

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра комп'ютерних наук

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Інформаційна система для мерчандайзингу з використанням
технологій комп'ютерного зору

Виконав: студент
спеціальності

VI курсу, групи САМ-61
124 Системний аналіз
(шифр і назва спеціальності)

Керівник

Нормоконтроль

Завідувач кафедри

Рецензент

Чорновол О.В.

(підпис)

(прізвище та ініціали)

Фриз М.Є.

(підпис)

(прізвище та ініціали)

Никитюк В.В.

(підпис)

(прізвище та ініціали)

Боднарчук І.О.

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«___» грудня 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 124 Системний аналіз
(шифр і назва спеціальності)

Студенту Чорноволу Олександрову Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна система для мерчендайзингу з використанням технологій комп'ютерного зору

Керівник роботи Фриз Михайло Євгенович, к.т.н., доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «29» листопада 2024 року № 4/7-1138

2. Термін подання студентом завершеної роботи 17 грудня 2024р.

3. Вихідні дані до роботи Наукові публікації про методи комп'ютерного зору, засоби розробки застосунків з клієнт-серверною архітектурою та використання хмарних обчислювальних платформ.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1 Огляд методів комп'ютерного зору для розпізнавання об'єктів та їх застосування в мерчендайзингу. 2 Обґрунтування вибору архітектури та технологій для розробки інформаційної системи. 3 Реалізація та тестування інформаційної системи для мерчендайзингу. 4 Охорона праці та безпека в надзвичайних ситуаціях. Висновки. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1 Титульна сторінка. 2 Об'єкт та предмет дослідження. 3 Актуальність теми.

4 Мета та задачі дослідження. 5 Наукова новизна та практичне значення. 6 Архітектура системи. 7 Технології розробки. 8 Тестування системи. 9 Результати тестування продуктивності. 10 Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сенчишин В.С., доцент		
Безпека в надзвичайних ситуаціях	Клепчик В.М., ст. викладач		

7. Дата видачі завдання 29 листопада 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	29.11.2024	Виконано
2.	Підбір наукових джерел про методи комп'ютерного зору для розпізнавання та підрахунку об'єктів та архітектуру систем розподіленого обчислення.	29.11.2024-04.12.2024	Виконано
3.	Опрацювання наукових публікацій та збір даних про методи комп'ютерного зору.	29.11.2024-04.12.2024	Виконано
4.	Виконання дослідження методів комп'ютерного зору та систем розподіленого обчислення.	05.12.2024-09.12.2024	Виконано
5.	Оформлення розділу «Огляд методів комп'ютерного зору для розпізнавання об'єктів та їх застосування в мерчендайзингу»	05.12.2024-09.12.2024	Виконано
6.	Оформлення розділу «Обґрунтування вибору архітектури та технологій для розробки інформаційної системи для мерчендайзингу.»	05.12.2024-09.12.2024	Виконано
7.	Оформлення розділу «Реалізація та тестування системи автоматизації мерчендайзингу з використанням комп'ютерного зору.»	05.12.2024-09.12.2024	Виконано
8.	Виконання завдання до підрозділу «Охорона праці»	02.12.2024-09.12.2024	Виконано
9.	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»	02.12.2024-09.12.2024	Виконано
10.	Оформлення кваліфікаційної роботи	10.12.2024-11.12.2024	Виконано
11.	Нормоконтроль	12.12.2024-13.12.2024	Виконано
12.	Перевірка на плагіат	16.12.2024	Виконано
13.	Попередній захист кваліфікаційної роботи	17.12.2024	Виконано
14.	Захист кваліфікаційної роботи	23.12.2024	

Студент

(підпис)

Чорновол О.В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Фриз М.Є.

(прізвище та ініціали)

АНОТАЦІЯ

Інформаційна система для мерчендайзингу з використанням технологій комп'ютерного зору // Кваліфікаційна робота освітнього рівня «Магістр» // Чорновол Олександр Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група САМ-61 // Тернопіль, 2024 // С. 66, рис. – 9, табл. – 1, кресл. – 11, додат. – 3, бібліогр. – 50.

Ключові слова: мерчендайзинг, комп'ютерний зір, розпізнавання об'єктів, мобільний застосунок, автоматизація, хмарні технології, обробка зображень.

Кваліфікаційна робота присвячена розробці інформаційної системи для автоматизації мерчендайзингу з використанням технологій комп'ютерного зору. Система дозволяє мерчендайзерам за допомогою мобільного застосунку фотографувати полиці з товарами та автоматично визначати кількість товарів на зображенні. Це сприяє контролю якості представлення товарів, спрощенню звітності та порівнянню з конкурентами. В роботі проведено аналіз існуючих методів комп'ютерного зору для розпізнавання об'єктів та обґрунтовано вибір архітектури системи, що базується на моделі YOLOv5 та фреймворку Flask. Детально описано процес розробки мобільного застосунку для iOS та серверної частини, що розміщена на хмарній платформі AWS EC2. Проведено тестування розробленої системи та оцінено її ефективність. Результати дослідження показали, що запропонована система дозволяє автоматизувати процес підрахунку товарів, підвищити точність даних та скоротити час, необхідний для проведення мерчендайзингових заходів. Перспективи подальшого розвитку системи включають розпізнавання брендів, аналіз цінників та інтеграцію з системами управління запасами.

ANNOTATION

Information System for Merchandising Using Computer Vision Technologies // The educational level "Master" qualification work // Chornovol Oleksandr Volodymyrovych // Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SAm-61 group // Ternopil, 2024 // P. 66, fig. - 9, tables - 1, posters - 11, annexes - 3, ref. - 50.

Key words: merchandising, computer vision, object recognition, mobile application, automation, cloud technologies, image processing.

