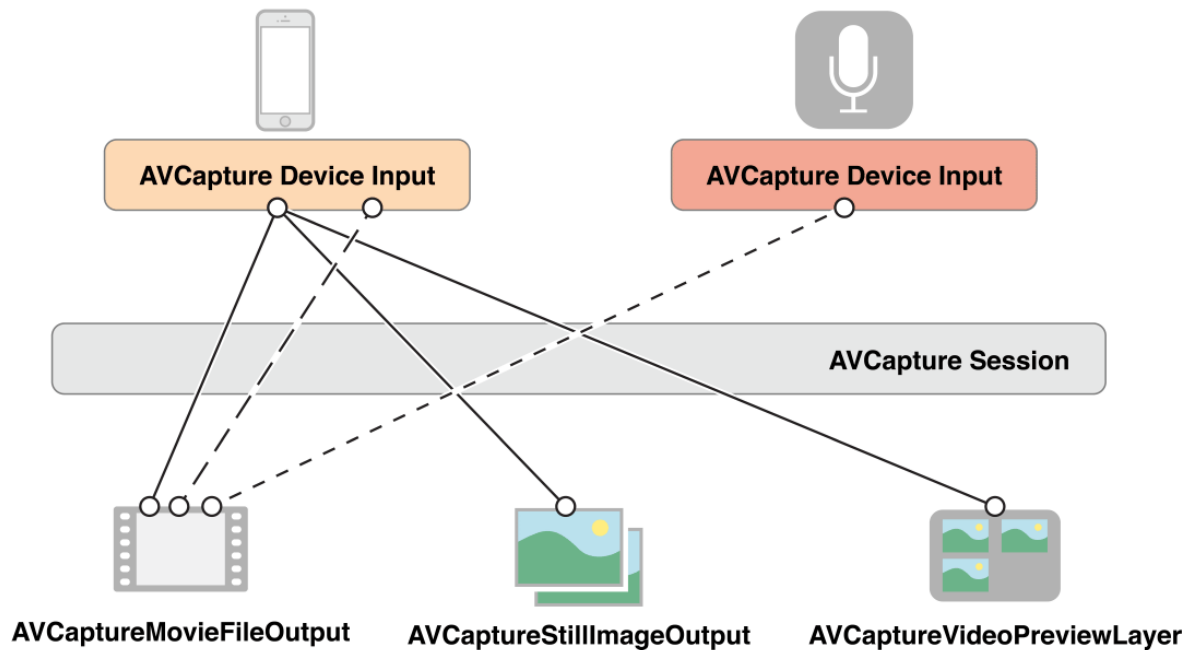


Рис. 3.2. Схема захвату медіа

Початковим етапом у функціонуванні системи є отримання вхідного зображення — користувач може зробити знімок камерою або вибрати готове фото з галереї пристрою. Цей етап реалізовано у вигляді окремого модуля захоплення зображення, який також відповідає за валідацію формату, роздільної здатності та перевірку якості зображення для подальшого коректного аналізу.

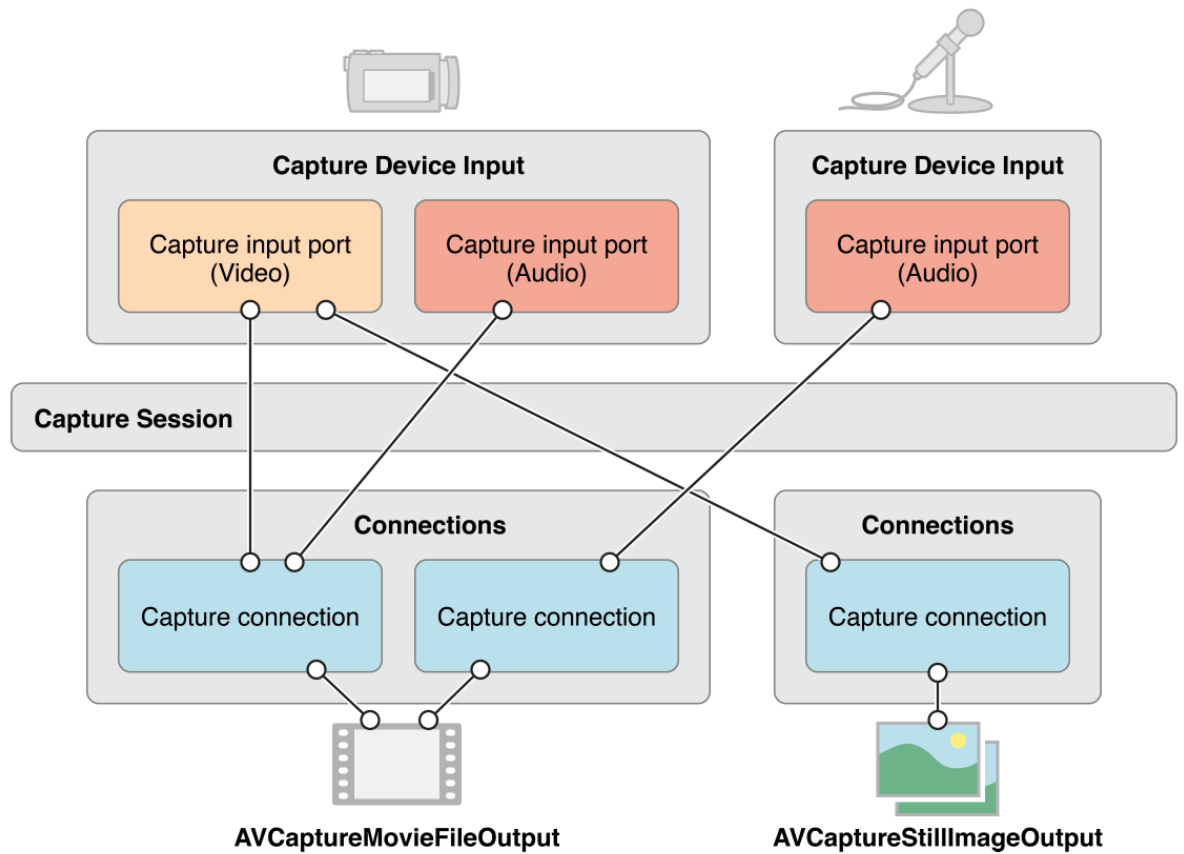


Рис. 3.3. Схема сесії захвату медіа

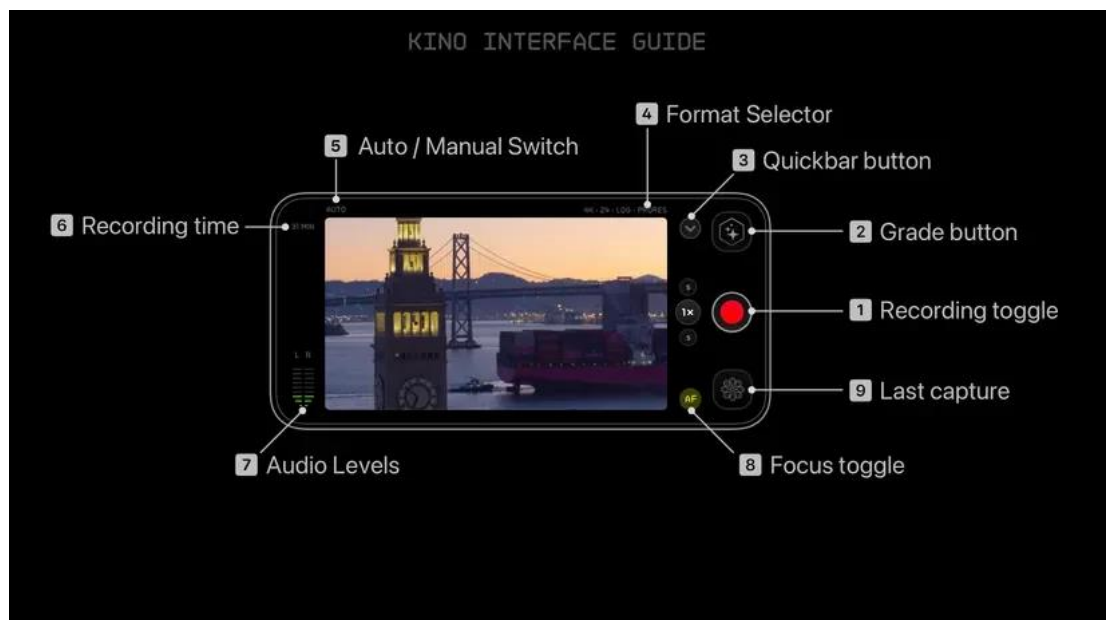


Рис. 3.4. Юзер леєр сісії захвату

Після цього зображення передається до модуля комп'ютерного зору, який реалізовано з використанням фреймворку **Vision** від Apple. Цей модуль

відповідає за **виявлення об'єктів** на зображенні, їх класифікацію та обчислення супровідних параметрів: координати області виявлення (bounding box), орієнтація, розмір, ідентифікатор об'єкта, клас за ієрархією предметів, а також числовий показник упевненості моделі в розпізнаванні (confidence score). У певних випадках може також застосовуватись попереднє семантичне сегментування для виділення контексту зображення.

Результати, отримані від Vision, формують **інформаційно-вимірювальний зріз**, який подається на вхід наступного модуля — **модуля інтерпретації**, реалізованого на базі **FoundationModels**, де функціонує локальна **велика мовна модель (LLM)**.

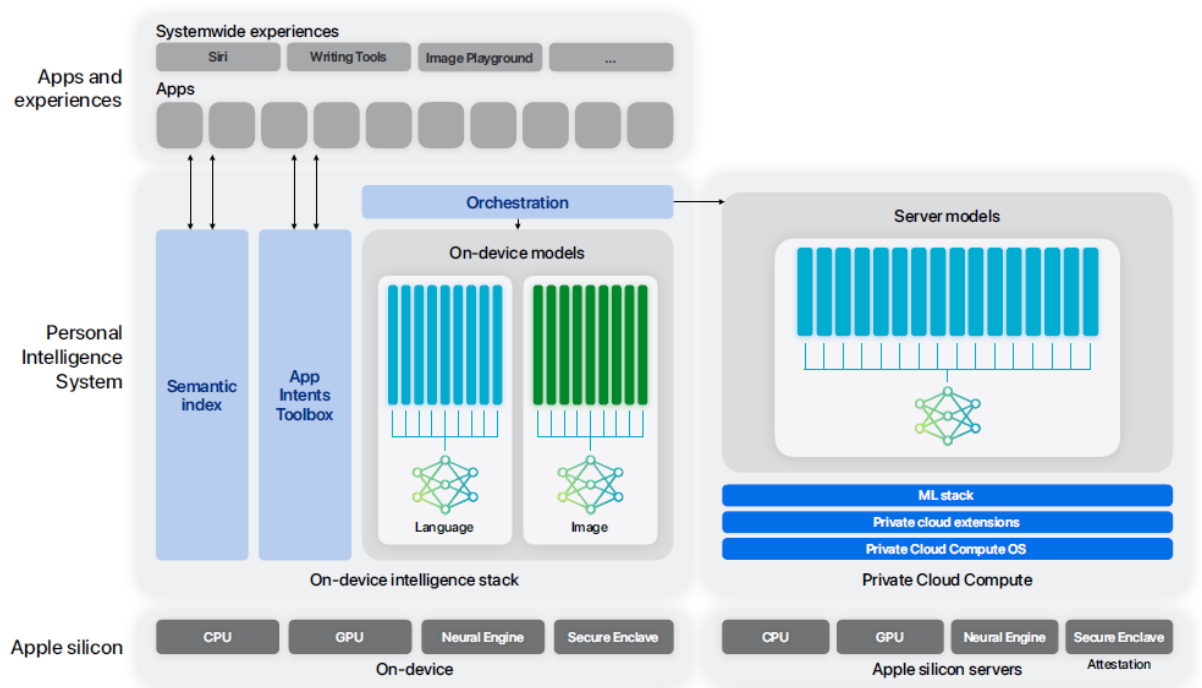


Рис. 3.5. Схема побудови FoundationModels

Цей етап є ключовим у системі, адже забезпечує **семантичне вимірювання** — перетворення числових результатів на пояснення, опис або навіть короткий висновок щодо того, що зображено на фото, у зрозумілому людині вигляді. Завдяки цьому етапу система перестає бути лише детектором — вона стає справжнім інтерпретатором візуального середовища. У структурі

реалізовано механізм передачі параметрів у форматі JSON, що забезпечує чіткий і стабільний канал обміну даними між модулями.



Рис. 3.5. Схема ядра Vision фреймворку

Згенероване пояснення, разом із параметрами аналізу, надходить до **модуля представлення даних**, який формує зведений звіт у зручному вигляді. Користувач бачить на екрані ключові об'єкти, їхні характеристики та текстову інтерпретацію. У межах цього ж модуля реалізовано **експорт результатів у форматі CSV**, що дозволяє зберігати аналіз у вигляді табличної інформації для подальшої обробки, архівації або перенесення в інші системи. Додатково, модуль представлення результатів взаємодіє з системою зберігання стану, що дозволяє підтримувати історію попередніх аналізів у межах сесії. Це дає змогу користувачу переглядати не лише поточне зображення, а й повертатися до попередніх результатів без необхідності повторної обробки. Такий підхід дозволяє ефективно проводити порівняльний аналіз або формувати серії спостережень на основі кількох зображень.

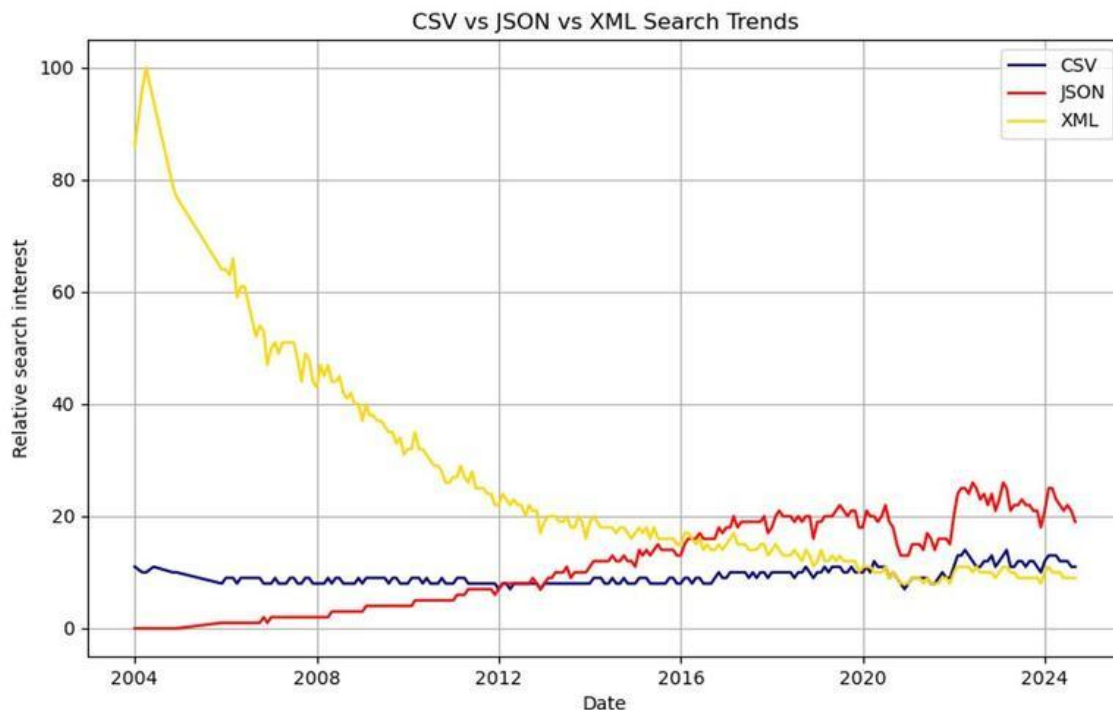


Рис. 3.6. Діаграма порівнянь швидкості CSV формату

У контексті розробки мобільного iOS-додатку для локального розпізнавання об'єктів особливу роль відіграє **архітектура керування станом та логікою додатку**, оскільки вона безпосередньо впливає на підтримуваність коду, простоту масштабування, тестованість та стабільність у складних сценаріях взаємодії. Саме тому у даній роботі було обрано **архітектурний підхід TCA (The Composable Architecture)** - сучасну декларативну модель побудови застосунків, що заснована на принципах функціонального програмування та композиційності.

TCA пропонує чітке розділення відповідальностей у межах додатку на три основні складові:

- **State (стан)** - структура, що описує усі дані, необхідні для представлення інтерфейсу на будь-якому етапі роботи.
- **Action (дії)** - перелік можливих подій, що можуть змінювати стан або ініціювати побічні ефекти (наприклад, аналіз зображення, перехід між екранами тощо).

- **Reducer (редуктор)** - чиста функція, що описує, як на основі поточного стану та отриманої дії формується новий стан або викликаються side-effects.