This thesis is devoted to the development of an information system for automating merchandising using computer vision technologies. The system allows merchandisers to take pictures of shelves with goods using a mobile application and automatically determine the number of goods in the image. This helps to control the quality of product presentation, simplify reporting, and compare with competitors. The paper analyzes existing computer vision methods for object recognition and justifies the choice of system architecture based on the YOLOv5 model and the Flask framework. The process of developing a mobile application for iOS and the server part, which is hosted on the AWS EC2 cloud platform, is described in detail. The developed system was tested and its effectiveness was evaluated. The results of the study showed that the proposed system allows to automate the process of counting goods, improve data accuracy and reduce the time required for merchandising activities. Prospects for further development of the system include brand recognition, price tag analysis, and integration with inventory management systems.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AI (англ. Artificial Intelligence) – Штучний інтелект.

API (англ. Application Programming Interface) – прикладний програмний інтерфейс.

ML (англ. Machine Learning) – Машинне навчання.

SKU (англ. Stock Keeping Unit) – Ідентифікатор товарної позиції.

UI (англ. User Interface) – користувацький інтерфейс

ЗМІСТ

ВСТУП.....	7
1 ОГЛЯД МЕТОДІВ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ТА ЇХ ЗАСТОСУВАННЯ В МЕРЧЕНДАЙЗИНГУ.....	10
1.1 Основні задачі комп'ютерного зору в мерчендайзингу.....	11
1.2 Методи розпізнавання об'єктів.....	13
1.3 Застосування комп'ютерного зору в мерчендайзингу.....	15
1.4 Вибір оптимального методу розпізнавання об'єктів.....	17
1.4 Висновки до першого розділу.....	19
2 ОБГРУНТУВАННЯ ВИБОРУ АРХІТЕКТУРИ ТА ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	22
2.1 Вимоги до системи.....	24
2.2 Опис архітектури системи.....	25
2.3 Обґрунтування вибору технологій.....	26
2.3.1 Технології для мобільного застосунку.....	28
2.3.2 Технології для серверної частини.....	31
2.3.3 Технології для комп'ютерного зору.....	35
2.4 Висновки до другого розділу.....	37
3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ МЕРЧАНДАЙЗИНГУ.....	39
3.1 Розробка мобільного застосунку.....	41
3.2 Розробка серверної частини.....	48
3.3 Тестування системи.....	50
3.4 Висновки до третього розділу.....	52
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	54
4.1 Охорона праці.....	54
4.1.1 Шкідливі та небезпечні фактори при користуванні персональними комп'ютерами та смартфонами.....	54
4.2 Безпека в надзвичайних ситуаціях.....	56
4.2.1 Підвищення стійкості роботи об'єктів господарської діяльності у воєнний час.....	56
4.3 Висновки до четвертого розділу.....	59
ВИСНОВКИ.....	60
СПИСОК ДЖЕРЕЛ.....	62
ДОДАТКИ	

ВСТУП

Актуальність теми. Актуальність теми обумовлена зростаючою потребою бізнесу в автоматизації процесів мерчендайзингу для підвищення ефективності та конкурентоспроможності [1]. Традиційні методи мерчендайзингу, що базуються на ручному зборі даних, є трудомісткими, повільними та схильними до помилок. Використання технологій комп'ютерного зору дозволяє автоматизувати процес підрахунку товарів, контролю їх наявності та розташування на полицях, що значно спрощує роботу мерчендайзерів та підвищує точність даних. Це дає можливість оперативно реагувати на зміни попиту, оптимізувати викладку товарів та підвищувати продажі. Враховуючи стрімкий розвиток мобільних технологій та хмарних сервісів, створення мобільного застосунку з використанням комп'ютерного зору є перспективним напрямком для автоматизації мерчендайзингу та отримання конкурентних переваг на ринку [2]. Розробка таких систем актуальна для різних галузей роздрібної торгівлі та має значний потенціал для подальшого розвитку та вдосконалення.

Мета і задачі дослідження. Розробка інформаційної системи для автоматизації процесів мерчендайзингу з використанням технологій комп'ютерного зору, що дозволить підвищити ефективність роботи мерчендайзерів та якість збору даних про товари на полицях магазинів. Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- Провести аналіз існуючих рішень та методів комп'ютерного зору для розпізнавання та підрахунку об'єктів на зображеннях.
- Обґрунтувати вибір архітектури та технологічного стеку для розробки системи, включаючи вибір моделі комп'ютерного зору (YOLOv5), фреймворку для серверної частини (Flask) та хмарного провайдера (AWS EC2).
- Розробити мобільний застосунок для iOS, що дозволяє робити фотографії полиць з товарами та надсилати їх на сервер для обробки.
- Розробити серверну частину, що отримує зображення, обробляє їх за допомогою моделі YOLOv5 та повертає кількість розпізнаних товарів.

- Провести тестування розробленої системи, включаючи оцінку точності розпізнавання, швидкодії та стабільності роботи.
- Визначити перспективи подальшого розвитку та вдосконалення системи, враховуючи можливості розпізнавання брендів, аналізу цінників та інтеграції з іншими системами.
- Оцінити вплив розробленої системи на процес мерчендайзингу та ефективність роботи мерчендайзерів.

Об'єкт дослідження. Об'єктом дослідження є процес мерчендайзингу в роздрібній торгівлі та можливості його автоматизації.

Предмет дослідження. Предметом дослідження є інформаційна система для автоматизації мерчендайзингу з використанням технологій комп'ютерного зору, зокрема методи та алгоритми розпізнавання та підрахунку товарів на зображеннях полиць магазинів, архітектура системи, що включає мобільний застосунок та серверну частину, а також вплив впровадження цієї системи на ефективність процесу мерчендайзингу.

Наукова новизна одержаних результатів. Наукова новизна одержаних результатів полягає в розробці та практичній реалізації комплексної інформаційної системи для автоматизації мерчендайзингу, що інтегрує мобільний застосунок, серверну частину на базі YOLOv5 та хмарну платформу AWS EC2 [4]. Новизна проявляється в адаптації та оптимізації алгоритмів комп'ютерного зору для специфічних умов зйомки товарів на полицях магазинів, а також в розробці ефективного механізму взаємодії між мобільним застосунком та серверною частиною для забезпечення швидкої та точної обробки даних. Запропонований підхід дозволяє підвищити ефективність роботи мерчендайзерів, мінімізувати вплив людського фактору та отримати більш достовірну інформацію про стан товарів на полицях.