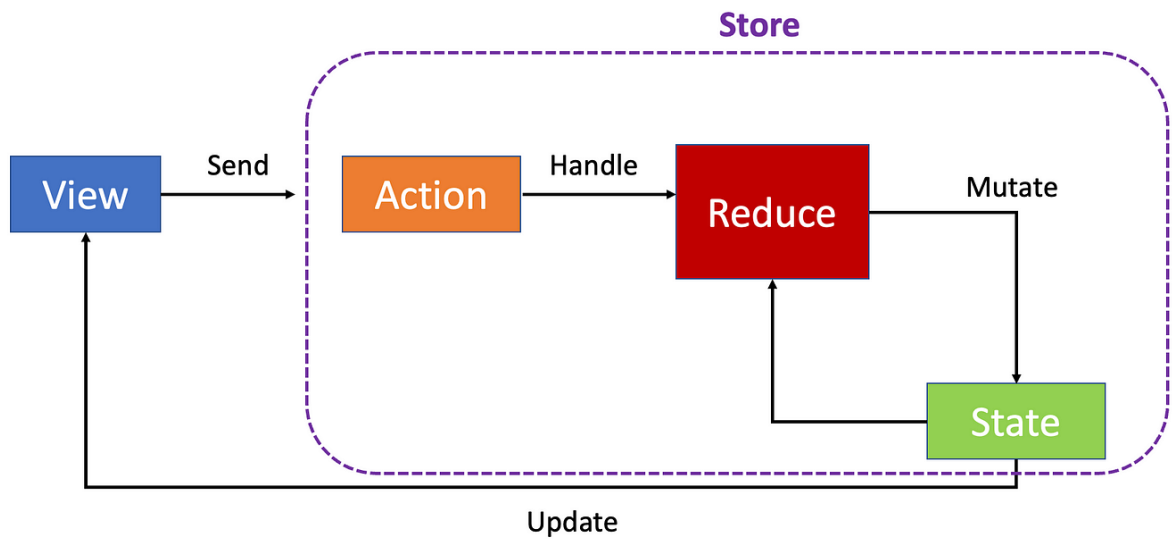


Рис. 3.7. Діаграма архітектури

Цей підхід дозволяє побудувати архітектуру, в якій **усі зміни стану є передбачуваними та контрольованими**, а також можуть бути протестовані в ізоляції — без потреби запускати UI чи емулювати пристрій.

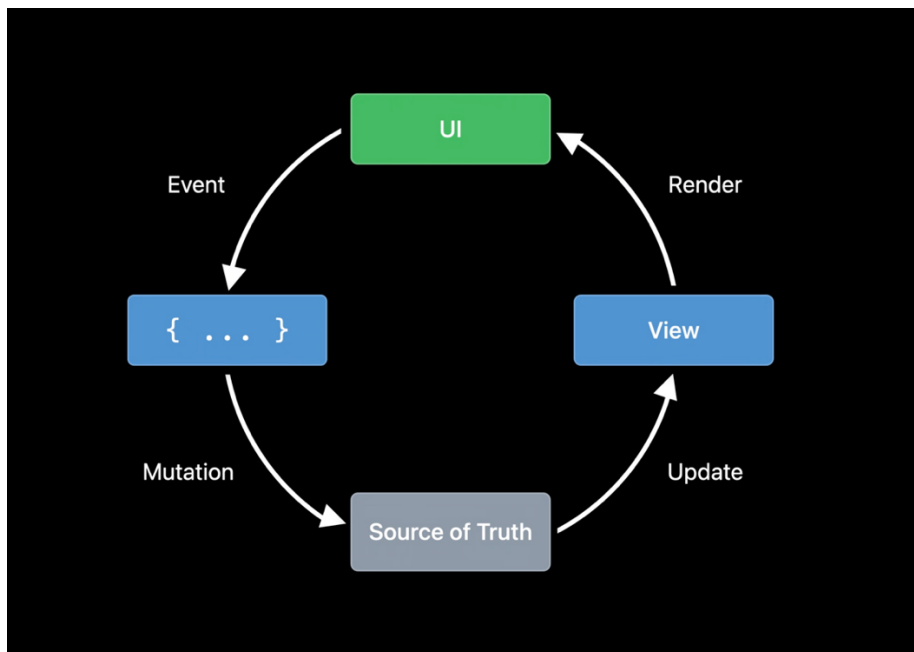


Рис. 3.8. Схема життєвого циклу обробки

У нашій системі кожен екран (наприклад, головний екран із завантаженням фото, екран перегляду результатів, модуль експорту) представлений як окремий **фічевий модуль**, який містить власний State, Action, Reducer, View та, при потребі, Effect. Такі модулі можна легко вкладати один в одного, будуючи ієрархію функціональності без дублювання коду.

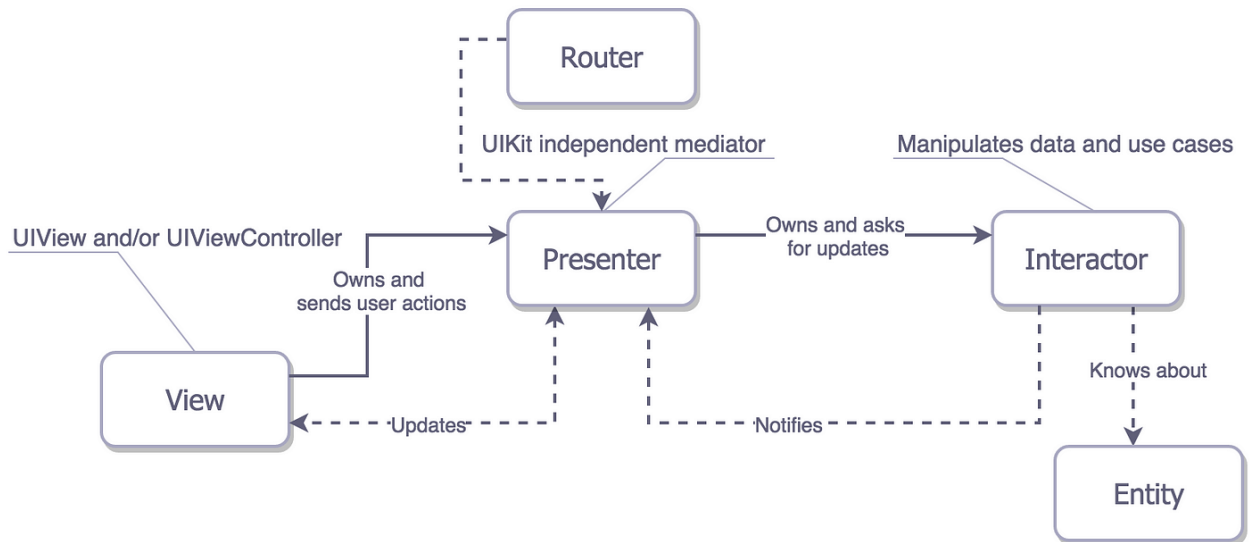


Рис. 3.9. Схема роботи в рамках UIKit реалізації

Особливістю розробленої інформаційно-вимірювальної системи є її повна автономність, що означає повне виконання всіх етапів обробки зображення без залучення зовнішніх серверів, API чи хмарних обчислювальних ресурсів. Уся логіка аналізу — від отримання фото до виводу інтерпретованого результату — реалізована виключно локально, на пристрої користувача. Це дозволяє досягти максимальної незалежності від підключення до мережі, а також зменшує ризики, пов'язані з передачею персональних або чутливих даних.

Такий підхід забезпечує не лише стабільну роботу системи в умовах відсутності інтернету, а й підвищує загальну надійність, безпеку та зручність експлуатації. Зокрема, це є надзвичайно важливим у випадках польових вимірювань, мобільних лабораторій або ситуацій, коли додаток використовується в місцях без стабільного зв'язку. Завдяки автономності,

система здатна працювати з високою стабільністю та передбачуваністю у найрізноманітніших сценаріях використання — від освітніх до науково-дослідницьких або побутових [4].

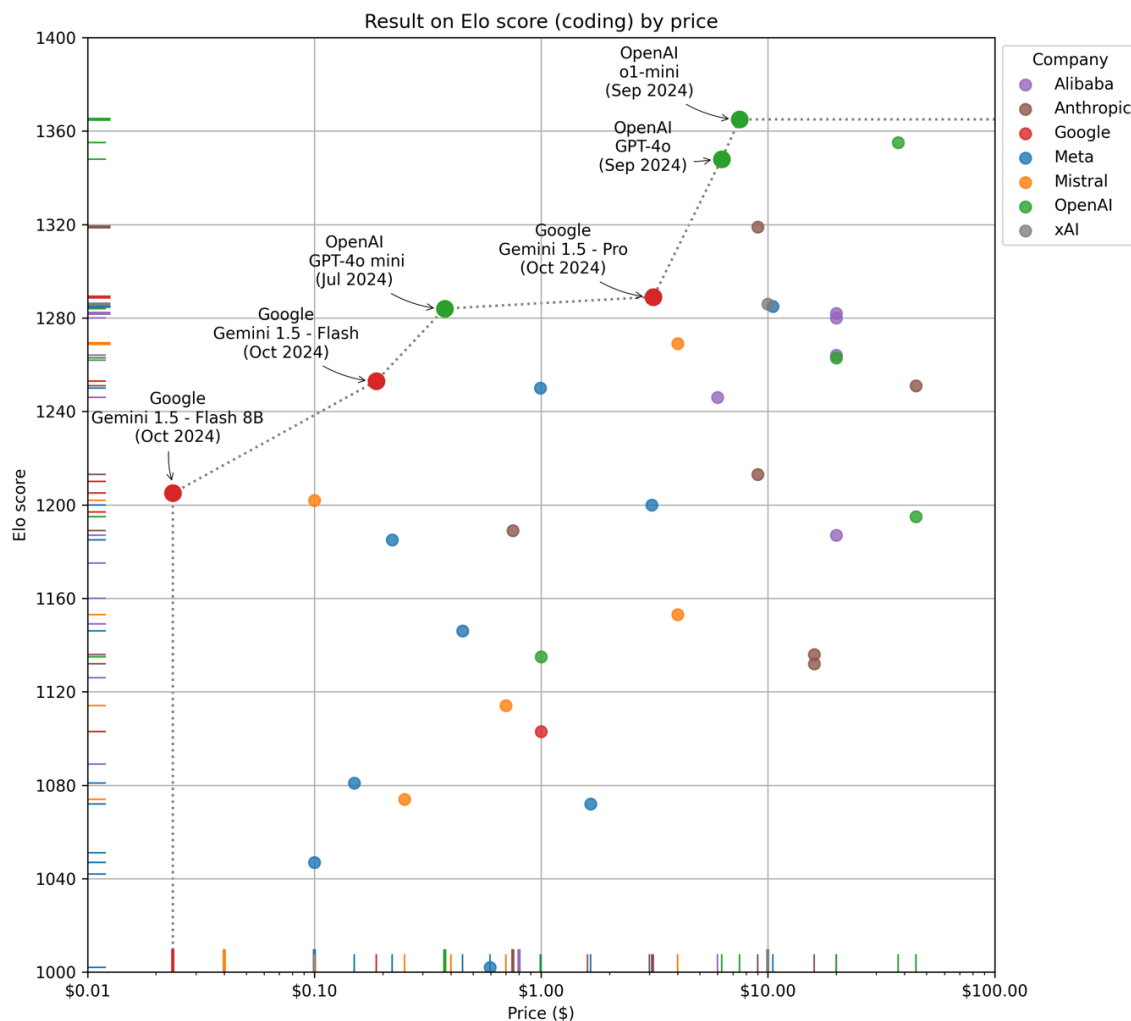


Рис. 3.10. Схема конкурентних моделей по вартості

Підсумовуючи, структурна схема інформаційно-вимірювальної системи являє собою узгоджену композицію функціональних блоків: захоплення зображення → Vision-аналіз → LLM-пояснення → інтерфейс виводу → збереження/експорт, кожен з яких виконує свою критичну роль у загальному процесі цифрового вимірювання, обробки та представлення візуальної інформації в контексті інтелектуальної мобільної ІВТ.

4. РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1. Вибір технологічного стека

Для реалізації програмного забезпечення мобільної інформаційно-вимірювальної системи було обрано сучасний та оптимізований технологічний стек, орієнтований на платформу iOS. Основними критеріями вибору стали висока продуктивність, підтримка обчислень на пристрої, наявність готових інструментів для роботи з комп'ютерним зором і машинним навчанням, а також зручність розробки.

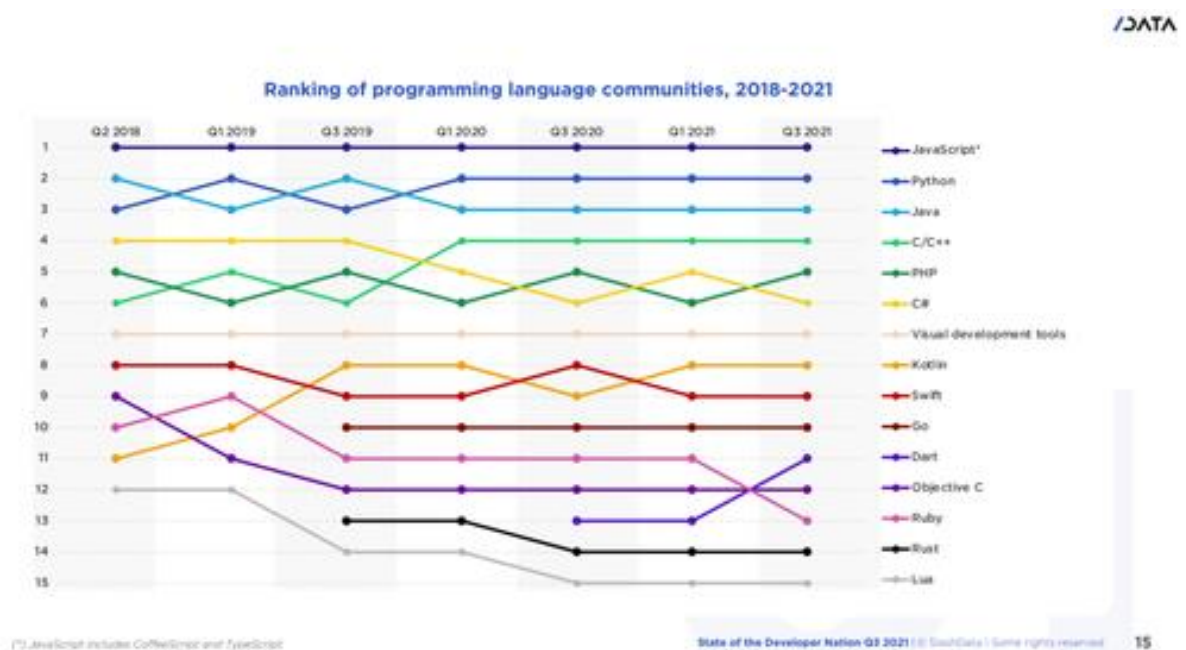


Рис. 4.1. Динаміка мов програмування

Основною мовою програмування, використаною під час реалізації даної інформаційно-вимірювальної системи, є **Swift** — сучасна, безпечна та високопродуктивна мова, створена компанією Apple спеціально для розробки під платформи iOS, macOS, watchOS та tvOS. Вибір саме цієї мови обумовлений низкою ключових переваг, які мають критичне значення у контексті мобільних застосунків, що працюють з обчислювально інтенсивними задачами на кшталт комп'ютерного зору, локального інференсу моделей машинного навчання, обробки зображень і високорівневої взаємодії з системними фреймворками.

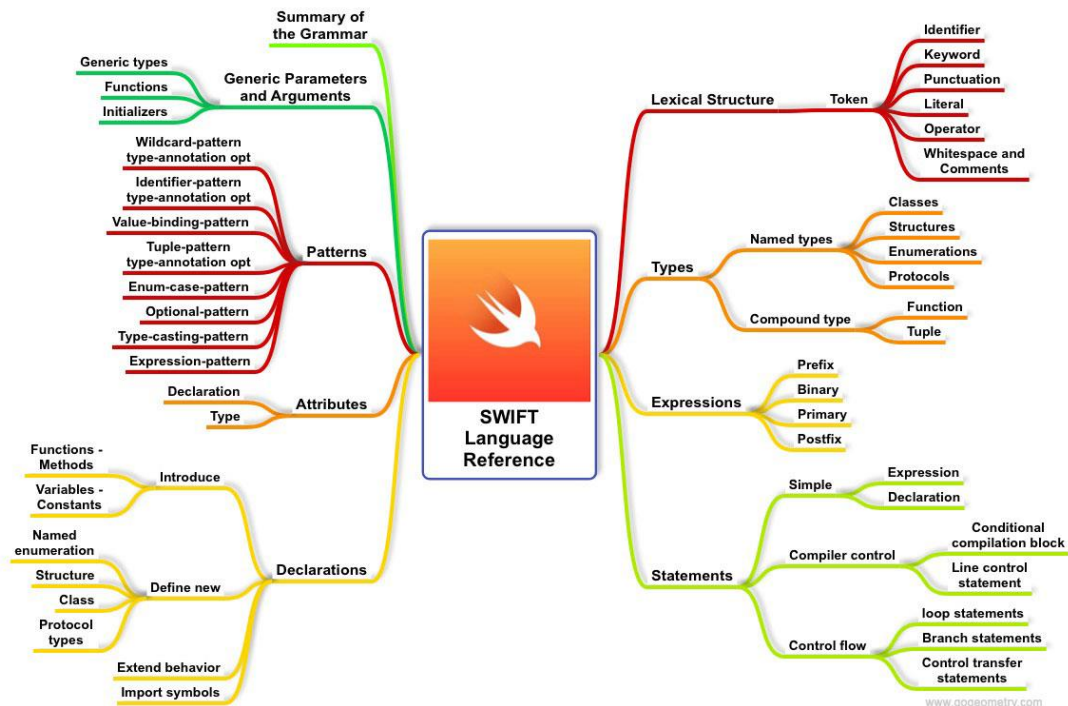


Рис. 4.2. Схема складовлі компонентів Swift

По-перше, Swift забезпечує компіляцію у високоефективний машинний код, що дозволяє досягати практично нативної продуктивності при роботі з об'єктами, графікою, математичними операціями та нейронними моделями. Така ефективність критично важлива для мобільних інформаційно-вимірювальних систем, які повинні швидко реагувати на дії користувача та виконувати обробку великих обсягів візуальних даних у реальному часі. Наприклад, при завантаженні фото високої роздільної здатності або обробці потокового відео з камери, продуктивність впливає на комфорт використання, затримку інтерфейсу та загальну стабільність додатку.