Практичне значення одержаних результатів. Практичне значення одержаних результатів полягає в створенні готового до впровадження програмного продукту, що дозволяє автоматизувати ключові процеси мерчендайзингу. Розроблений мобільний застосунок значно спрощує роботу мерчендайзерів, зменшуючи час, витрачений на збір даних, та мінімізуючи помилки. Автоматичний підрахунок товарів на полицях дозволяє оперативно

отримувати актуальну інформацію про наявність продукції, що сприяє оптимізації логістичних процесів та управління запасами. Система також надає можливості для аналізу ефективності мерчендайзингових заходів та прийняття обґрунтованих рішень щодо викладки товарів. Впровадження розробленої системи може призвести до збільшення продажів, зниження витрат та підвищення конкурентоспроможності підприємств роздрібно́ї торгівлі [5].

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на XII науково-технічній конференції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя (м. Тернопіль, 2024 р.).

Публікації. Основні результати кваліфікаційної роботи опубліковано у двох працях конференції (Див. додатки А).

Структура й обсяг кваліфікаційної роботи. Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, списку літератури з 50 найменувань та 3 додатків. Загальний обсяг кваліфікаційної роботи складає 66 сторінки, з них 50 сторінки основного тексту, який містить 9 рисунків та 1 таблицю.

1 ОГЛЯД МЕТОДІВ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ТА ЇХ ЗАСТОСУВАННЯ В МЕРЧЕНДАЙЗИНГУ

Комп'ютерний зір – галузь штучного інтелекту, що надає комп'ютерам здатність “бачити” та інтерпретувати зображення, стрімко трансформує різноманітні сфери діяльності, включаючи роздрібну торгівлю та, зокрема, мерчендайзинг [3]. У сучасному конкурентному середовищі ефективний мерчендайзинг є критично важливим для залучення покупців, стимулювання продажів та максимізації прибутку. Традиційні методи мерчендайзингу, що часто покладаються на ручну працю, візуальні перевірки та суб'єктивні оцінки, стають все менш ефективними в умовах зростаючої складності ринку, динамічних змін у поведінці споживачів та величезних обсягів даних. Ручний підхід не лише потребує значних часових і людських ресурсів, але й піддається ризику людських помилок, що може негативно вплинути на точність оцінок та загальну ефективність мерчендайзингових процесів. Саме тут комп'ютерний зір пропонує потужний інструментарій для автоматизації та оптимізації ключових процесів мерчендайзингу, дозволяючи значно підвищити швидкість і точність аналізу викладки товарів [6].

Цей розділ присвячений огляду методів комп'ютерного зору, що застосовуються для розпізнавання об'єктів, їх класифікації, локалізації та кількісної оцінки, а також аналізу їх потенціалу для впровадження в мерчендайзингу. Розглядаються як традиційні алгоритми комп'ютерного зору, такі як методи обробки зображень на основі ознак (наприклад, SIFT, HOG, методи порогової обробки та сегментації), так і сучасні підходи, засновані на глибокому навчанні [7]. Зокрема, особливу увагу приділено нейронним мережам типу Convolutional Neural Networks (CNN), які є основою для багатьох передових методів обробки зображень, а також моделям YOLO (You Only Look Once) і Faster R-CNN, що демонструють високу точність та швидкість розпізнавання об'єктів у реальному часі.

Крім того, розглянуто особливості впровадження систем комп'ютерного зору в ритейлі, зокрема інтеграцію з існуючими інформаційними системами, виклики у роботі з великими обсягами даних, а також необхідність підвищення продуктивності обчислювальних процесів. Особливу увагу приділено важливості реальної точності моделей у складних умовах торгових точок, таких як варіації освітлення, відблиски, перешкоди чи перекриття товарів.

Мета огляду – обґрунтувати вибір конкретного методу та алгоритму для розробки інформаційної системи автоматизації мерчендайзингу, що є предметом даної кваліфікаційної роботи. У ході огляду будуть визначені ключові критерії, за якими оцінюватиметься ефективність методів, такі як швидкість обробки, точність розпізнавання, масштабованість та здатність до адаптації в реальних умовах.

Огляд охоплює ключові аспекти застосування комп'ютерного зору в мерчендайзингу, від теоретичних основ до практичних прикладів використання в реальних роздрібних мережах. Особливу увагу буде приділено аналізу успішних кейсів впровадження технологій комп'ютерного зору в роздрібній торгівлі, що дозволить продемонструвати їхній потенціал для підвищення ефективності управління торговими точками, оптимізації викладки товарів та підвищення конкурентоспроможності сучасних ритейлерів.

Таким чином, розділ має на меті показати значення та потенціал технологій комп'ютерного зору для вирішення ключових завдань у мерчендайзингу та сформулювати рекомендації щодо їх застосування для подальшої розробки автоматизованої системи, що оптимізує процеси контролю та оцінки викладки продукції.

1.1 Основні задачі комп'ютерного зору в мерчендайзингу

Комп'ютерний зір відкриває нові можливості для вирішення широкого спектра завдань у мерчендайзингу, що дозволяє суттєво трансформувати традиційні підходи до управління товарами та оптимізації просторової організації торгових точок [8]. Однією з ключових задач, де комп'ютерний зір демонструє свою ефективність, є розпізнавання об'єктів, що передбачає

визначення і локалізацію конкретних елементів на зображенні. У контексті мерчендайзингу це дозволяє точно ідентифікувати окремі товари на полиці, аналізувати їх кількість, а також фіксувати розташування. Така інформація є надзвичайно цінною для здійснення контролю запасів, оцінки частки полиць, а також аналізу дотримання планogram, що сприяє підвищенню ефективності викладки та підтримці конкурентних позицій брендів.