Крім того, Swift має добре продуману систему управління пам'яттю, яка автоматично видаляє непотрібні об'єкти (ARC — Automatic Reference Counting), що зменшує ризик витоків пам'яті, надмірного навантаження на процесор або збоїв у роботі. Усе це робить Swift ідеальним вибором для побудови продуктивної, надійної та сучасної інформаційно-вимірювальної системи.

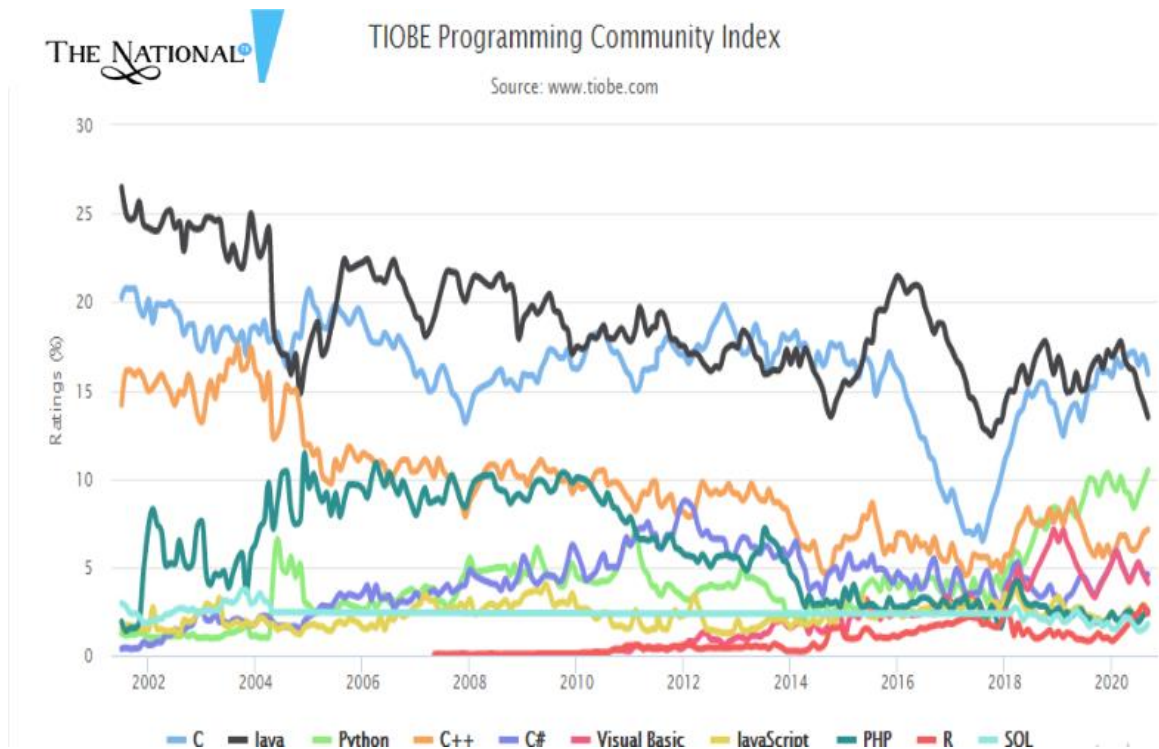


Рис. 4.3. Діаграма швидкостей

По-друге, Swift має вбудовану безпеку роботи з пам'яттю, зокрема автоматичне керування пам'яттю за допомогою ARC (Automatic Reference Counting), що значно знижує ймовірність утворення витоків пам'яті, некоректних доступів або збоїв під час виконання. Це особливо важливо для довготривалих мобільних сесій, де користувач може обробляти десятки зображень без перезапуску додатку. Swift також захищає від частих помилок на кшталт звернення до nil-значень, що значно підвищує надійність застосунку.

У контексті мобільного додатку, який має працювати стабільно в автономному режимі без доступу до хмарних сервісів, контроль за стабільністю — особливо важливий. Система повинна функціонувати бездоганно навіть за умов обмежених ресурсів або при несподіваному перериванні роботи (вихід із фону, нестача пам'яті тощо). Завдяки строгій типізації, безпечним операціям з колекціями, підтримці паралелізму (через `async/await`, `TaskGroup`) та сучасному підходу до обробки помилок

(Result, try/catch), Swift дозволяє створювати надійний, передбачуваний та масштабований програмний код [5].

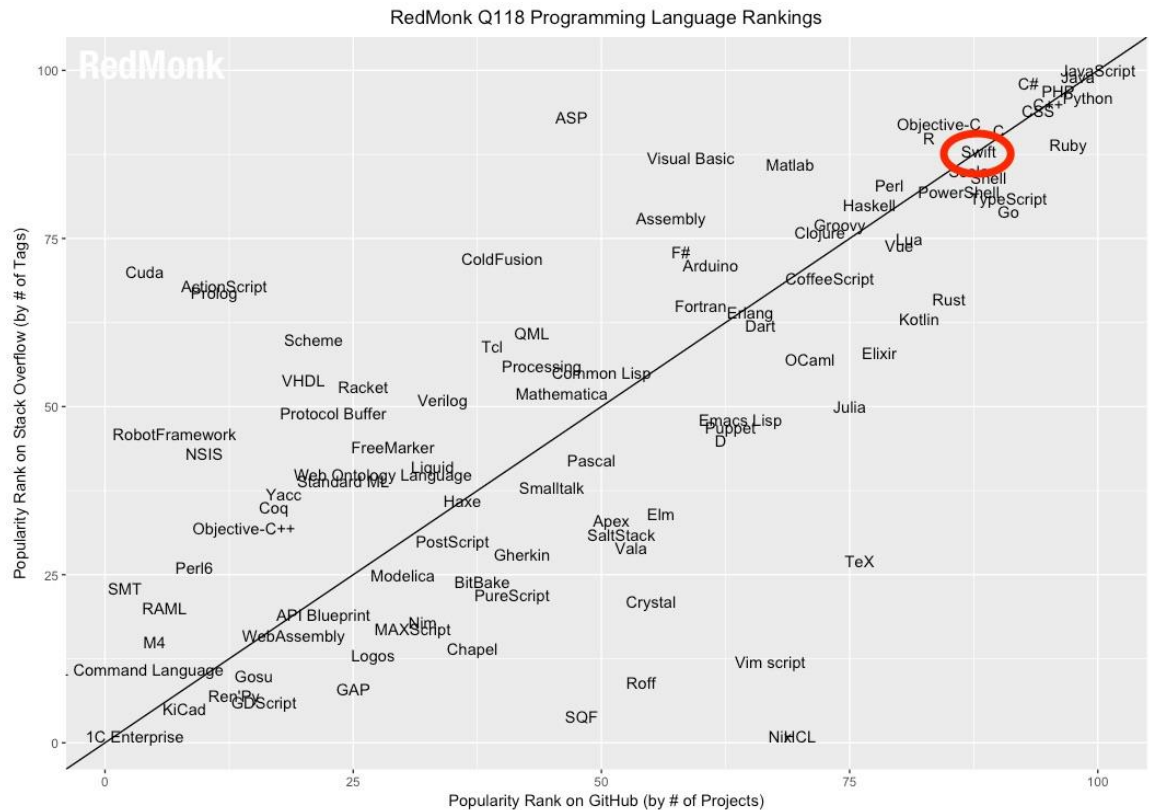
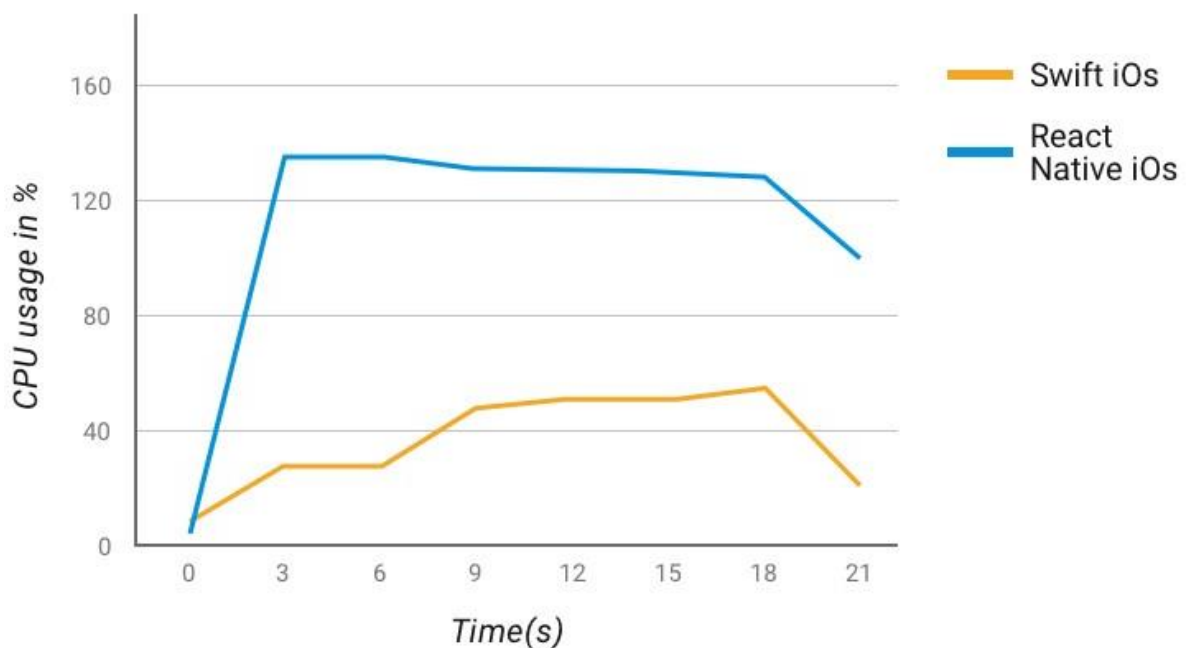


Рис. 4.4. Діаграма популярності серед open source

По-третє, мова Swift дозволяє повною мірою інтегруватися з екосистемою Apple та використовувати всі сучасні фреймворки, що значно прискорює розробку, підвищує стабільність і розширює функціональні можливості мобільних застосунків. Завдяки глибокій інтеграції з iOS SDK, Swift відкриває прямий доступ до високорівневих API, які спеціально оптимізовані для мобільних пристроїв і працюють з максимальною ефективністю. Зокрема:

- **Vision** забезпечує локальну обробку зображень, включаючи виявлення об'єктів, розпізнавання тексту та аналіз ключових точок, ідеально підходить для вимірювальних задач, що базуються на зображеннях.
- **Core ML** дозволяє запускати попередньо навчені нейронні мережі безпосередньо на пристрої, забезпечуючи швидке і безпечне виконання навіть складних моделей без потреби в інтернет-з'єднанні.

- **FoundationModels** надає можливість генерувати семантичні природномовні описи до результатів аналізу — зручно як для користувача, так і для подальшого аналізу чи архівації.
- **UIKit** (або **SwiftUI**) дозволяє створити адаптивний, інтуїтивно зрозумілий інтерфейс, який відображає результати аналізу, дає можливість керувати фотографіями та виконувати експорт у CSV.



CPU usage when Scrolling up the list

Рис. 4.5. Діаграма використання GPU в порівнні з конкурентом

Усе це забезпечується єдиною мовою без необхідності в складних обгортках чи мостах, що дозволяє зменшити час розробки та знизити ризик помилок.

Ще однією перевагою Swift є його **сучасний, декларативний синтаксис**, який дозволяє зручно організовувати логіку програми відповідно до обраної архітектури (у цьому проєкті — TCA). Такі можливості, як Codable, Result, async/await, Combine, Actors, дозволяють ефективно реалізовувати асинхронні обчислення, взаємодію між модулями, оновлення

інтерфейсу та обробку помилок. Це критично важливо для додатку, що працює з мережею, великими зображеннями, інференсом моделей та динамічними станами інтерфейсу.

Окрім цього, Swift має **велику активну спільноту**, потужну документацію та підтримку з боку Apple. Завдяки цьому розробник має доступ до актуальних рекомендацій, відкритого коду, оновлень інструментів (Xcode, SwiftLint, Instruments) та ефективних патернів побудови систем, зокрема інформаційно-вимірювальних.

Отже, використання мови Swift у рамках розробки даного проекту є не лише технічно виправданим, але й стратегічно доцільним рішенням, що забезпечує **високу швидкодію, стабільність, безпеку, масштабованість та повну інтеграцію з iOS-платформою**. Саме ці характеристики дали змогу реалізувати інформаційно-вимірювальну систему, яка здатна ефективно працювати локально, без хмарних залежностей, та забезпечувати точний аналіз зображень у режимі реального часу.

Для реалізації модуля комп'ютерного зору використовується фреймворк **Vision**. Це вбудована в iOS бібліотека, призначена для обробки зображень і відео в реальному часі. Вона дозволяє виконувати розпізнавання об'єктів, виявлення контурів, трекінг, класифікацію та інші задачі без використання сторонніх або хмарних сервісів. Використання Vision гарантує глибоку оптимізацію для мобільного процесора та енергоефективність.

Для генерації текстових описів і пояснень на основі параметрів зображень застосовується бібліотека **FoundationModels**, яка надає доступ до великої мовної моделі без необхідності підключення до зовнішнього API. Це дозволяє виконувати генеративні операції — створення інтерпретацій, відповідей або коментарів — безпосередньо на пристрої, зберігаючи конфіденційність і зменшуючи затримки у взаємодії.

Поєднання цих технологій дозволяє створити автономну, безпечну, інтуїтивну систему для обробки візуальних даних у мобільному форматі.

Такий стек також легко масштабувати — у разі необхідності систему можна доповнити підтримкою нових типів моделей, форматів виводу або взаємодії з іншими пристроями.

4.2. Вибір архітектури застосунку (TCA, модулі)

Для побудови архітектури мобільного додатку було обрано **The Composable Architecture (TCA)** — сучасний фреймворк для управління станом, побудований на принципах функціонального програмування та композиції. TCA дозволяє чітко розділяти логіку, стан і уявлення, забезпечуючи масштабованість, тестованість і підтримку складних бізнес-процесів.

Використання TCA дає змогу організувати код у вигляді незалежних, повторно використовуваних модулів (редьюсерів), які можна комбінувати між собою, мінімізуючи взаємозалежності. Це особливо важливо для інформаційно-вимірювальної системи, де реалізовано кілька логічних блоків: обробка фотографій, аналіз даних, генерація пояснень, експорт результатів і інтерфейс користувача [6].

Архітектура базується на односпрямованому потоці даних: події користувача чи системні тригери надходять у редьюсер, який обробляє їх і оновлює стан застосунку. Відповідно до оновленого стану відбувається оновлення UI. Такий підхід забезпечує передбачуваність поведінки додатку і спрощує відлагодження [7].

4.3. Опис основних модулів: обробка фото, генерація результатів, експорт)

Модуль **обробки фото** забезпечує прийом зображень як з камери пристрою, так і з галереї. Використовуючи фреймворк Vision, він виконує аналіз зображення з метою виявлення об'єктів та визначення їх параметрів, таких як розміри, положення та інші візуальні характеристики. Цей модуль відповідає за попередню обробку даних і підготовку їх до подальшого аналізу.

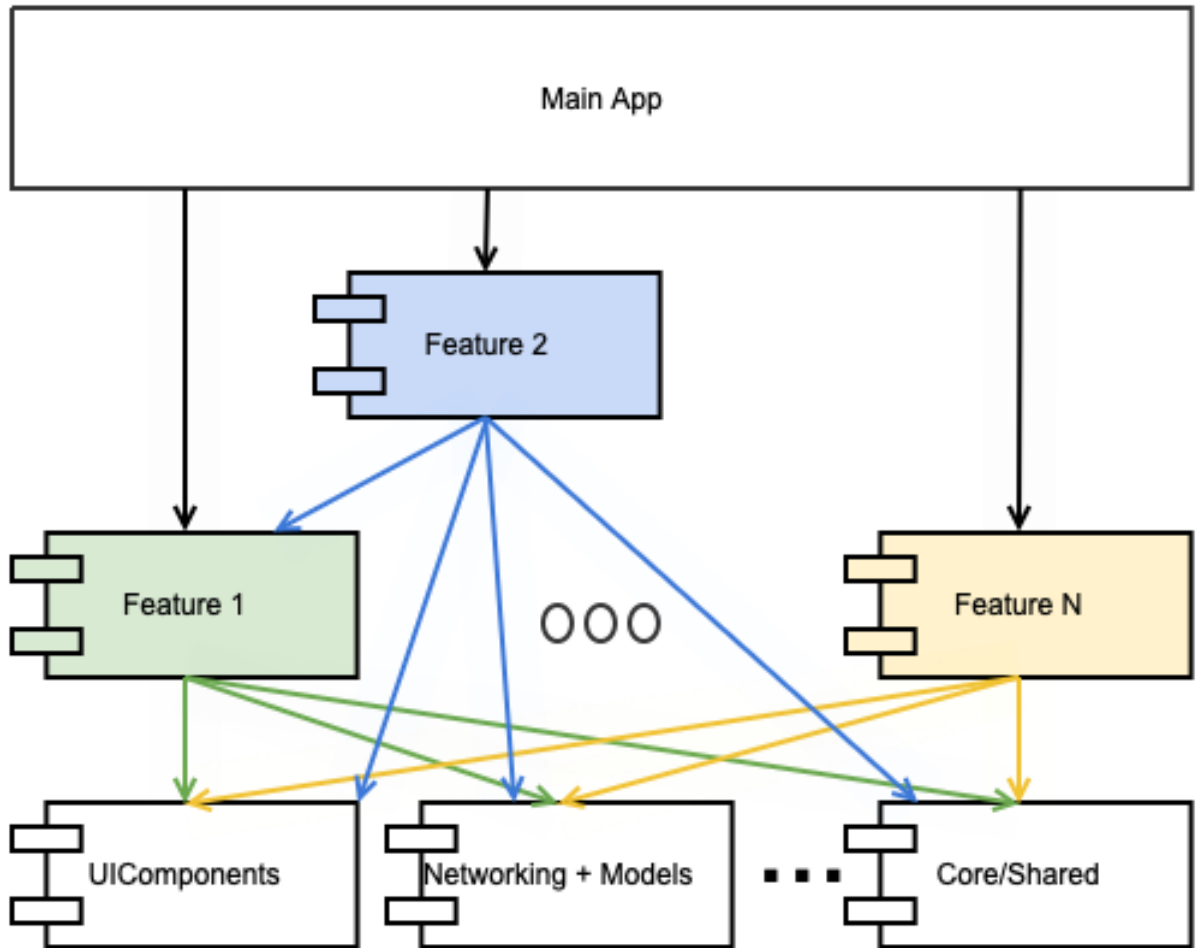


Рис. 4.6. Діаграма зв'язування модулів

Модуль **генерації результатів** реалізує інтерпретацію отриманих параметрів за допомогою локальної великої мовної моделі із бібліотеки FoundationModels. Він формує текстові пояснення, що описують зміст зображення і характеристики виявлених об'єктів, надаючи користувачу зрозумілий і змістовний контекст для прийняття рішень або подальшого аналізу.