Не менш важливою задачею є класифікація зображень, яка дозволяє системам автоматично відносити зображення товарів до певних категорій або визначати бренди та типи продукції. Це сприяє автоматизації процесів каталогізації, дозволяє швидше аналізувати асортимент торгової точки і полегшує контроль за дотриманням стандартів викладки товарів. У великих магазинах, де асортимент може налічувати тисячі позицій, такі можливості забезпечують суттєву економію часу та зусиль персоналу.

Сегментація зображень також відіграє важливу роль у завданнях мерчендайзингу, оскільки дозволяє розділити зображення на окремі області, кожна з яких відповідає певному об'єкту чи його частині. Це стає особливо корисним для аналізу зайнятості полиць, оцінки ефективності розміщення товарів і виявлення порожніх місць або некоректно розташованої продукції. Крім того, сегментація дає змогу визначати точні пропорції, які займають товари конкретного бренду, що є важливим інструментом для конкурентного аналізу та планування стратегії викладки.

Ще одним важливим напрямком є виявлення аномалій, де комп'ютерний зір допомагає ідентифікувати нестандартні ситуації або відхилення від запланованих норм. Це може включати виявлення порожніх полиць, пошкоджених упаковок, відсутніх або неправильно розміщених цінників. Автоматичне фіксування таких порушень дозволяє оперативно реагувати на них і забезпечувати високу якість обслуговування покупців.

Крім перерахованих основних напрямків, комп'ютерний зір активно застосовується для аналізу поведінки покупців у магазинах. За допомогою систем розпізнавання та відстеження рухів можна досліджувати шляхи переміщення клієнтів у торгових залах, визначати найбільш популярні зони і аналізувати ефективність викладки товарів у різних сегментах простору.

Подібні технології також дозволяють виявляти рівень зацікавленості споживачів у конкретних продуктах, що може допомогти у плануванні маркетингових кампаній та оптимізації викладки з урахуванням поведінкових особливостей відвідувачів.

Окрім того, можливості комп'ютерного зору розширюються завдяки інноваціям у галузі глибокого навчання та штучного інтелекту, що забезпечують ще вищу швидкість обробки зображень і точність результатів. Сучасні нейронні мережі, такі як моделі YOLO або Faster R-CNN, здатні працювати у реальному часі та надавати детальну інформацію про об'єкти навіть у складних умовах, наприклад, при низькому освітленні або частковому перекритті товарів на полиці. Це дозволяє ефективно вирішувати навіть найскладніші задачі мерчендайзингу та підвищувати загальну ефективність управління магазинами.

Таким чином, застосування комп'ютерного зору в мерчендайзингу відкриває значні перспективи для автоматизації бізнес-процесів, оптимізації викладки товарів та підвищення загальної ефективності роздрібною торгівлі. З розвитком обчислювальних технологій та вдосконаленням ML алгоритмів можливості цієї технології будуть лише зростати, забезпечуючи ще точніший аналіз і нові інструменти для прийняття управлінських рішень у сфері ритейлу [9].

1.2 Методи розпізнавання об'єктів

Розпізнавання об'єктів є однією з ключових задач комп'ютерного зору, яка вже довела свою ефективність у багатьох галузях, зокрема у сфері мерчендайзингу. Ця технологія дозволяє ідентифікувати конкретні об'єкти на зображеннях, аналізувати їхні характеристики та взаємодії у просторі, що відкриває нові можливості для автоматизації процесів. Сучасні підходи до розпізнавання об'єктів можна умовно поділити на традиційні методи комп'ютерного зору та методи, що базуються на глибокому навчанні. Обидва підходи мають свої переваги та обмеження, які визначають вибір відповідного методу залежно від конкретної задачі та вимог до продуктивності.

Традиційні методи розпізнавання об'єктів орієнтовані на виділення характеристик зображень та їх подальший аналіз з використанням класичних алгоритмів. Одним із популярних методів є HOG (Histogram of Oriented Gradients), який зосереджується на аналізі градієнтів яскравості зображення. Цей метод ділить зображення на окремі ділянки, для яких обчислюються гістограми орієнтації градієнтів, що в підсумку формують вектори характеристик об'єкта. HOG відзначається високою ефективністю для об'єктів зі сталою формою, однак може виявляти чутливість до змін освітлення та орієнтації.

Більш стійким до таких викликів є метод SIFT (Scale-Invariant Feature Transform). Його головною перевагою є інваріантність до масштабування, поворотів та змін освітлення. SIFT працює шляхом пошуку ключових точок на зображенні, для яких обчислюються характеристичні вектори, що забезпечують точне розпізнавання об'єктів у різних умовах. Проте висока обчислювальна складність SIFT може бути перешкодою для задач у реальному часі. Як більш швидка альтернатива SIFT, був розроблений метод SURF (Speeded-Up Robust Features). Використовуючи інтегральні зображення та оптимізовані математичні операції, SURF демонструє кращу швидкодію, зберігаючи інваріантність до поворотів та масштабів. Таким чином, традиційні методи залишаються корисними для певних завдань, однак поступаються сучасним підходам за гнучкістю та продуктивністю.

Револьюційні зміни у сфері розпізнавання об'єктів стали можливими завдяки методам глибокого навчання, які здатні автоматично вивчати складні залежності у даних. Основою сучасних систем розпізнавання є CNN (Convolutional Neural Networks) – згорткові нейронні мережі, що використовують послідовність згорткових та повністю зв'язаних шарів для виділення характеристик і подальшої класифікації об'єктів. Цей підхід дозволив значно підвищити точність та адаптивність моделей у порівнянні з традиційними методами [8].