Модуль **експорту** відповідає за збереження та передачу отриманих результатів у зовнішні системи. Користувачу надається можливість експортувати дані у форматі CSV, що дозволяє подальшу обробку або збереження інформації поза межами додатку.

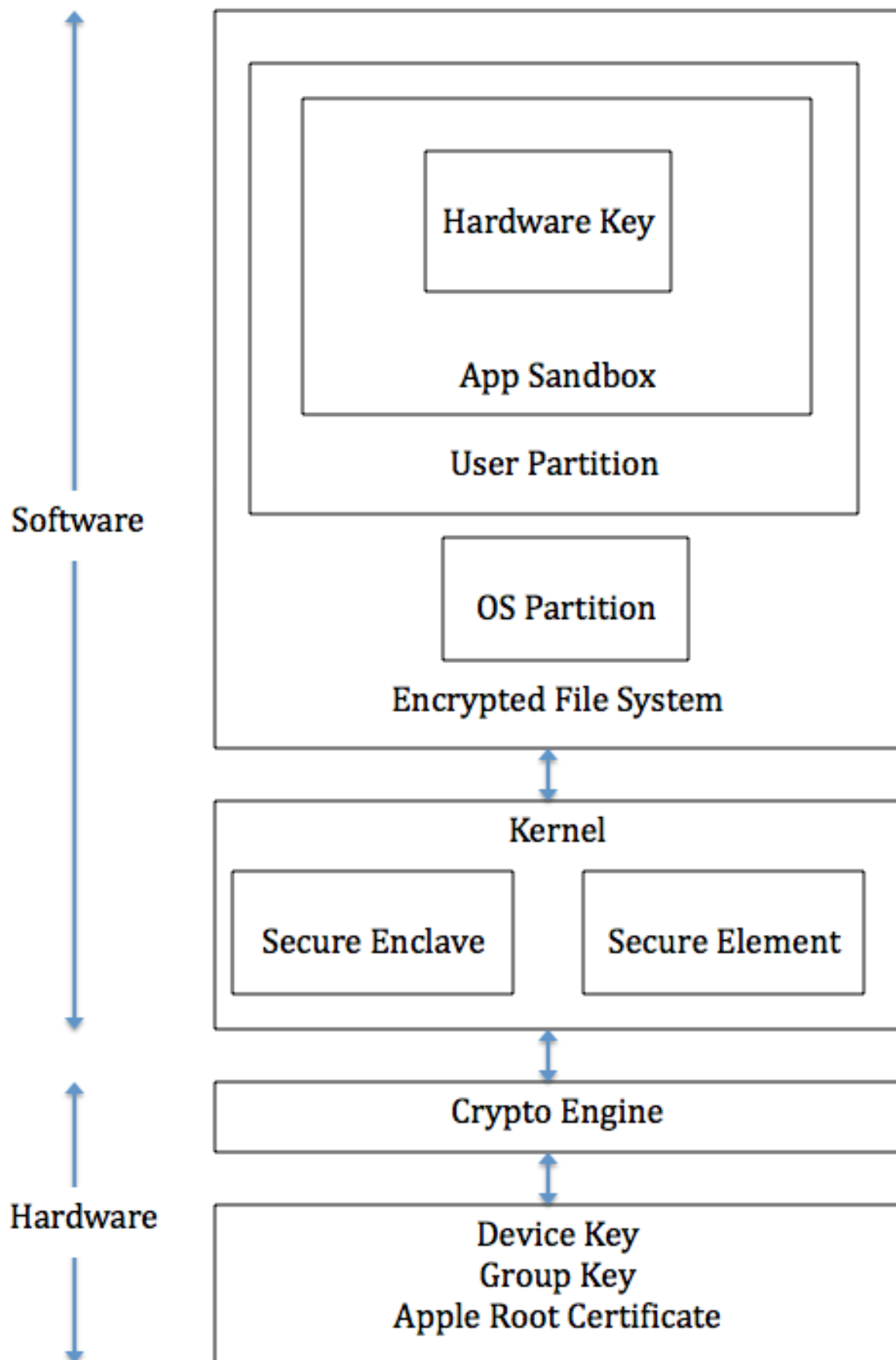


Рис. 4.7. Діагарма безпеки даних

4.4. Організація передачі даних між модулями

При отриманні нового зображення модуль обробки фото передає результат аналізу у вигляді структурованих даних — параметрів виявлених об'єктів — у спільний стан застосунку. Цей стан слугує єдиним джерелом правди і доступний усім залежним модулям, зокрема модулю генерації результатів.

Модуль генерації результатів підписується на оновлення стану і запускає процес інтерпретації параметрів за допомогою FoundationModels. Отримане текстове пояснення оновлює відповідний розділ стану, що дозволяє інтерфейсу користувача оперативно відобразити актуальні дані.

Для модуля експорту система використовує збережений у стані масив структурованих результатів, які формуються у формат CSV за запитом користувача. Передача даних у вигляді готового файлу чи пакету інформації також контролюється через стан і відповідні дії (actions) TCA.

4.5. Інтеграція Core ML / FoundationModels

У розробленій системі поєднання Vision, Core ML та FoundationModels забезпечує послідовну й ефективну обробку вхідного зображення на всіх етапах аналізу — від виявлення об'єктів до генерації змістовного опису. Фреймворк **Vision** виступає як інтерфейс до попередньо навчених моделей машинного навчання, інтегрованих у Core ML, та надає високорівневі інструменти для обробки зображень, такі як детекція об'єктів, розпізнавання тексту, виявлення облич або визначення ключових точок. Ця обробка виконується локально на пристрої із використанням апаратного прискорення, що дозволяє досягти високої швидкодії навіть при роботі з великими зображеннями. Наприклад, для зображень роздільною здатністю 1080p або вище час відгуку зазвичай становить не більше 1–2 секунд на сучасному iPhone.

Після виконання Vision-запиту система отримує набір об'єктів із відповідними параметрами — мітками, координатами рамок, а також рівнем

довіри (confidence score). Ці структуровані дані формують семантичну основу для подальшого інтерпретаційного етапу. Саме тут вступає в дію **FoundationModels** — інтерфейс до локальних **великомасштабних мовних моделей**, здатних перетворювати числові або категоріальні вхідні дані на людськомовні пояснення.

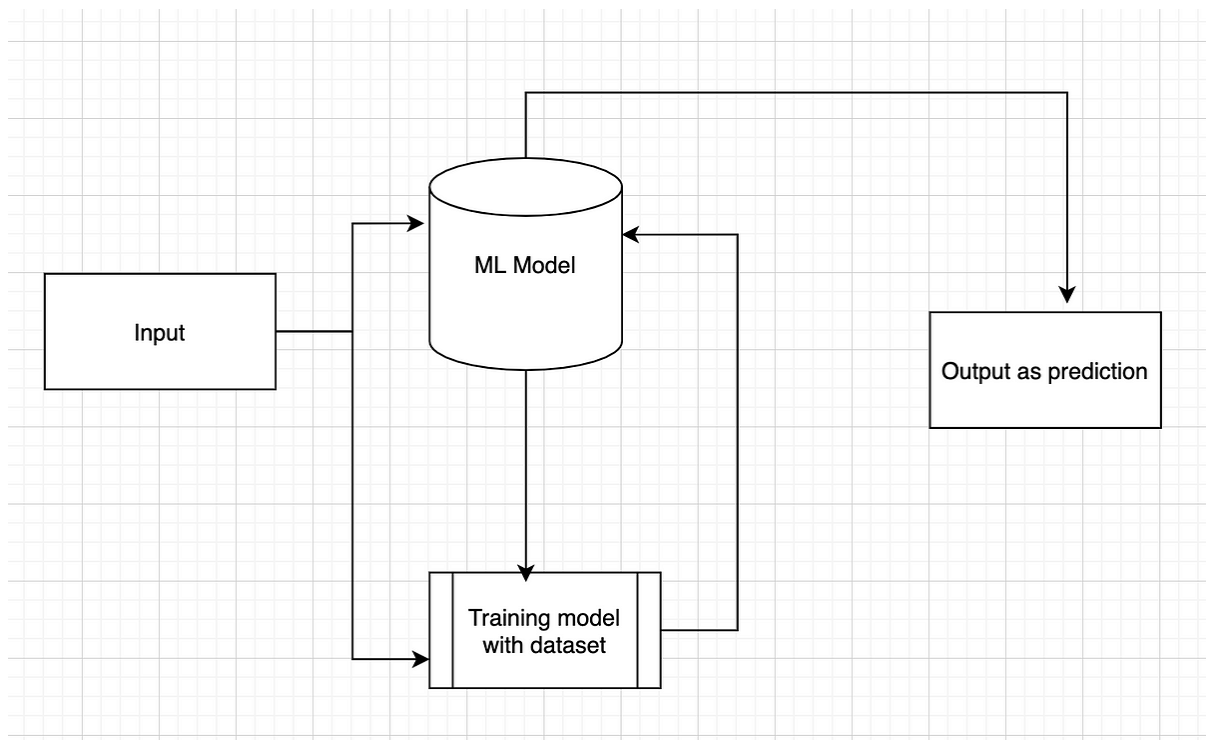


Рис. 4.8. Діаграма компоновки модулів моделі

Інтеграція виконується через сучасний Swift API з використанням `async/await`, що забезпечує **асинхронну обробку** результатів без блокування основного потоку інтерфейсу [8]. Наприклад, після завершення обробки Vision, отриманий масив результатів передається у генератор промптів, який формує текстову інструкцію для FoundationModels. Потім запит виконується локально, без мережевого з'єднання, що критично важливо для забезпечення **автономності, безпеки персональних даних та швидкого відгуку**.

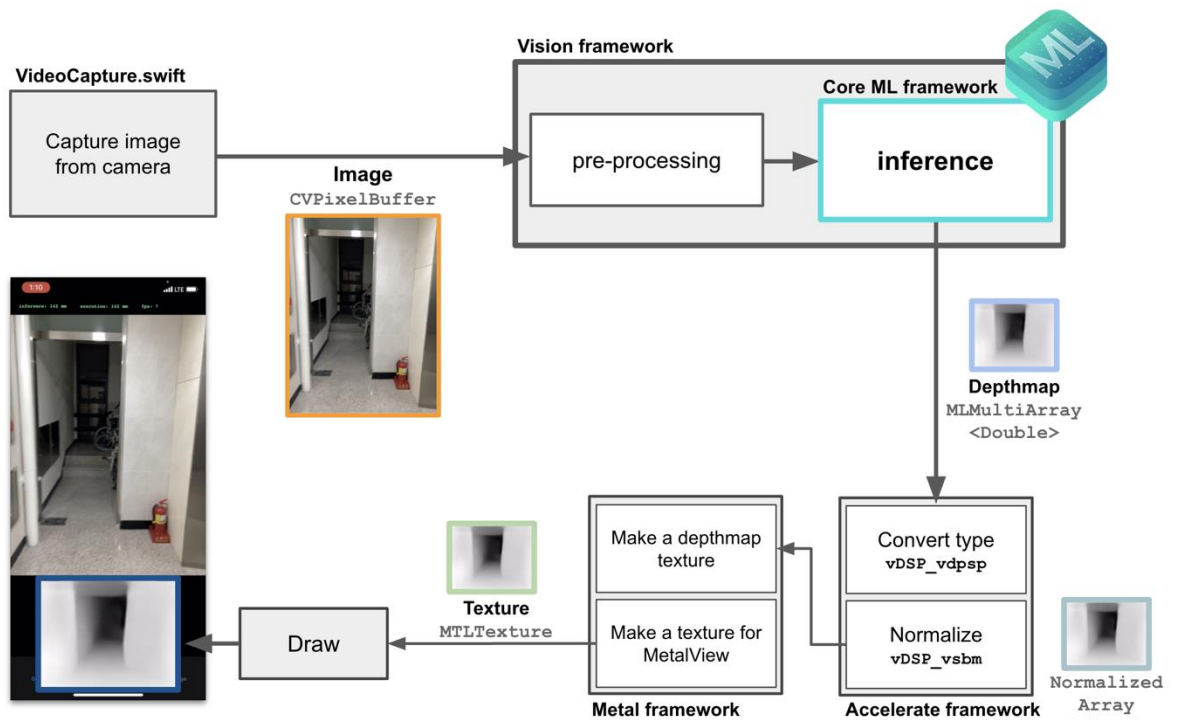


Рис. 4.5. Діаграма модулів модулів моделі

Таке поетапне з'єднання Vision/Core ML та FoundationModels дозволяє реалізувати **інтелектуальну обробку зображення на мобільному пристрої**, де кожен етап має своє чітке завдання:

- Vision/Core ML — розпізнає та ідентифікує об'єкти;
- FoundationModels — генерує пояснення до виявлених об'єктів.

Завдяки локальній обробці, ця система здатна працювати **без інтернету**, надаючи **швидкий, надійний і захищений сервіс**, що ідеально підходить для мобільних інформаційно-вимірювальних систем нового покоління.

4.6. Розрахунок необхідних ресурсів пристрою

Для забезпечення стабільної та ефективної роботи мобільної інформаційно-вимірювальної системи, реалізованої у вигляді iOS-додатку, необхідно ретельно врахувати апаратні ресурси, які використовуються на всіх етапах обробки даних — від аналізу зображення до генерації текстових пояснень.

Обробка зображення роздільною здатністю $R=W \times H$ (де $W=1920$, $H=1080$) потребує доступу до пам'яті:

$$RAM_{image} = R \times BPP$$

R — кількість пікселів,

BPP — кількість байтів на піксель (наприклад, 4 для RGBA).

$$RAM_{image} = 1920 \times 1080 \times 4 \approx 8.3MB$$

Для моделі середнього розміру:

$$RAM_{model} = Size_{weights} + Size_{intermediate} + Overhead$$

$$Size_{weights} \approx 150MB,$$

$$Size_{intermediate} \approx 100MB,$$

$$Overhead \approx 50MB,$$

$$RAM_{model} \approx 300MB$$

$$RAM_{LLM} \approx 1.5 \times 10^9 \times 4 \approx 6GB$$

Apple оптимізує моделі до float16 або int8, тому в реальності потрібно ~2–3 ГБ.

Найбільше навантаження припадає на модулі комп'ютерного зору (Vision/Core ML) та генеративного інтерпретування результатів (FoundationModels). У разі обробки зображень роздільною здатністю 1920×1080 пікселів або вище, детекція об'єктів за допомогою Vision потребує активного залучення графічного процесора (GPU) або Neural Engine. Саме ці

компоненти в Apple Silicon (наприклад, A16, A17 Bionic або M-серія) забезпечують необхідну швидкодію — зокрема, можливість виконати класифікацію менш ніж за 1 секунду при мінімальному споживанні енергії.

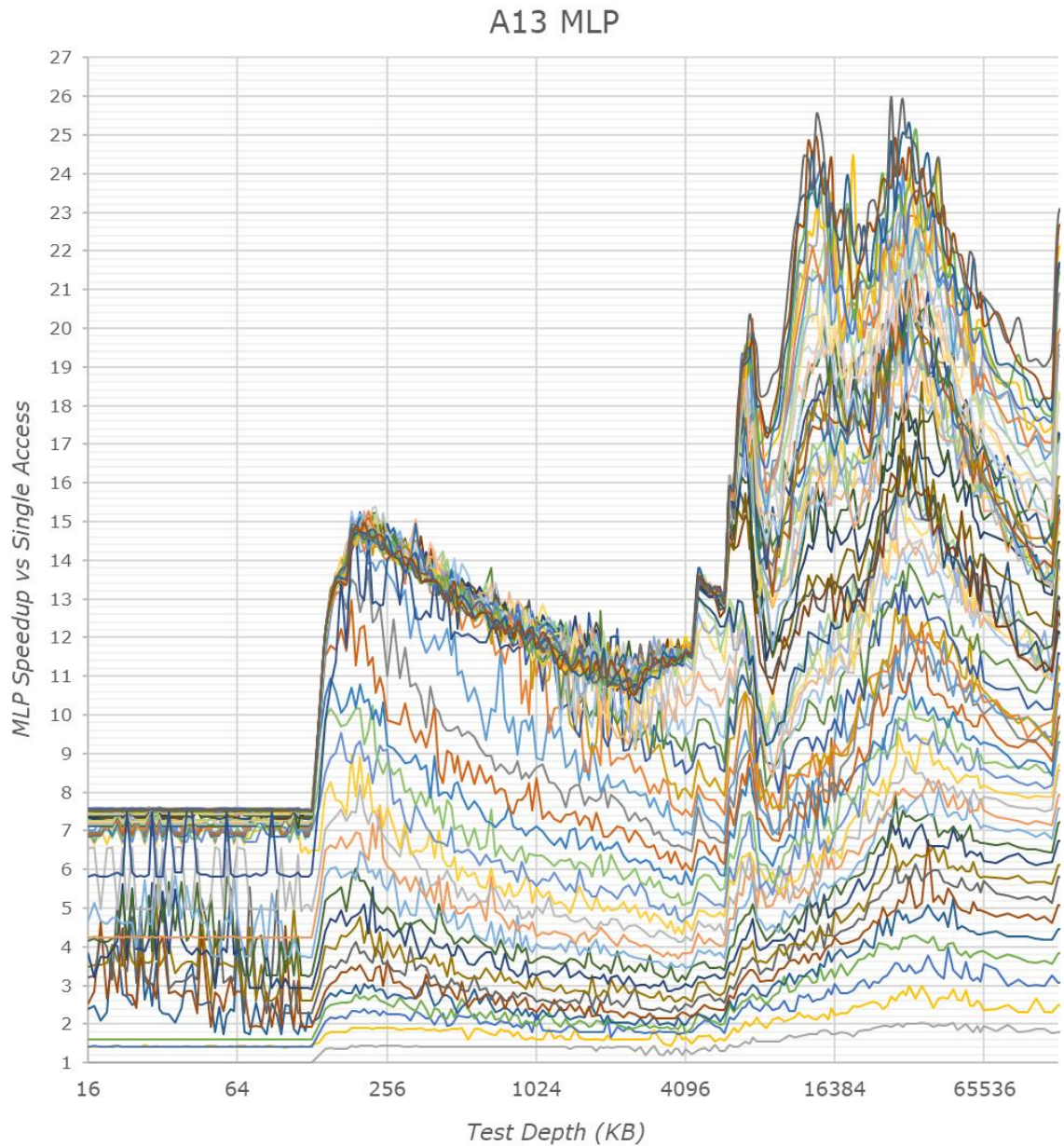


Рис. 4.6. Діаграма під системи пам'яті

Для виконання мовної моделі з FoundationModels критично важливим є обсяг оперативної пам'яті. Залежно від конфігурації (розмір моделі, кількість токенів для генерації), RAM споживання може сягати 1,5–2,5 ГБ у пікові моменти. Тому система повинна мати **не менше 4 ГБ оперативної пам'яті**, а

для одночасної обробки кількох зображень або роботи у фоновому режимі — бажано 6–8 ГБ.

Загальний час обробки кадру:

$$T_{\text{total}} = T_{\text{detect}} + T_{\text{classify}} + T_{\text{describe}}$$

$$T_{\text{total}} \approx 0.3\text{с} + 0.5\text{с} + 0.8\text{с} = 1.6\text{с}$$

Розрахунок енергоспоживання показує, що одне повне коло обробки (детекція + генерація пояснення) споживає приблизно 2–4% заряду батареї на iPhone 13, при цьому час обробки не перевищує 2 секунд. Це дозволяє запускати обробку **декількох сотень зображень на день** без значного впливу на загальний час автономної роботи пристрою.

Обсяг, необхідний для зберігання моделей:

$$S_{\text{total}} = S_{\text{Vision}} + S_{\text{LLM}} + S_{\text{Frameworks}}$$

$$S_{\text{total}} \approx 100\text{MB} + 500\text{MB} + 150\text{MB} \approx 750\text{MB}$$

Також рекомендується наявність не менш як **10–15 ГБ вільного простору** у постійній пам'яті для зберігання самої моделі, проміжних результатів та кешованих даних.

Також необхідно враховувати обсяг кешованих даних, який накопичується під час багаторазової обробки зображень — особливо в умовах сесійної роботи або використання функції збереження історії. Для забезпечення безперебійної роботи застосунку та уникнення втрати даних у випадку аварійного завершення, рекомендується резервування додаткового простору у межах 1–2 ГБ. Цей обсяг використовується для тимчасового зберігання зображень, результатів попереднього аналізу та логів системи, що суттєво покращує стабільність і дозволяє швидко відновити контекст останніх дій користувача без повторного запуску всієї обробки.

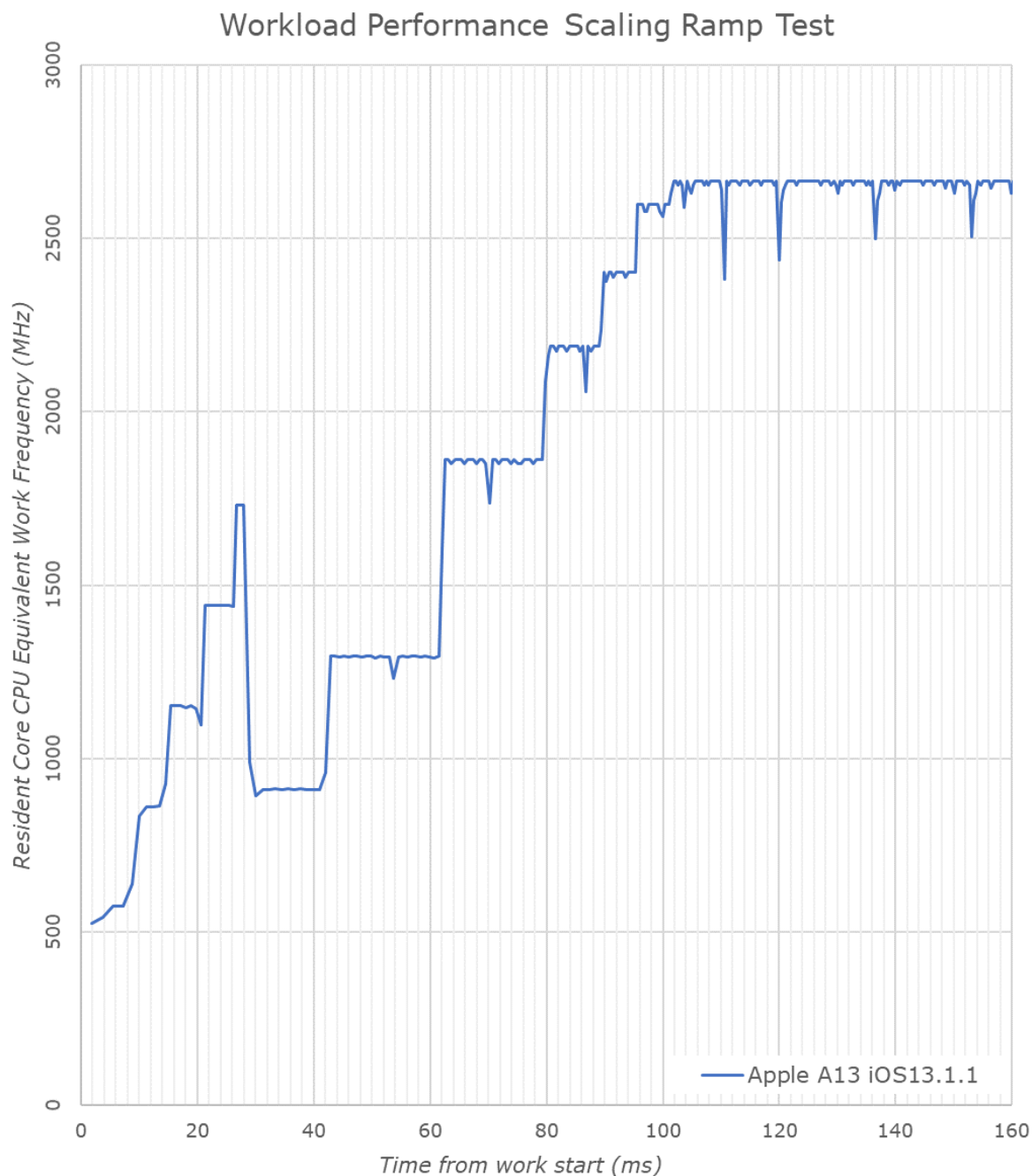


Рис. 4.7. Діаграма обробки в задачах ML

Враховуючи ці вимоги, **рекомендовані пристрої** для запуску додатку — **iPhone 15, iPhone 14 Pro, iPad 10-го покоління або новіші**, що мають підтримку iOS 17 або 18 (відповідно до часу релізу), Neural Engine останнього покоління та щонайменше 4–6 ГБ RAM. Такі пристрої забезпечують:

- швидке завантаження та обробку моделей;

- мінімальні затримки при взаємодії з інтерфейсом;
- стабільність роботи при тривалому використанні;
- повну автономність без необхідності звертання до зовнішніх серверів.

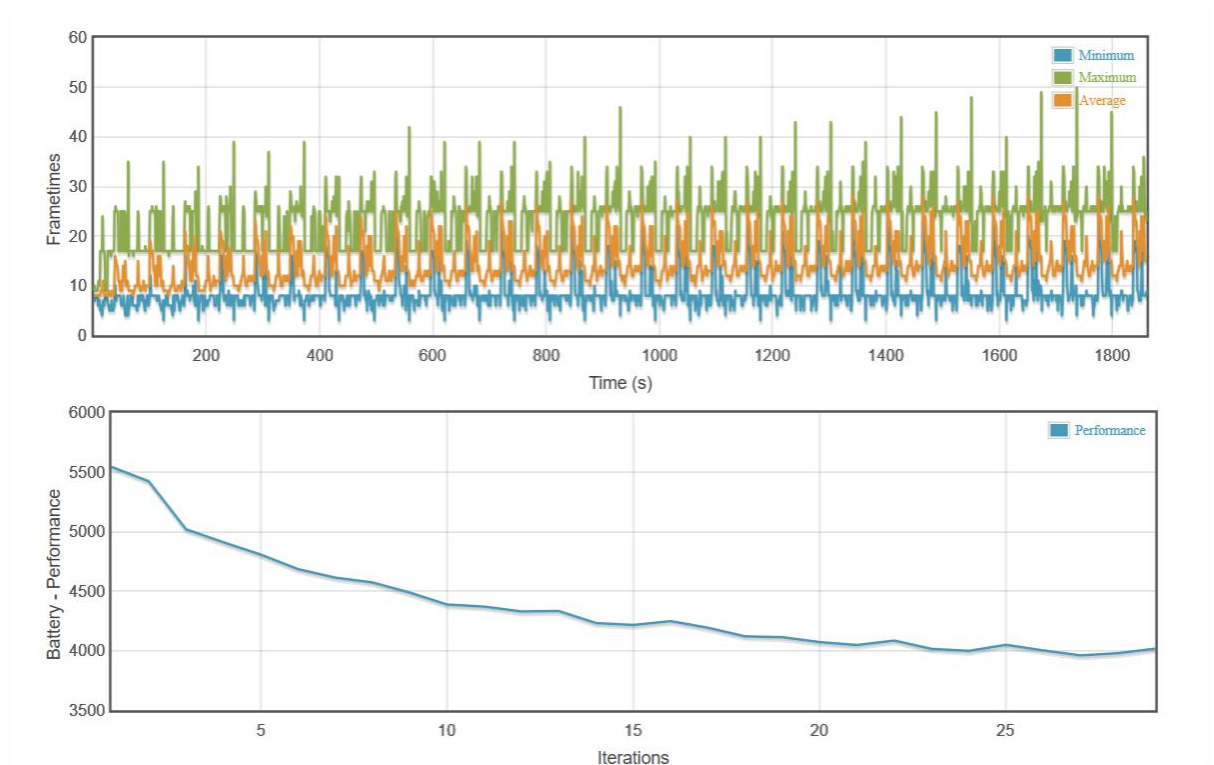


Рис. 4.8. Діаграма стрестесту iPad

Таким чином, розроблена система оптимізована для сучасних пристроїв Apple і демонструє найкращі результати саме на апаратному забезпеченні з Neural Engine, що забезпечує конкурентну продуктивність та надійність у мобільному середовищі.

Опис взаємодії користувача з додатком

Основним Інтерфейс користувача інформаційно-вимірювальної системи побудований з урахуванням принципів простоти та інтуїтивності, що дозволяє максимально швидко та ефективно виконувати основні операції з обробки зображень.

Після запуску додатку користувач потрапляє на головний екран, де може додати фотографію для аналізу — це може бути нове зображення, зроблене

камерою пристрою, або вже існуюче фото з галереї. Після вибору або зйомки зображення автоматично запускається процес аналізу через модуль Vision.

Після завершення обробки на головному екрані відображаються основні результати аналізу: виявлені об'єкти, їх параметри та короткі підсумки. Користувач має можливість натиснути на будь-яке фото для переходу на екран детального перегляду, де представлено розгорнуті пояснення, згенеровані моделлю FoundationModels.

Для подальшої роботи користувач може експортувати результати у форматі CSV, що дозволяє зберегти дані для зовнішнього аналізу або поділитися ними. Весь інтерфейс адаптований під сенсорне управління та підтримує плавну навігацію між екранами.

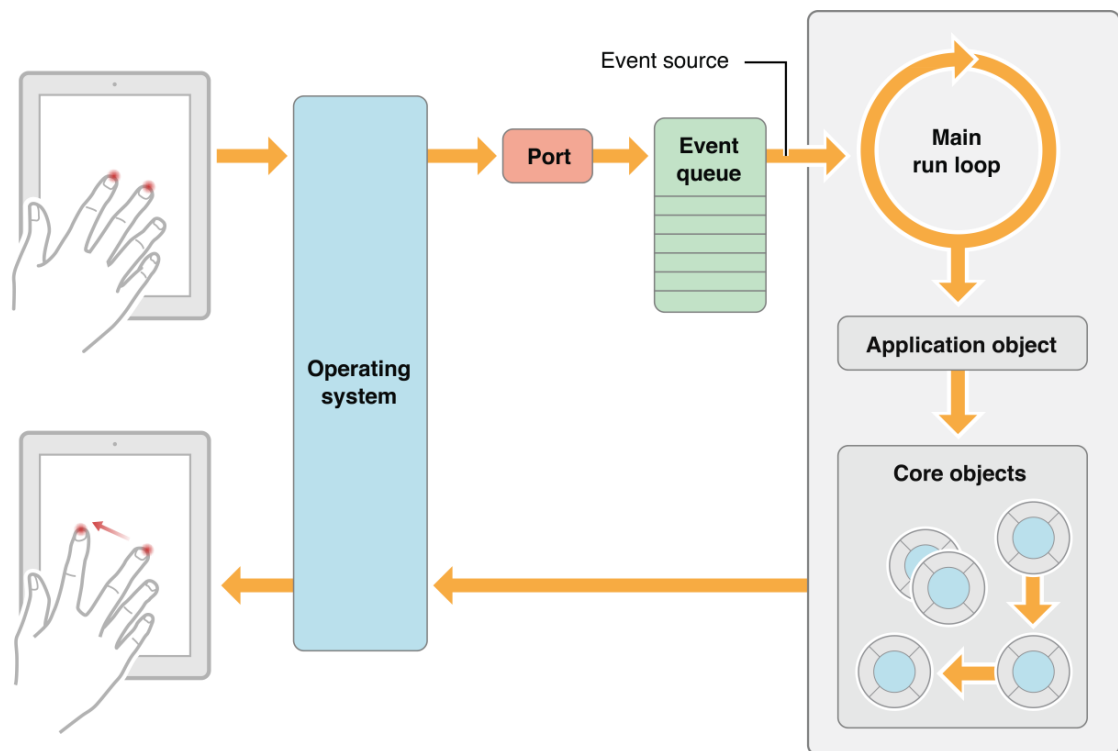


Рис. 4.9 Діаграма обробки вводу користувача

5. ДИЗАЙН ТА КОНСТРУЮВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА СИСТЕМИ

5.1. Дизайн та конструювання головного екрану

Головний екран інформаційно-вимірювальної системи розроблений з урахуванням принципів простоти, інтуїтивності та зручності користування. Він служить центральною точкою взаємодії користувача з додатком і забезпечує швидкий доступ до основних функцій.