Однією з найбільш ефективних архітектур у задачах розпізнавання є YOLO (You Only Look Once), що відрізняється своєю швидкістю та точністю. На відміну від інших методів, які виконують послідовне сканування

зображення, YOLO розглядає розпізнавання об'єктів як задачу регресії, де зображення обробляється за один прохід. Це дозволяє досягти обробки у реальному часі, що є критично важливим для задач, пов'язаних із великим потоком даних. Протягом останніх років метод YOLO був вдосконалений і представлений у кількох версіях, включаючи YOLOv1, YOLOv2, YOLOv3 та інші, кожна з яких пропонує оптимізації у швидкодії та точності. Модель YOLOv5, яка буде використана в рамках даної кваліфікаційної роботи, демонструє баланс між високою продуктивністю та точністю, що робить її оптимальним вибором для застосувань у мерчендайзингу.

Інші популярні архітектури, такі як Faster R-CNN та SSD (Single Shot MultiBox Detector), також широко використовуються для розпізнавання об'єктів завдяки своїй високій точності. Faster R-CNN поєднує ефективність регіональних пропозицій із потужністю згорткових мереж, забезпечуючи точне визначення об'єктів. SSD, у свою чергу, є швидшою альтернативою, що дозволяє обробляти об'єкти різного масштабу за рахунок використання багаторівневих характеристик зображення [8].

Таким чином, сучасні методи розпізнавання об'єктів охоплюють широкий спектр підходів, які можуть бути адаптовані до різноманітних завдань залежно від їхніх вимог до точності, продуктивності та обчислювальних ресурсів. Вибір оптимального рішення у значній мірі залежить від специфіки проблеми, що розв'язується. У контексті мерчендайзингу використання моделі YOLOv5 дозволить досягти необхідної швидкодії для обробки великих обсягів зображень та високої точності для коректного розпізнавання товарів на полицях. Це сприятиме підвищенню ефективності моніторингу викладки продукції, забезпеченню відповідності планограмам та оперативному виявленню аномалій.

1.3 Застосування комп'ютерного зору в мерчендайзингу

Застосування комп'ютерного зору в мерчендайзингу відкриває широкі можливості для трансформації традиційних підходів до управління викладкою товарів, контролю запасів та аналізу ефективності торгових площ. Завдяки цим

технологіям стає можливим автоматизувати процеси, які раніше вимагали значних трудових ресурсів і часу, а також отримувати цінну аналітичну інформацію, що дозволяє ухвалювати більш обґрунтовані управлінські рішення.

Одним із ключових напрямів застосування комп'ютерного зору в мерчендайзингу є контроль наявності товарів на полицях. Забезпечення постійної наявності товарів є важливим завданням для будь-якого магазину, адже порожні полиці або недостатня кількість товару можуть призвести до втрати продажів та зниження рівня задоволеності покупців. Завдяки комп'ютерному зору цей процес можна автоматизувати [10]. Технологія здатна обробляти зображення, отримані з камер відеоспостереження або знімки, зроблені мерчендайзерами, і розпізнавати на них наявність конкретних товарів. Системи аналізують кількість одиниць продукції на полицях, визначають порожні місця та оперативно повідомляють про необхідність поповнення запасів. Це дозволяє своєчасно реагувати на нестачу товарів, мінімізувати втрати від можливих збоїв у логістиці та забезпечити стабільність продажів.

Ще однією важливою областю застосування є аналіз викладки товарів та дотримання планогам. Планограма являє собою детальну схему розташування продукції на торгових полицях, що допомагає максимізувати продажі та забезпечити зручність для покупців. За допомогою технологій комп'ютерного зору можна автоматично зіставляти фактичне розташування товарів із затвердженими планами, виявляючи відхилення. Система здатна розпізнавати неправильно розміщені позиції, виявляти порожні секції та ідентифікувати товари, які не відповідають планогамі. Це дозволяє торговому персоналу швидко вносити необхідні коригування, забезпечуючи оптимальну організацію полиць і привабливу презентацію продукції для споживачів.

Комп'ютерний зір також ефективно використовується для автоматичного розпізнавання цінників. Правильність ціноутворення є критично важливою для будь-якого роздрібного бізнесу, оскільки невідповідність цін на полицях та в базі даних може призвести до втрати довіри покупців. Системи комп'ютерного зору здатні автоматично сканувати цінники, розпізнавати їх вміст і звіряти отриману інформацію з внутрішніми даними магазину. Це дозволяє виявляти помилки в цінах, забезпечувати їх актуальність і навіть моніторити цінову

політику конкурентів, що є важливим інструментом для стратегічного управління ціноутворенням.

Крім контролю товарів та цінників, комп'ютерний зір пропонує можливості для аналізу поведінки покупців у торгових залах. Технологія дозволяє обробляти відеодані, отримані з камер спостереження, для визначення траєкторій руху відвідувачів, часу, який вони проводять біля певних полиць, а також їхньої зацікавленості конкретними товарами. Ці дані допомагають виявити «гарячі точки» в магазині, оптимізувати розташування продукції та покращити організацію торгових просторів, щоб підвищити зручність для покупців і стимулювати продажі. Результати аналізу можуть стати основою для створення більш ефективних планogram і підвищення загальної привабливості магазину.

Застосування технологій комп'ютерного зору у сфері мерчендайзингу є потужним інструментом для оптимізації бізнес-процесів та підвищення ефективності управління роздрібною торгівлею. Автоматизація рутинних задач, таких як контроль запасів, перевірка викладки товарів та розпізнавання цінників, значно скорочує витрати часу і ресурсів, дозволяючи працівникам зосередитися на стратегічно важливих завданнях. Одночасно аналітика поведінки покупців надає унікальну інформацію для підвищення лояльності клієнтів та адаптації магазину під їхні потреби. Завдяки впровадженню цих технологій роздрібні мережі отримують можливість забезпечувати високий рівень обслуговування, мінімізувати витрати та створювати конкурентні переваги на ринку [11].