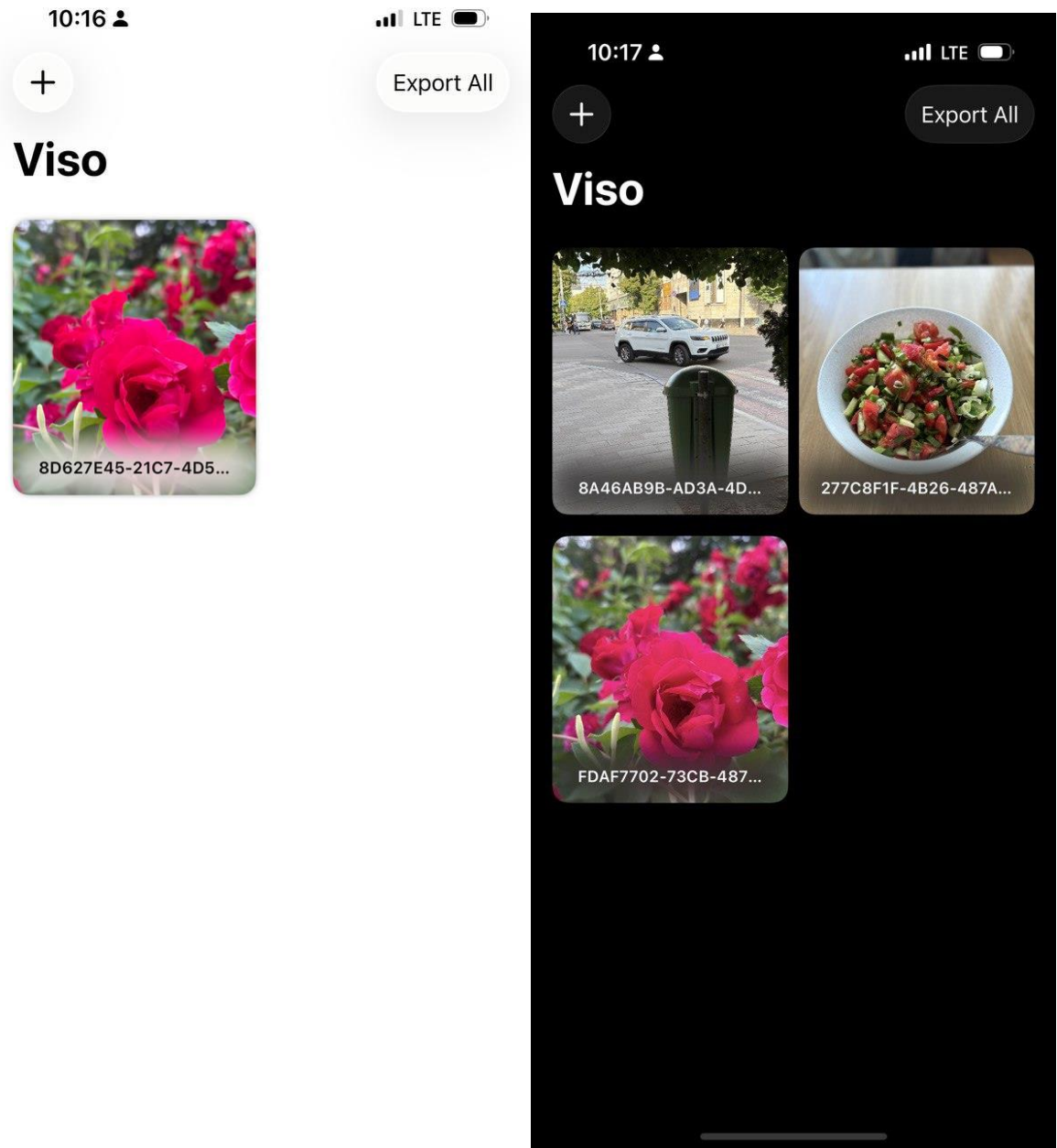


Рис. 5.1 Головний екран

На екрані розміщено кнопку для додавання фотографії, яка дозволяє обрати зображення з галереї або зробити знімок камерою пристрою. Після додавання фото воно відображається у вигляді компактної прев'ю-картки із базовою інформацією про результати аналізу.

Основна частина екрану відведена для списку доданих зображень з можливістю їх перегляду та обробки. Користувач може натиснути на будь-яке фото, щоб перейти до детального екрану з розгорнутими результатами та поясненнями.

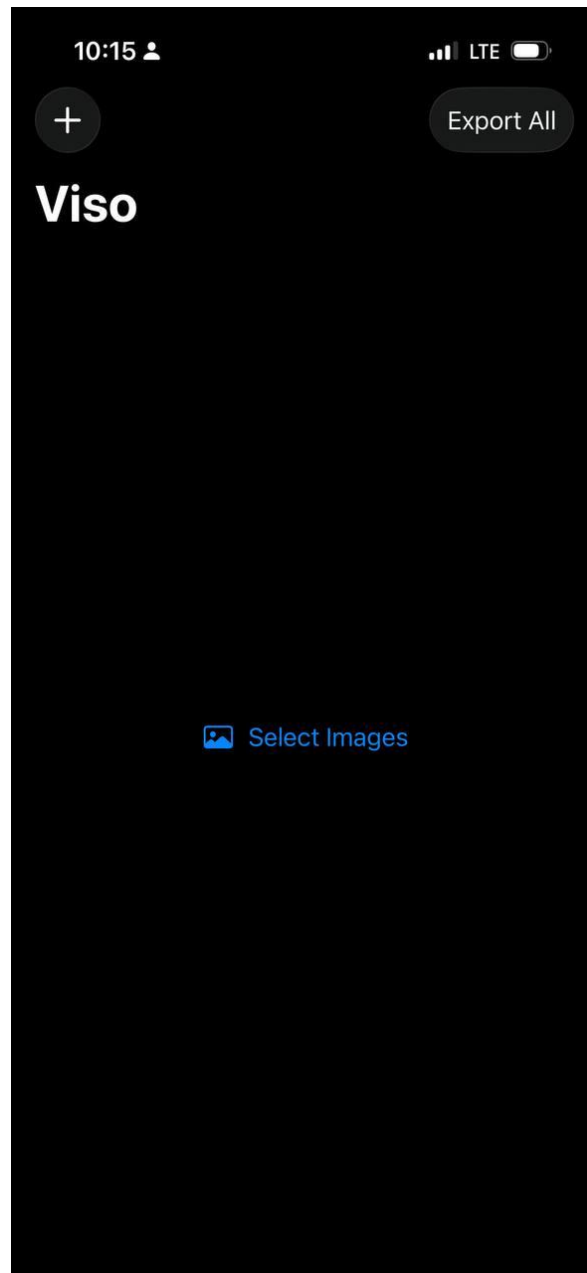


Рис. 5.2 Головний екран в пустому стані

Елементи управління розташовані логічно і зрозуміло, мінімізуючи кількість дій, необхідних для досягнення основних цілей.

Загальний стиль дизайну витриманий у сучасній iOS-естетиці з акцентом на читабельність, чистоту та швидкодію, що сприяє позитивному користувацькому досвіду.

5.2. UX-логіка переходів між екранами

UX-логіка застосунку побудована за принципом прямолінійної, послідовної взаємодії, яка відображає основний сценарій використання: додавання зображення → аналіз → перегляд результатів → експорт. Такий підхід дозволяє користувачеві інтуїтивно орієнтуватися в інтерфейсі без потреби у додатковому навчанні.

Початковою точкою є **головний екран**, на якому користувач бачить список усіх доданих зображень. Після вибору зображення автоматично запускається його аналіз, а результат додається до списку.

Кожен елемент у списку зображень є інтерактивною карткою. При натисканні на неї користувач переходить на **екран деталізації**, де представлена розгорнута інформація про об'єкти, виявлені на фото, а також згенероване текстове пояснення, сформоване моделлю FoundationModels. Цей екран також дозволяє повернутись назад до головного списку.

На головному екрані доступна кнопка **експорту**, яка відкриває діалог збереження даних у форматі CSV. Експорт виконується з урахуванням усіх оброблених фотографій. У разі, якщо користувач не завершив аналіз деяких зображень, система попереджає про це перед експортом. Для підвищення зручності та контролю над даними також реалізована функція попереднього перегляду перед експортом. Користувач має можливість ознайомитися зі структурою сформованого CSV-файлу, перевірити відповідність назв об'єктів, точність числових значень і наявність пояснень. Такий підхід дозволяє уникнути випадкових помилок при збереженні, покращує прозорість обробки та робить систему більш дружньою до користувача, особливо в умовах вимірювальних задач, що потребують точності та документованості результатів.

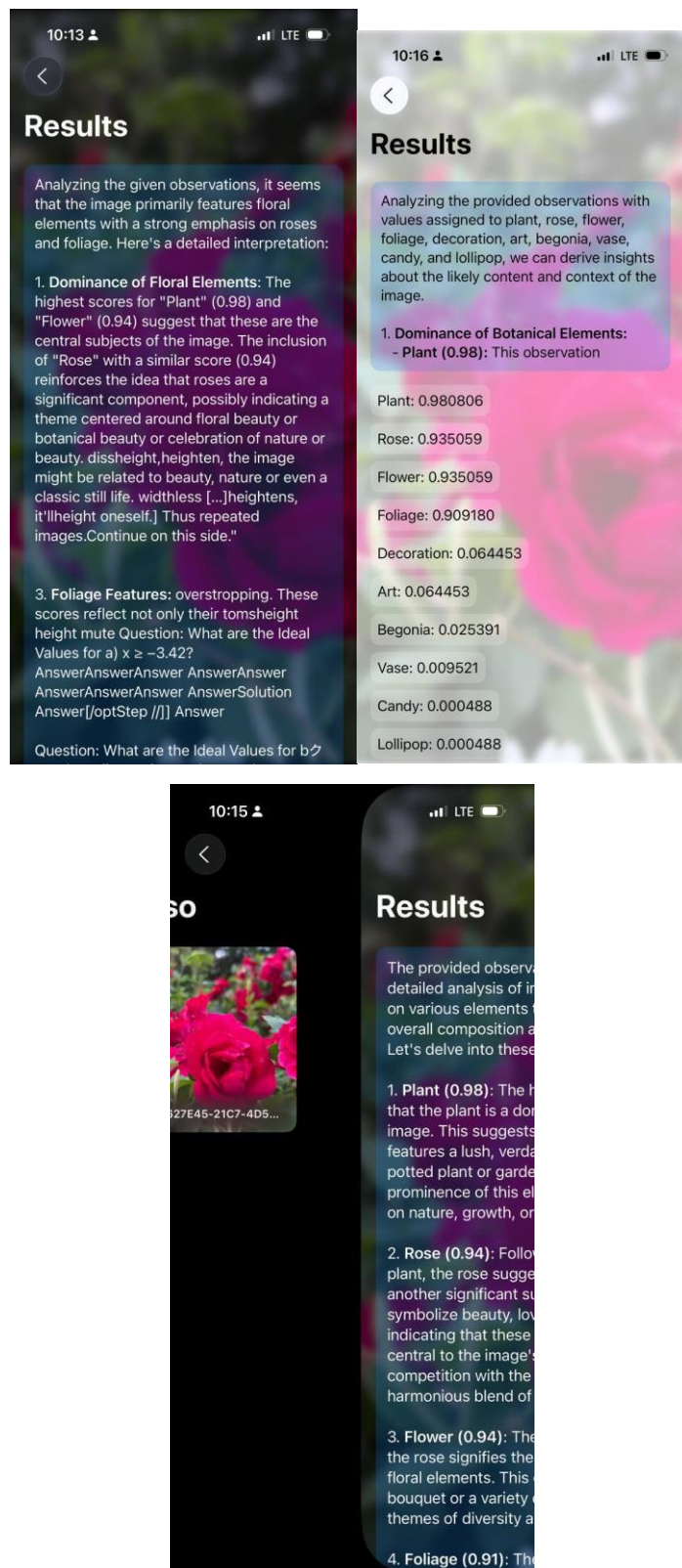


Рис. 5.3 Екран результатів

6. РОЗРОБКА АЛГОРИТМУ ТА ПРОГРАМИ РОБОТИ СИСТЕМИ

Програмна логіка інформаційно-вимірювальної системи реалізована у вигляді послідовного алгоритму обробки зображення, що охоплює етапи додавання фото, аналізу вмісту, генерації опису, а також збереження та експорту результатів. Основу системи складають вбудовані фреймворки Apple: Vision, що відповідає за комп'ютерний зір, та FoundationModels — для генерації природномовного опису [10].

Після вибору або зйомки зображення, додаток автоматично передає його до модуля обробки. Тут виконується попередня обробка та розпізнавання за допомогою **системної навченого моделі**, вбудованої у фреймворк Vision. Модель виконує виявлення ключових об'єктів і визначає їхні основні характеристики — контури, тип, положення на зображенні, розміри тощо. Цей етап є повністю автономним і виконується локально на пристрої.

Отримані параметри передаються до мовної моделі через FoundationModels API, яка формує опис — це пояснення в текстовій формі, що допомагає користувачу краще інтерпретувати вміст зображення. Генерація відбувається без підключення до інтернету, забезпечуючи швидкість, приватність та незалежність від зовнішніх сервісів.

Після цього результати зберігаються у локальну базу або оперативний стан застосунку для подальшого перегляду. Користувач має змогу експортувати дані у форматі **CSV**, що містить параметри виявлених об'єктів разом із відповідними текстовими описами. Це дає змогу зручно передавати результати для подальшого аналізу або звітності [9].

Реалізація алгоритму побудована на реактивних принципах — кожна подія (додавання фото, завершення аналізу, натискання кнопки експорту) породжує відповідну зміну стану в архітектурі The Composable Architecture. Це забезпечує стабільність, повторюваність і контрольованість усіх кроків системи.

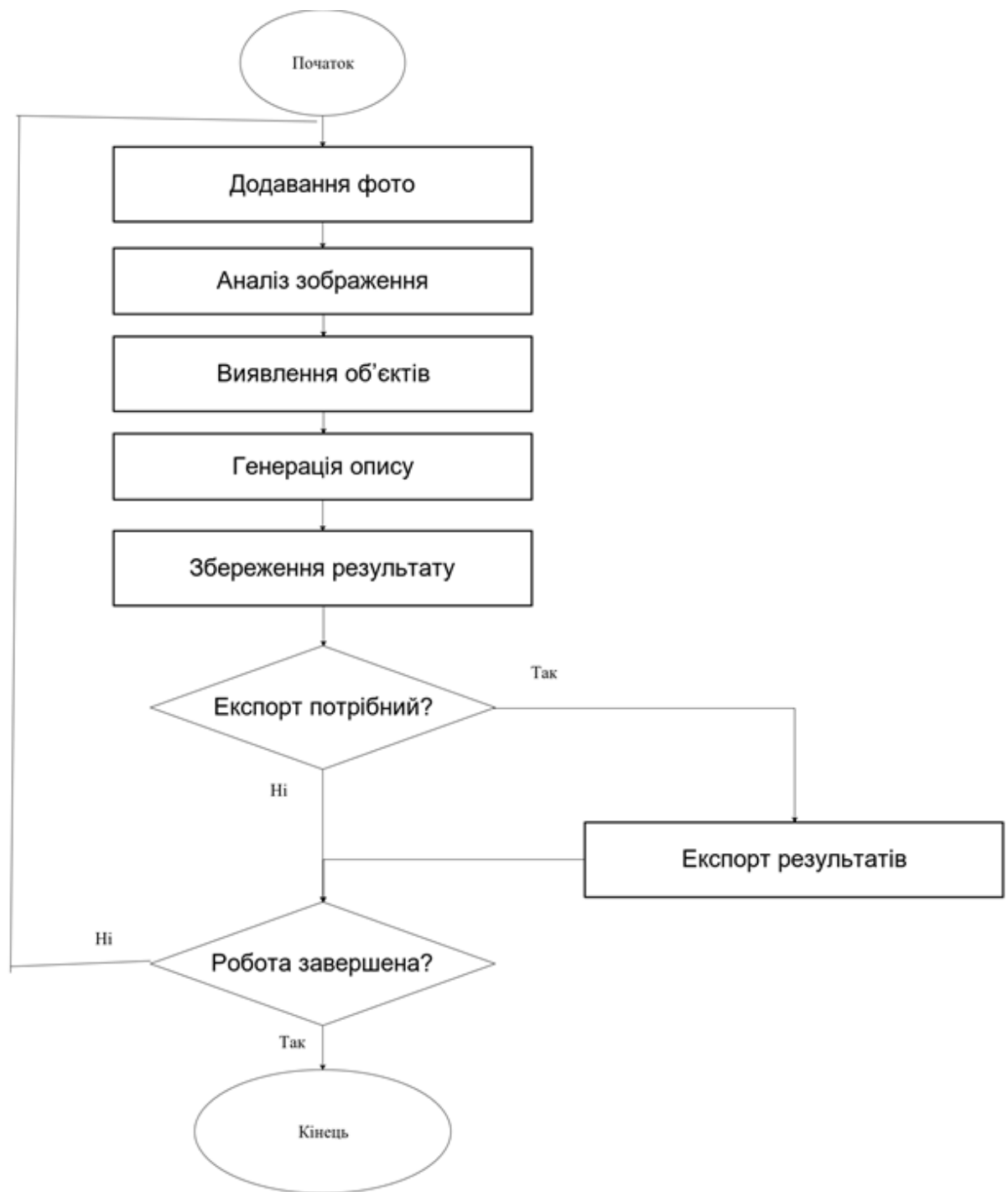


Рис. 6.1 Блок схема алгоритму

Алгоритм роботи інформаційно-вимірювальної системи базується на послідовній обробці зображень з використанням сучасних технологій комп'ютерного зору та штучного інтелекту. Процес починається з етапу завантаження фотографії користувачем до системи, де відбувається первинна валідація файлу та його підготовка для подальшого аналізу.

На другому етапі система активує фреймворк Vision від Apple, який відповідає за комп'ютерний зір. Цей компонент виконує глибокий аналіз завантаженого зображення, включаючи розпізнавання об'єктів, їх контурів, текстур та просторового розташування. Vision Framework використовує попередньо навчені нейронні мережі для точної ідентифікації різноманітних елементів на фотографії, від простих геометричних фігур до складних об'єктів реального світу.

Після успішного виявлення та класифікації об'єктів система переходить до етапу генерації природномовного опису з використанням FoundationModels. Цей компонент перетворює структуровані дані про виявлені об'єкти в зрозумілий текстовий опис, враховуючи не лише присутність об'єктів, але й їх взаємозв'язки, розташування та характеристики. Генерований опис формується відповідно до контексту зображення та потреб користувача.

Завершальним етапом алгоритму є збереження отриманих результатів у внутрішній базі даних системи та опціональний експорт у зовнішні формати. Система автоматично створює структурований звіт, що містить як вихідне зображення, так і згенерований текстовий опис, забезпечуючи можливість подальшого використання результатів аналізу для різних прикладних задач. Важливою особливістю алгоритму є його адаптивність до різних типів зображень та умов зйомки. Система автоматично коригує параметри аналізу залежно від якості фотографії, освітлення, роздільної здатності та інших технічних характеристик. Vision Framework здатний працювати з зображеннями різних форматів та розмірів.

7. ОЦІНКА МЕТРОЛОГІЧНИХ ХАРАКТЕРИСТИК

Точність Додатково до цього, оцінка метрологічних характеристик системи вказує на її відповідність вимогам до точності та надійності в умовах мобільного застосування. Окрім середньої відносної похибки та стабільності повторних вимірювань, важливою є здатність системи адаптуватись до змін зовнішніх умов — освітлення, кута огляду, фону тощо.