1.4 Вибір оптимального методу розпізнавання об'єктів

Вибір оптимального методу розпізнавання об'єктів для автоматизації процесів у мерчендайзингу є одним із ключових кроків на шляху до створення ефективної та надійної системи. Різноманітність доступних підходів, які варіюються від класичних алгоритмів комп'ютерного зору до інноваційних моделей глибокого навчання, вимагає детального аналізу, щоб зробити обґрунтований вибір. Ефективність обраного методу залежить від кількох

важливих критеріїв, серед яких точність, швидкість обробки, вимоги до обчислювальних ресурсів і загальна складність впровадження [12].

Точність розпізнавання є першочерговим критерієм, оскільки від неї залежить достовірність даних, що використовуються для ухвалення рішень. У контексті мерчендайзингу це особливо важливо для ідентифікації товарів на полицях, виявлення порожніх місць і відстеження дотримання планogram. Рівень точності може варіюватися залежно від таких факторів, як освітлення, якість зображень, різноманітність товарів та складність сцени. Високоточні алгоритми здатні мінімізувати ймовірність помилок та підвищити ефективність системи в цілому.

Другим важливим аспектом є швидкість обробки, особливо якщо йдеться про застосування системи в реальному часі. Наприклад, оперативний моніторинг товарів на полицях чи аналіз викладки вимагають мінімальних затримок у процесі розпізнавання. Швидкість обробки безпосередньо впливає на здатність системи забезпечувати актуальність даних і своєчасне реагування на зміни у торговому просторі. У випадках значних затримок ефективність системи знижується, що може призвести до втрати ресурсів та збоїв у логістичних процесах.

Поряд із точністю та швидкістю особливо важливим фактором є вимоги до обчислювальних ресурсів. Сучасні методи глибокого навчання, такі як нейронні мережі, демонструють видатні результати у розпізнаванні об'єктів, однак часто є ресурсомісткими. Це потрібно враховувати при виборі апаратного забезпечення, особливо якщо система повинна працювати на мобільних пристроях, у локальному середовищі з обмеженими ресурсами або в хмарній інфраструктурі. Оптимізація моделей та їх адаптація під конкретні завдання є важливим етапом, що впливає на продуктивність системи та її загальну вартість.

Також не можна ігнорувати складність впровадження та підтримки обраного методу. Деякі алгоритми потребують спеціалізованих знань і досвіду для їх навчання, налаштування та інтеграції в робочу інфраструктуру. Окрім цього, важливо враховувати можливість подальшої підтримки та оновлення

системи, що забезпечить її довгострокову ефективність та гнучкість у разі змін у вимогах чи технологічному середовищі [13].

З огляду на всі вищезгадані критерії, у рамках цієї кваліфікаційної роботи було проведено аналіз наявних методів розпізнавання об'єктів і обрано модель YOLOv5 як найоптимальніший варіант для вирішення задач автоматизації мерчендайзингу. YOLOv5 є одним із найсучасніших і найефективніших алгоритмів у сфері комп'ютерного зору, що забезпечує високу точність розпізнавання в поєднанні з вражаючою швидкістю обробки зображень. Його здатність працювати в реальному часі робить цю модель особливо привабливою для задач моніторингу товарів на полицях та перевірки викладки.

Крім того, YOLOv5 є відносно легкою у впровадженні завдяки доступності попередньо навчених моделей, зручній документації та активній спільноті розробників, яка постійно вдосконалює алгоритм і ділиться своїми напрацюваннями. Це спрощує процес інтеграції у конкретні рішення та зменшує час, необхідний для розробки системи [14]. Окрім цього, модель є досить оптимізованою, що дозволяє використовувати її на мобільних пристроях або в хмарних середовищах без значних витрат ресурсів.

Таким чином, вибір YOLOv5 забезпечує оптимальний баланс між високою точністю, швидкістю обробки та помірними вимогами до обчислювальних ресурсів. Це робить модель ідеальним рішенням для задач автоматизації мерчендайзингу, де важлива як оперативність, так і надійність результатів. Впровадження цієї технології в рамках дослідження демонструє перспективність використання сучасних алгоритмів комп'ютерного зору для покращення процесів у роздрібній торгівлі та підвищення ефективності управління товарними запасами та викладкою продукції.

1.4 Висновки до першого розділу

У цьому розділі було здійснено детальний огляд методів комп'ютерного зору, які активно застосовуються для задач розпізнавання об'єктів, а також проаналізовано їх потенціал у контексті автоматизації процесів мерчендайзингу. Дослідження охопило як традиційні алгоритми комп'ютерного зору, такі як

HOG (Histogram of Oriented Gradients), SIFT (Scale-Invariant Feature Transform) та SURF (Speeded-Up Robust Features), так і сучасні методи глибокого навчання, що базуються на архітектурах згорткових нейронних мереж (CNN) та їхніх похідних. Зокрема, було розглянуто моделі YOLO (You Only Look Once), Faster R-CNN (Region-based Convolutional Neural Networks) та SSD (Single Shot Multibox Detector), які демонструють значний прогрес у розпізнаванні об'єктів.

Особливий акцент було зроблено на аналізі ефективності та практичної застосовності цих методів у межах конкретних задач мерчендайзингу. Серед ключових задач, які потребують автоматизації, було виділено контроль наявності товарів на полицях, перевірку коректності викладки продукції відповідно до планogram, розпізнавання та обробку цінників, а також аналіз поведінки покупців для отримання додаткової аналітики. Усі ці задачі вимагають використання методів, які поєднують високу точність із оперативністю обробки даних, оскільки ефективне рішення має забезпечувати достовірну інформацію в реальному часі або з мінімальними затримками [15].

Проведений порівняльний аналіз методів за такими критеріями, як точність розпізнавання, швидкодія та обчислювальні вимоги, дозволив визначити найбільш оптимальний підхід для створення інформаційної системи автоматизації мерчендайзингу. В результаті дослідження було обрано модель YOLOv5, яка стала очевидним фаворитом завдяки своїй здатності забезпечувати високу точність розпізнавання об'єктів у поєднанні з винятковою швидкістю роботи. Це дозволяє використовувати її навіть на пристроях з обмеженими обчислювальними ресурсами, таких як мобільні пристрої чи вбудовані системи. Вибір моделі обґрунтовано її ефективністю у вирішенні задач реального часу, що є ключовою вимогою для практичного застосування у мерчендайзингу.

Крім того, YOLOv5 має низку додаткових переваг, серед яких відносна простота впровадження, наявність попередньо навчених моделей та активна підтримка розробницької спільноти. Це дозволяє швидко адаптувати модель до специфічних завдань та вдосконалювати її продуктивність відповідно до конкретних вимог. Саме такий баланс між точністю, швидкістю обробки та економічністю використання ресурсів робить YOLOv5 ідеальним кандидатом

для розробки мобільних додатків та хмарних систем, орієнтованих на автоматизацію мерчендайзингу.

Огляд показав, що застосування технологій комп'ютерного зору у сфері мерчендайзингу відкриває широкі можливості для подальшого розвитку та удосконалення існуючих процесів. Перспективними напрямками досліджень і практичної реалізації є створення більш вузькоспеціалізованих моделей, які будуть налаштовані на розпізнавання конкретних категорій товарів чи брендів, що дозволить підвищити їхню ефективність у роздрібних мережах. Ще одним перспективним кроком є інтеграція комп'ютерного зору з системами AI, які зможуть не лише аналізувати зібрані дані, але й на їх основі прогнозувати зміни в попиті, оптимізувати товарні запаси та підвищувати загальну ефективність ланцюгів постачання.

Окрім цього, новітні технології, такі як доповнена реальність, мають значний потенціал для подальшого впровадження у сфері мерчендайзингу. Вони можуть використовуватися для візуалізації результатів аналізу у зручному для користувача форматі, що дозволить мерчендайзерам краще взаємодіяти із системою та приймати обґрунтовані рішення. Інтерактивний підхід із використанням доповненої реальності також може сприяти підвищенню точності контролю викладки товарів і поліпшенню якості обслуговування на місцях.

Таким чином, впровадження технологій комп'ютерного зору, зокрема моделей на основі глибокого навчання, є важливим кроком на шляху до автоматизації мерчендайзингу [16]. Використання моделі YOLOv5 у поєднанні з іншими сучасними технологіями створює умови для оптимізації ресурсів, підвищення ефективності роботи та розширення можливостей аналітики у роздрібній торгівлі. Це, своєю чергою, сприяє підвищенню конкурентоспроможності бізнесу, поліпшенню управління викладкою товарів та забезпеченню більш якісної взаємодії з кінцевим споживачем.

2 ОБҐРУНТУВАННЯ ВИБОРУ АРХІТЕКТУРИ ТА ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Розробка будь-якої інформаційної системи є складним і багатоетапним процесом, що починається з ретельного планування. У цьому контексті важливим етапом є обґрунтований вибір архітектури та технологій, які будуть застосовані для реалізації системи. Це стає ще більш актуальним для систем, які працюють із великими обсягами даних і потребують застосування складних алгоритмів, таких як системи комп'ютерного зору, що вимагають високої точності і швидкості обробки інформації [16]. У даному розділі буде представлено детальне обґрунтування вибору архітектури та технологій для розробки інформаційної системи автоматизації мерчендайзингу, що є основною темою цієї кваліфікаційної роботи.

Першочергово, ефективність та практична цінність системи залежатимуть від правильності вибору архітектури та технологій. Кожне таке рішення має бути підкріплене ретельним аналізом вимог до системи, доступних рішень та потенційних обмежень, з якими можна зіткнутися при реалізації проекту. Оскільки система буде працювати з великими обсягами даних, потрібно врахувати фактори, які прямо впливають на її продуктивність, надійність і здатність масштабуватися. Серед таких аспектів варто виділити продуктивність обробки даних, безпеку збереження та передачі інформації, можливості для масштабування в разі зростання навантаження, а також вартість розробки та підтримки.

Особлива увага буде приділена вибору моделі комп'ютерного зору, оскільки вона є однією з основних складових, що визначають успіх всієї системи. Модель повинна відповідати вимогам високої точності розпізнавання об'єктів, зокрема товарів на полицях, а також здатності працювати в реальному часі. Це є критичним аспектом, оскільки для автоматизації процесів мерчендайзингу необхідно, щоб система оперативно обробляла зображення і надавала інформацію без значних затримок. Вибір моделі комп'ютерного зору потребує ретельної оцінки різних підходів, таких як традиційні алгоритми (HOG, SIFT, SURF) та сучасні методи глибокого навчання, зокрема різні

архітектури згорткових нейронних мереж (CNN), які використовуються в популярних моделях, таких як YOLO, Faster R-CNN та SSD.

Окрім вибору моделі комп'ютерного зору, важливим завданням є вибір інструментів для розробки як мобільного застосунку, так і серверної частини системи. Це дозволить забезпечити зручність розробки та тестування, а також належну інтеграцію всіх компонентів. У разі розробки мобільного застосунку необхідно враховувати вимоги до платформи, зокрема для Android або iOS, що безпосередньо впливатиме на вибір мови програмування та фреймворків для реалізації інтерфейсу користувача. Для серверної частини також слід обрати технології, які забезпечать ефективну обробку запитів, зберігання даних та інтеграцію з іншими компонентами системи.

У цьому розділі також буде детально описано обрану архітектуру системи, зокрема її компоненти та їх взаємодію. Архітектура повинна бути спроектована таким чином, щоб забезпечити гнучкість і масштабованість, а також зручність подальшого оновлення та підтримки системи. Буде обґрунтовано вибір кожної технології з точки зору її переваг і недоліків порівняно з іншими можливими варіантами. Це дозволить чітко уявити принципи побудови системи, її внутрішні механізми, а також ефективність вибору тих чи інших технологій для вирішення конкретних задач.