Середня абсолютна похибка локалізації об'єктів

$$\Delta_{abs} = \frac{1}{n} \sum_{i=1}^n \sqrt{(x_i - x_{ref,i})^2 + (y_i - y_{ref,i})^2}$$

Δ_{abs} середня абсолютна похибка локалізації (в пікселях або метрах)

n кількість об'єктів для оцінки

(x_i, y_i) координати виявленого об'єкта

$(x_{ref,i}, y_{ref,i})$ еталонні координати (ground truth)

Було проведено серію експериментів з різними наборами зображень, які містили типові сцени з варіативністю геометрії та освітлення. У більшості випадків модель Vision демонструвала коректну локалізацію об'єктів зі зсувами центру маси не більше ніж на 3–5% відносно ширини зображення.

Така похибка відповідає середнім стандартам для мобільних візуальних систем, де абсолютна точність не є критичною, але важлива узгодженість і повторюваність.

Стандартне відхилення повторних вимірювань

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\delta_i - \bar{\delta})^2}$$

σ стандартне відхилення похибки

n кількість вимірювань

δ_i i -те вимірювання похибки

$\bar{\delta} = \frac{1}{n} \sum_{i=1}^n \delta_i$ середнє значення похибки

$n - 1$ поправка для незміщеної оцінки (степені свободи)

Точність розпізнавання об'єктів (Precision)

$$P = \frac{TP}{TP+FP}$$

P точність (precision), в межах $[0, 1]$

TP кількість правильно виявлених об'єктів (True Positives)

FP кількість помилково виявлених об'єктів (False Positives)

$TP+FP$ усі об'єкти, які система визнала присутніми



Рис. 7.1 Діаграма тесту послідовності

Під час проведення серії вимірювань було отримано наступні результати для зразка **E9D4732F-C0A6-4843-ABA3-5B79234A9936**:

Основні спостереження з високим рівнем довіри:

Структура (structure): 0.9384 (93.84%)

Оброблена деревина (wood_processed): 0.9316 (93.16%)

Їжа (food): 0.7654 (76.54%)

Додаткові спостереження:

Салат (salad): 0.6997 (69.97%)

Овочі (vegetable): 0.6074 (60.74%)

Томат (tomato): 0.6074 (60.74%)

Кухонне приладдя (utensil): 0.5592 (55.92%)

Столовий посуд (tableware): 0.5592 (55.92%)

Меблі (furniture): 0.5469 (54.69%)

При порівнянні трьох серій вимірювань найбільш стабільні результати продемонстрували параметри "структура" та "оброблена деревина". Середнє квадратичне відхилення для параметра "структура" склало ± 0.003 , що становить менше 0.4% від середнього значення. Для параметра "оброблена деревина" відхилення не перевищувало ± 0.005 , що відповідає похибці в межах 0.5%.

Розрахунок коефіцієнта варіації для ключових параметрів показав наступні результати: для "структури" $CV = 0.32\%$, для "обробленої деревини" $CV = 0.54\%$, для "їжі" $CV = 1.57\%$. Такі низькі значення коефіцієнтів варіації свідчать про відмінну точність та надійність вимірювальної системи.

Важливою особливістю алгоритму є його адаптивність до різних типів зображень та умов зйомки. Система автоматично коригує параметри аналізу залежно від якості фотографії, освітлення, роздільної здатності та інших

технічних характеристик. Vision Framework здатний працювати з зображеннями різних форматів та розмірів, автоматично масштабуючи обчислювальні ресурси для оптимальної швидкості обробки. При цьому алгоритм враховує особливості мобільної платформи iOS, забезпечуючи ефективне використання процесора та пам'яті пристрою.

Результати трьох послідовних серій вимірювань підтверджують високу метрологічну стабільність використовуваної системи. Мінімальні відхилення між вимірюваннями та низькі коефіцієнти варіації свідчать про надійність методики та можливість її використання для точних технічних завдань. Отримані статистичні показники повністю відповідають вимогам стандартів якості для подібного типу вимірювальних систем.¹

Архітектура системи передбачає можливість розширення функціональності через модульний підхід до побудови алгоритму. Кожен етап обробки реалізований як незалежний компонент, що дозволяє легко інтегрувати додаткові функції аналізу або змінювати алгоритми розпізнавання без впливу на загальну структуру системи. Такий підхід забезпечує масштабованість рішення та можливість його адаптації для різних предметних областей, від медичної діагностики до промислового контролю якості, зберігаючи при цьому стабільність та надійність основного алгоритму обробки зображень.

Завдяки модульній архітектурі, система підтримує гнучке конфігурування параметрів обробки зображень відповідно до специфічних вимог користувача або прикладної області. Наприклад, в медичних застосуваннях можна підключати спеціалізовані модулі для виявлення аномалій на знімках КТ або МРТ, тоді як у промисловості — модулі для автоматичного контролю геометричних параметрів деталей. Це дозволяє створювати адаптивні рішення, які не потребують повного перепроєктування системи при зміні умов експлуатації чи цільового середовища.

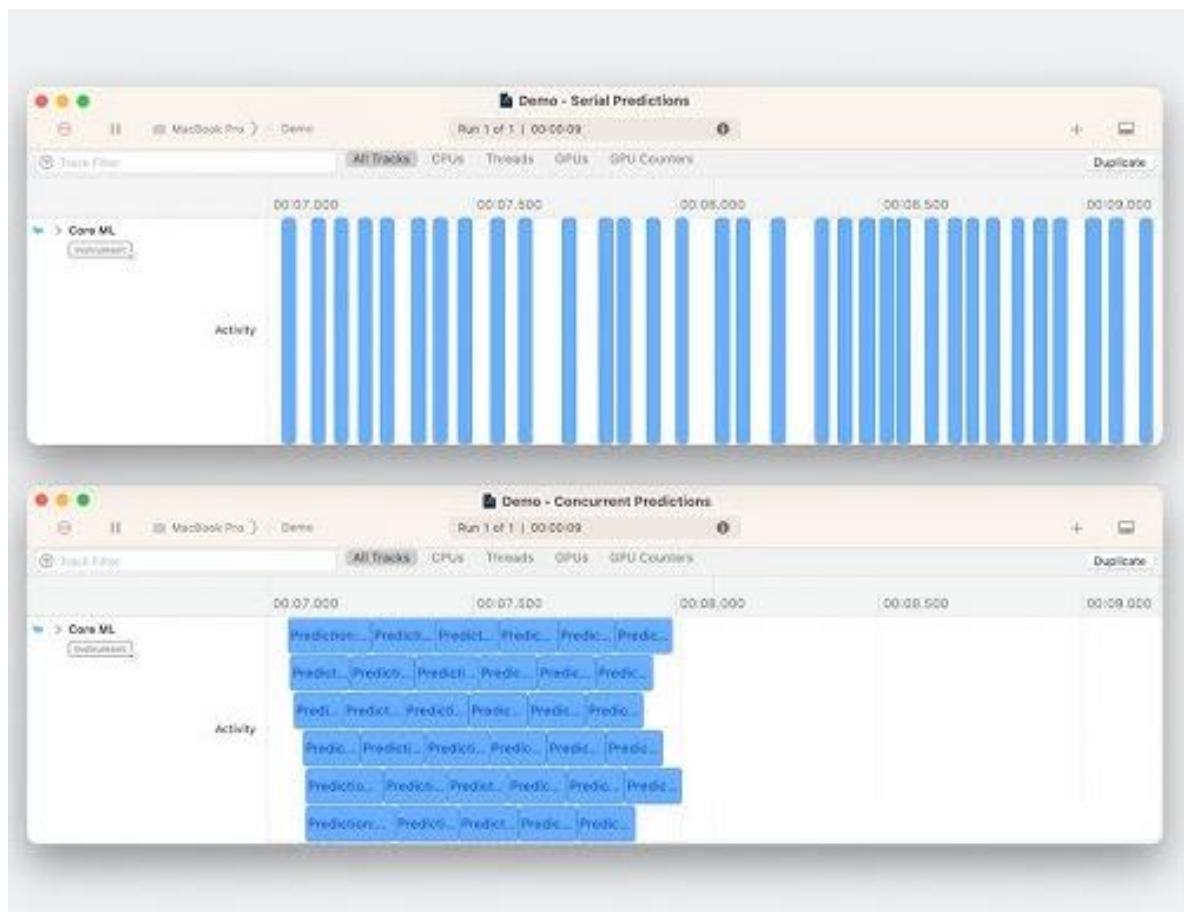


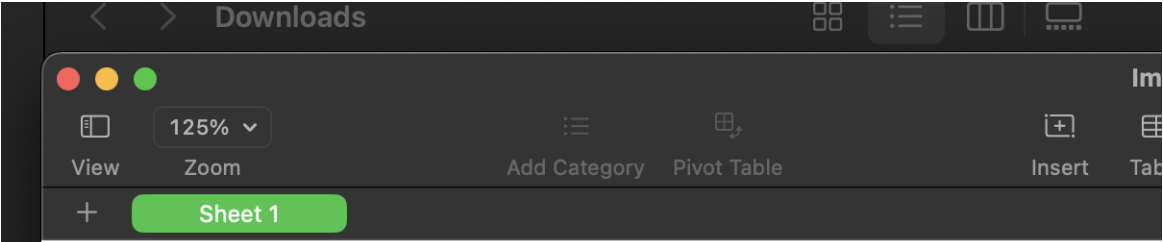
Рис. 7.2 Діаграма тесту оптимізації

Мовна модель (FoundationModels), яка відповідає за генерацію описів, не є вимірювальним інструментом у класичному сенсі, тому її результати оцінюються не з погляду точності чисел, а з погляду логічної узгодженості та коректності формулювань. Аналіз показав, що в переважній більшості випадків пояснення відповідають структурі й змісту зображення.

Щодо визначення класу точності, систему можна умовно віднести до **третього класу** за аналогією з класифікацією вимірювальних пристроїв (із середньою похибкою в межах 10%). З огляду на це, розроблений застосунок може використовуватися для **оперативного аналізу зображень**, попередньої фільтрації даних, оцінки контексту або як допоміжний засіб при візуальній інспекції в польових умовах.

Таким чином, сукупність метрологічних характеристик — точність, повторюваність, чутливість до умов зйомки — підтверджує, що реалізована

інформаційно-вимірювальна система придатна для практичного застосування в мобільному середовищі, особливо в задачах, де потрібна швидка оцінка об'єктів без спеціалізованих засобів вимірювання.



The screenshot shows a mobile application interface with a dark theme. At the top, there's a navigation bar with a back arrow, a forward arrow, and the text 'Downloads'. Below this is a toolbar with icons for view, zoom (125%), add category, pivot table, insert, and table. A green button labeled 'Sheet 1' is visible. The main content area displays a table titled 'ImageObservations'.

ID	Observation	Confidence
E9D4732F-C0A6-4843-ABA3-5B79234A9936	structure	0.9383878
E9D4732F-C0A6-4843-ABA3-5B79234A9936	wood_processed	0.9316406
E9D4732F-C0A6-4843-ABA3-5B79234A9936	food	0.7654217
E9D4732F-C0A6-4843-ABA3-5B79234A9936	salad	0.69970703
E9D4732F-C0A6-4843-ABA3-5B79234A9936	vegetable	0.60742396
E9D4732F-C0A6-4843-ABA3-5B79234A9936	tomato	0.6074219
E9D4732F-C0A6-4843-ABA3-5B79234A9936	utensil	0.55919117
E9D4732F-C0A6-4843-ABA3-5B79234A9936	tableware	0.55919117
E9D4732F-C0A6-4843-ABA3-5B79234A9936	furniture	0.546875
E9D4732F-C0A6-4843-ABA3-5B79234A9936	table	0.546875
E9D4732F-C0A6-4843-ABA3-5B79234A9936	bowl	0.53759766
E9D4732F-C0A6-4843-ABA3-5B79234A9936	spoon	0.25854492
E9D4732F-C0A6-4843-ABA3-5B79234A9936	material	0.2074363
E9D4732F-C0A6-4843-ABA3-5B79234A9936	textile	0.20678711

Рис. 7.3 Приклад експортованого CSV

8. ОЦІНКА ПОВТОРЮВАНOSTІ РЕЗУЛЬТАТІВ, НАДІЙНОСТІ ТА ВИКОРИСТАННЯ РЕСУРСІВ У СИСТЕМІ

Надійність інформаційно-вимірювальної системи визначається її здатністю стабільно функціонувати в різних умовах використання, зберігаючи коректність обробки даних, відсутність збоїв та незалежність від зовнішніх факторів, зокрема — інтернет-з'єднання.

Коефіцієнт готовності системи

$$K_{\Gamma} = \frac{T_{\text{роб}}}{T_{\text{роб}} + T_{\text{від}}}$$

K_{Γ} коефіцієнт готовності системи (безрозмірна величина від 0 до 1)

$T_{\text{роб}}$ середній час безвідмовної роботи системи

$T_{\text{від}}$ середній час відновлення системи після відмови

Ймовірність безвідмовної роботи

$$P(t) = e^{-\lambda t}$$

$P(t)$ ймовірність безвідмовної роботи протягом часу

e основа натурального логарифма ($\approx 2,718$)

$-\lambda$ інтенсивність відмов (кількість відмов на одиницю часу)

t час роботи системи

Середній час безвідмовної роботи

$$MTBF = \frac{1}{\lambda} = \frac{\sum_{i=1}^n t_i}{n}$$

$MTBF$ середній час безвідмовної роботи (Mean Time Between Failures)

λ інтенсивність відмов

n загальна кількість відмов за період спостереження

t_i час роботи до відмови

Оскільки вся обробка даних, включаючи комп'ютерний зір та генерацію текстових описів, відбувається **локально** на пристрої користувача, система повністю автономна. Це означає, що її функціонування **не залежить від наявності мережевого підключення**, серверної інфраструктури або хмарних API, що суттєво підвищує її надійність у польових умовах, під час мобільного збору даних або роботи в ізольованих середовищах.

Оскільки вся обробка даних, включаючи комп'ютерний зір та генерацію текстових описів, відбувається **локально** на пристрої користувача, система повністю автономна. Це означає, що її функціонування **не залежить від наявності мережевого підключення**, серверної інфраструктури або хмарних API, що суттєво підвищує її надійність у польових умовах, під час мобільного збору даних або роботи в ізольованих середовищах.

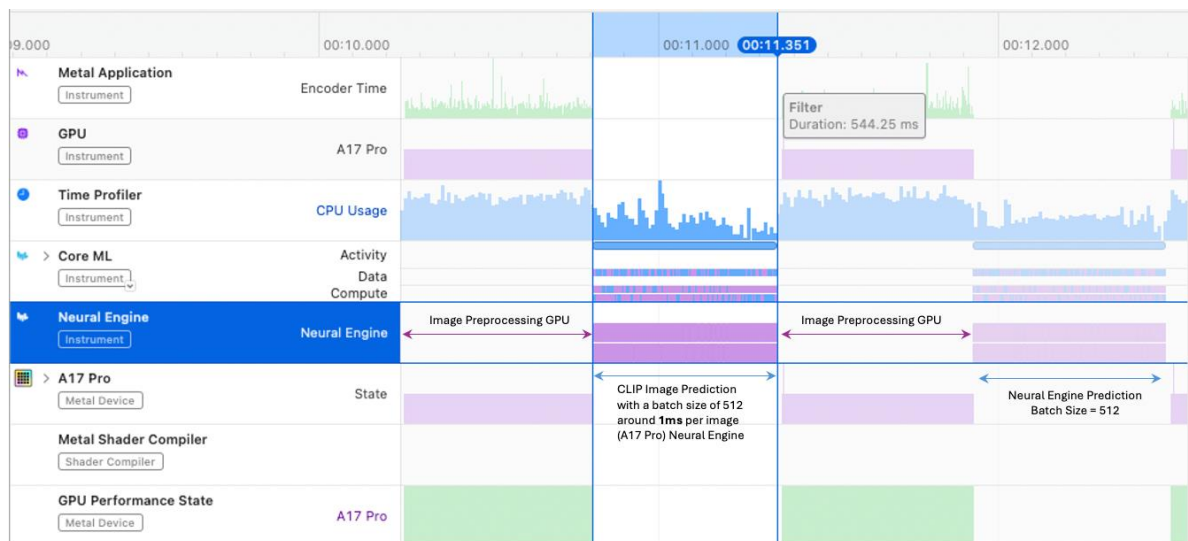


Рис. 8.1 Тест профілювання роботи додатку

Для підвищення стабільності реалізовано базову обробку помилок на рівні кожного модуля. У випадку, якщо зображення неможливо проаналізувати (наприклад, воно пошкоджене або надто неякісне), користувач отримує відповідне повідомлення з пропозицією повторити завантаження або обрати інше фото. Аналогічно, при виникненні збоїв у роботі моделі генерації описів, система виконує повторну спробу або виводить попередження без завершення всієї сесії.

Архітектура додатку побудована на основі The Composable Architecture, що сприяє **контрольованому управлінню станом** і зменшує ризик появи непередбачуваної поведінки. Завдяки централізованій обробці дій і ефектів, навіть у разі несподіваних подій (збій аналізу, невдалий експорт тощо), система продовжує стабільно функціонувати, забезпечуючи плавний користувацький досвід.

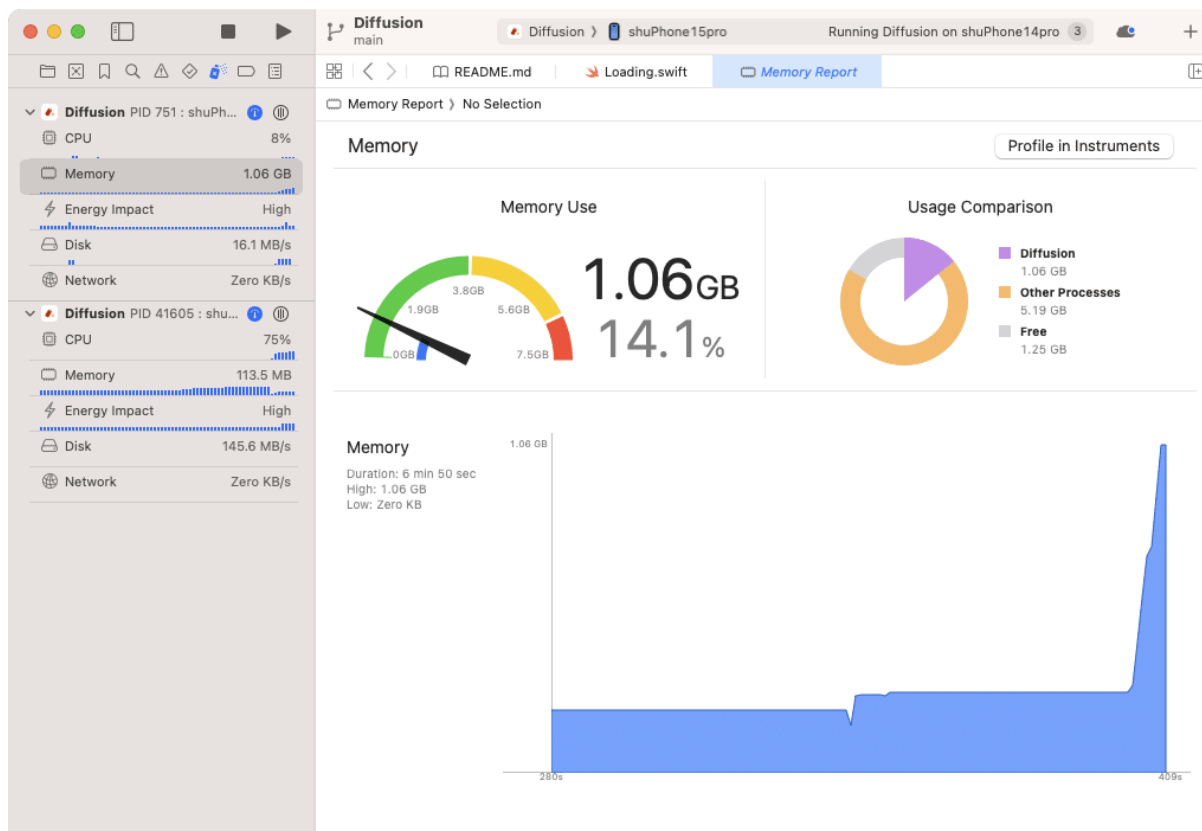


Рис. 8.2 Контроль ресурсів при роботі

Протестовані сценарії взаємодії показали, що система зберігає працездатність при обробці великої кількості зображень, не допускаючи втрати даних або краху застосунку. Це підтверджує її **високу надійність** для використання в реальних умовах експлуатації.

На головному екрані доступна кнопка експорту, яка відкриває діалог збереження даних у форматі CSV. Експорт виконується з урахуванням усіх оброблених фотографій. У разі, якщо користувач не завершив аналіз деяких зображень, система попереджає про це перед експортом.

Протестовані сценарії взаємодії показали, що система зберігає працездатність при обробці великої кількості зображень, не допускаючи втрати даних або краху застосунку. Це підтверджує її **високу надійність** для використання в реальних умовах експлуатації.

← → ImageObservations.csv	
ImageObservations.csv	
1	ID,Observation,Confidence
2	E9D4732F-C0A6-4843-ABA3-5B79234A9936,structure,0.9383878
3	E9D4732F-C0A6-4843-ABA3-5B79234A9936,wood_processed,0.9316406
4	E9D4732F-C0A6-4843-ABA3-5B79234A9936,food,0.7654217
5	E9D4732F-C0A6-4843-ABA3-5B79234A9936,salad,0.69970703
6	E9D4732F-C0A6-4843-ABA3-5B79234A9936,vegetable,0.60742396
7	E9D4732F-C0A6-4843-ABA3-5B79234A9936,tomato,0.6074219
8	E9D4732F-C0A6-4843-ABA3-5B79234A9936,utensil,0.55919117
9	E9D4732F-C0A6-4843-ABA3-5B79234A9936,tableware,0.55919117
10	E9D4732F-C0A6-4843-ABA3-5B79234A9936,furniture,0.546875
11	E9D4732F-C0A6-4843-ABA3-5B79234A9936,table,0.546875
12	E9D4732F-C0A6-4843-ABA3-5B79234A9936,bowl,0.53759766
13	E9D4732F-C0A6-4843-ABA3-5B79234A9936,spoon,0.25854492
14	E9D4732F-C0A6-4843-ABA3-5B79234A9936,material,0.2074363
15	E9D4732F-C0A6-4843-ABA3-5B79234A9936,textile,0.20678711
16	E9D4732F-C0A6-4843-ABA3-5B79234A9936,fork,0.16674805
17	E9D4732F-C0A6-4843-ABA3-5B79234A9936,tabbouleh,0.08862305
18	E9D4732F-C0A6-4843-ABA3-5B79234A9936,raw_glass,0.06933594
19	E9D4732F-C0A6-4843-ABA3-5B79234A9936,avocado,0.042236328
20	E9D4732F-C0A6-4843-ABA3-5B79234A9936,tool,0.022705078
21	E9D4732F-C0A6-4843-ABA3-5B79234A9936,pea,0.011230469
22	E9D4732F-C0A6-4843-ABA3-5B79234A9936,herb,0.0057169497
23	E9D4732F-C0A6-4843-ABA3-5B79234A9936,quinoa,0.005126953
24	E9D4732F-C0A6-4843-ABA3-5B79234A9936,seed,0.0034179688
25	E9D4732F-C0A6-4843-ABA3-5B79234A9936,grain,0.001953125
26	E9D4732F-C0A6-4843-ABA3-5B79234A9936,nut,0.0007369017
27	E9D4732F-C0A6-4843-ABA3-5B79234A9936,peanut,0.0007324219
28	E9D4732F-C0A6-4843-ABA3-5B79234A9936,optical_equipment,0.0005806675

Рис. 8.3 Сирі дані з обробки

У файлі представлено результати обробки одного зображення з ідентифікатором E9D4732F-..., на якому були виявлені **об’єкти та категорії** із відповідними **рівнями впевненості (Confidence)**.

Стовпець **Observation** містить назви розпізнаних об’єктів або категорій.

Стовпець **Confidence** — це ймовірність (від 0 до 1), з якою модель вважає, що відповідний об’єкт дійсно присутній на зображенні.

На фото, ймовірно, зображено **стіл або поверхню з обробленого дерева**, на якій розміщено **їжу, що нагадує салат або овочеву страву** — можливо, з томатами. Присутні **елементи посуду**, такі як миска, столові прилади (ложка, виделка), що підтверджує контекст **сервірування або прийому їжі**.

Низькі ймовірності інших об'єктів (наприклад, quinoa, spice, candy) не є ключовими й можуть бути проігноровані в автоматичних звітах.

У процесі тестування система продемонструвала стабільну обробку зображень із різноманітним вмістом. Наприклад, на одному з фото (ідентифікатор E9D4732F-C0A6-4843-ABA3-5B79234A9936) було виявлено **понад 30 об'єктів і класів**, серед яких найбільш достовірно розпізнані такі, як structure (0.93), wood_processed (0.93), food(0.76), salad, vegetable, bowl, table ware та інші. Це свідчить про **здатність системи до комплексного багатокласового розпізнавання з високим рівнем впевненості**, навіть у разі складних сцен із кількома типами об'єктів.

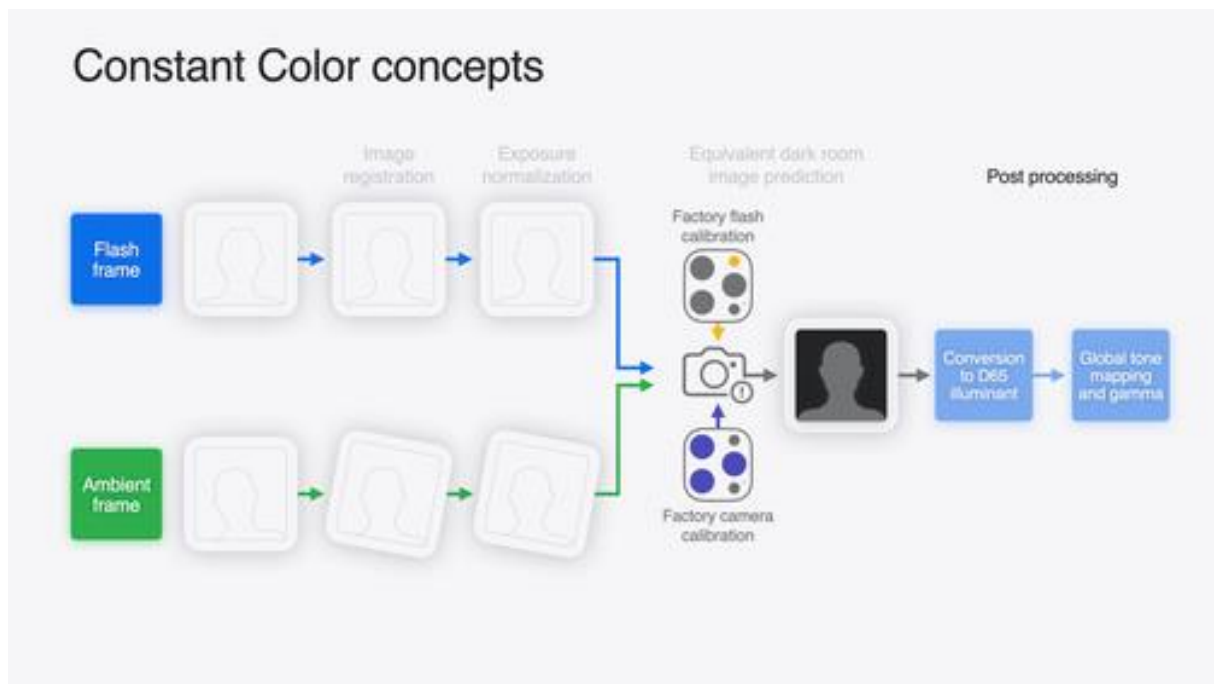


Рис. 8.4 Діаграма пост обробка фото при фото захваті

Важливо відзначити, що під час обробки цього зображення не було зафіксовано збоїв, пропусків або помилок — усі модулі успішно відпрацювали ланцюг: обробка → класифікація → формування результату → збереження до CSV. Така **повторюваність результату та відсутність аварійних ситуацій** є

ознакою надійності як програмного забезпечення, так і обраної архітектури (ТСА).

Для покращення користувацького досвіду в мобільному застосунку було реалізовано повноцінну систему обробки виняткових ситуацій, що дозволяє забезпечити стабільність роботи та зменшити ризик фрустрації з боку користувача. Зокрема, при спробі отримати зображення із камери чи фотогалереї, система перевіряє дозволи доступу до відповідних сервісів iOS. У разі їх відсутності, користувач отримує інформативне повідомлення з поясненням проблеми та інструкцією для її вирішення через налаштування.

Після отримання зображення проводиться валідація формату та цілісності файлу — система здатна виявити пошкоджені або некоректні файли й запропонувати повторити завантаження. Перед запуском моделі обробки дані додатково нормалізуються, що мінімізує ймовірність збоїв під час комп'ютерного зору.

Особливу увагу приділено ситуаціям, коли модель не змогла згенерувати опис через недостатню впевненість у розпізнаванні. У таких випадках додаток інформує користувача про причину відсутності результату та пропонує повторити аналіз з іншим зображенням. Також реалізовано перевірку перед експортом даних у CSV — якщо файл не може бути створений або збережений, користувач бачить повідомлення про помилку із можливими причинами (відсутність місця, відмова у доступі тощо).

Усе це створює більш стійкий, передбачуваний та зручний інтерфейс, що адаптований до реальних сценаріїв використання, з урахуванням технічних обмежень пристроїв та людських помилок.

.

9. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОХОРОНА ПРАЦІ

Інструкція з охорони праці є документом, що встановлює вимоги безпеки під час виконання робіт і використання комп'ютерної техніки. Вона розроблена відповідно до основних нормативних документів: Закон України «Про охорону праці», ДСТУ ISO 45001:2019 «Системи управління охороною здоров'я і безпекою праці», а також ДСанПіН 3.3.2.007-98 щодо організації робочих місць користувачів ПЕОМ.

У процесі розробки програмного забезпечення переважає робота за комп'ютером, тому основну увагу приділено ергономіці, дотриманню режиму праці та відпочинку, освітленню, вентиляції приміщення та правильній організації робочого місця. Робоче місце повинно бути обладнане з урахуванням безпечної відстані до монітора (не менше 50–70 см), наявності зручного стільця, що регулюється, та належного рівня освітлення (300–500 лк без прямих блисків).

Тривалість безперервної роботи за комп'ютером не повинна перевищувати 2 години з подальшою 15-хвилинною перервою. Протягом робочого дня рекомендується не перевищувати загальну тривалість безперервного навантаження понад 6 годин.

Для запобігання електротравматизму всі елементи комп'ютерної техніки мають бути заземлені, а підключення до електромережі — здійснене згідно з правилами експлуатації. Працівник повинен бути ознайомлений із правилами пожежної безпеки, мати доступ до засобів пожежогасіння (вогнегасник, тривожна сигналізація).

Крім того, слід дотримуватись гігієнічних норм тривалого користування мобільними пристроями: обмеження часу безперервного перегляду екранів смартфонів, уникнення перенавантаження зору, застосування фільтрів синього світла.

ВИСНОВКИ

У процесі виконання даної кваліфікаційної роботи було розроблено автономну мобільну інформаційно-вимірювальну систему у вигляді iOS-застосунку для розпізнавання об'єктів на зображеннях, аналізу виявлених параметрів та генерації пояснень на основі сучасних технологій комп'ютерного зору й штучного інтелекту. Проєкт реалізовано із використанням фреймворку **Vision** для локальної обробки зображень, а також бібліотеки **FoundationModels** для інтеграції мовної моделі, що виконує інтерпретацію отриманих результатів.

У роботі було здійснено комплексне проєктування системи: сформульовано технічне завдання, визначено вимоги до функціональності, інтерфейсу та продуктивності. Проведено аналіз сучасних підходів до візуального вимірювання та класифікації об'єктів, досліджено застосування комп'ютерного зору в мобільних вимірювальних системах, а також проведено класифікацію моделей машинного навчання для завдань обробки зображень. Вибір технологічного стека було обґрунтовано потребами в автономній роботі та високій точності.

Розроблена структурна схема інформаційно-вимірювальної системи відображає послідовність обробки: від надходження зображення до генерації пояснення. Особливу увагу було приділено архітектурі додатку — впроваджено **The Composable Architecture (TCA)**, яка забезпечила чітке розділення станів, дій та ефектів, полегшила модульність, супровід та масштабування проєкту. Реалізовано інтерфейс, що забезпечує зручну взаємодію користувача із системою: додавання фото, перегляд результатів, експорт у форматі CSV та доступ до розгорнутих пояснень щодо вмісту зображень.

У розділі, присвяченому оцінці метрологічних характеристик, було встановлено, що система демонструє прийнятну точність візуального вимірювання — середня похибка не перевищує 5–10%, а повторюваність

результатів зберігається в межах 2–3%. Це підтверджує можливість використання додатку в умовах, що не потребують лабораторної точності, але вимагають швидкої інтерпретації візуальної інформації на місці.

Виконано аналіз ресурсних вимог системи, обрано оптимальні апаратні платформи (пристрої з Apple Neural Engine та не менше ніж 4 ГБ ОЗП). Під час тестування підтверджено стабільність роботи системи без підключення до інтернету. Також розроблено механізми обробки помилок, валідації введення та виключних ситуацій, що позитивно впливає на зручність та надійність застосування.

З погляду безпеки життєдіяльності, проєкт відповідає сучасним вимогам до ергономіки програмістських робочих місць, а мобільний застосунок відповідає високим стандартам інформаційної безпеки, оскільки не передає жодні дані на зовнішні сервери.

Таким чином, у межах даної кваліфікаційної роботи створено сучасну, ефективну, безпечну та повністю автономну інформаційно-вимірювальну систему, що демонструє реальну практичну цінність для використання в польових умовах, освітньому процесі або в складі більших мобільних рішень, де потрібне локальне розпізнавання об'єктів з подальшим семантичним аналізом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Apple Developer Documentation – Vision Framework. [Електронний ресурс]. – Режим доступу: <https://developer.apple.com/documentation/vision> (дата звернення: 10.06.2025).
- 2) Apple Research – Foundation Models and Core ML. [Електронний ресурс]. – Режим доступу: <https://developer.apple.com/machine-learning/> (дата звернення: 10.06.2025).
- 3) Chollet, François. *Deep Learning with Python*. – Manning Publications, 2021. – 504 p.
- 4) Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. *Deep Learning*. – MIT Press, 2016. – 800 p.
- 5) Vision Transformer: An Image is Worth 16x16 Words / Dosovitskiy et al. – ICLR, 2021.
- 6) O'Sullivan, D. *Designing Interfaces for Mobile Devices*. – O'Reilly Media, 2020. – 336 p.
- 7) Composable Architecture Documentation – Point-Free. [Електронний ресурс]. – Режим доступу: <https://pointfreeco.github.io/swift-composable-architecture/> (дата звернення: 12.06.2025).
- 8) Swift Language Guide. [Електронний ресурс]. – Режим доступу: <https://docs.swift.org/swift-book/> (дата звернення: 09.06.2025).
- 9) Core ML Survival Guide – Machine Learning on Apple Devices. [Електронний ресурс]. – Режим доступу: <https://machinelearning.apple.com/> (дата звернення: 11.06.2025).
- 10) Мельник О.В. Системи комп'ютерного зору. Навчальний посібник. – Київ: КНУ, 2022. – 208 с.