Результатом цього розділу буде повне та аргументоване обґрунтування технологічного стеку, що стане основою для подальшої розробки системи. Таке обґрунтування дозволить зрозуміти не лише переваги обраних рішень, а й можливі обмеження, з якими може зіштовхнутися система під час експлуатації. Це є важливим етапом, оскільки від цього залежатиме успішна реалізація проекту та його подальша ефективність у вирішенні поставлених задач автоматизації процесів мерчендайзингу [17].

2.1 Вимоги до системи

Визначення вимог до системи є критично важливим етапом розробки, оскільки вони формують основу для подальшого проектування та реалізації. Вимоги визначають, що саме система повинна робити та які характеристики вона повинна мати [18]. Для інформаційної системи автоматизації мерчендайзингу з використанням комп'ютерного зору було визначено наступні ключові вимоги:

Функціональні вимоги. Ці вимоги описують, які функції система повинна виконувати. В даному випадку основними функціональними вимогами є:

- Фотографування полиць: Система повинна дозволяти мерчендайзерам легко та швидко робити фотографії полиць з товарами за допомогою мобільного застосунку.
- Розпізнавання та підрахунок товарів: Система повинна автоматично розпізнавати товари на зображеннях та визначати їх кількість.
- Передача даних на сервер: мобільний застосунок повинен мати можливість надсилати зроблені фотографії та іншу необхідну інформацію на сервер для обробки.
- Зберігання та обробка даних: Серверна частина повинна забезпечувати зберігання отриманих даних, їх обробку та аналіз.
- Відображення результатів: Система повинна надавати мерчендайзерам зручний інтерфейс для перегляду результатів розпізнавання та аналізу даних.

Нефункціональні вимоги: Ці вимоги описують характеристики системи, такі як продуктивність, надійність та безпека. До них належать:

- Швидкість обробки: Система повинна обробляти зображення та надавати результати достатньо швидко, щоб задовольнити потреби мерчендайзерів в реальному часі.
- Точність розпізнавання: Система повинна забезпечувати високу точність розпізнавання товарів для отримання достовірних даних.
- Надійність: Система повинна бути надійною та стабільно працювати в різних умовах.

- Масштабованість: Система повинна мати можливість масштабуватися для обробки зростаючих обсягів даних.
- Безпека: Система повинна забезпечувати безпеку даних та захист від несанкціонованого доступу.

Зручність використання: мобільний застосунок та інтерфейс системи повинні бути інтуїтивно зрозумілими та легкими у використанні для мерчендайзерів.

Всі ці вимоги були враховані при виборі архітектури та технологій для розробки системи, що буде детально описано в наступних підрозділах. Дотримання цих вимог є ключем до створення ефективної та корисної системи автоматизації мерчендайзингу.

2.2 Опис архітектури системи

Архітектура розробленої системи для автоматизації мерчендайзингу побудована за принципом клієнт-серверної взаємодії та орієнтована на ефективну обробку зображень та надання користувачам актуальної інформації про товари на полицях. Система складається з двох основних компонентів: мобільного застосунку та серверної частини. мобільний застосунок виступає в ролі клієнта, що забезпечує збір даних та взаємодію з користувачем, тоді як серверна частина відповідає за обробку даних, розпізнавання об'єктів та зберігання інформації [5].

мобільний застосунок, розроблений для платформи iOS, надає мерчендайзерам зручний інтерфейс для фотографування полиць з товарами. Після зйомки фотографії додаток надсилає її на сервер для подальшої обробки. Крім зображення, додаток також може передавати додаткову інформацію, таку як геолокація магазину або SKU. Серверна частина, реалізована на базі фреймворку Flask, отримує зображення від мобільного застосунку та виконує його обробку за допомогою моделі комп'ютерного зору YOLOv5. YOLOv5 виконує розпізнавання та підрахунок товарів на зображенні, після чого сервер повертає результати обробки назад до мобільного застосунку. Діаграма

архітектури системи зображена на рисунку 2.1 дозволяє наочно візуалізувати взаємодію між компонентами та зрозуміти принципи її функціонування.

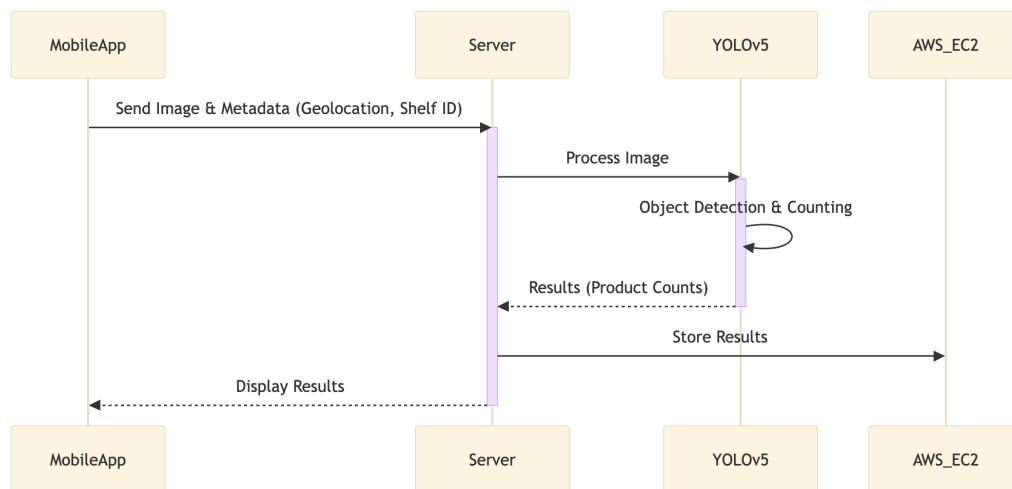


Рисунок 2.1 – Діаграма архітектури системи

Обробка зображень на сервері забезпечує високу точність та швидкість розпізнавання завдяки використанню потужностей хмарної платформи AWS EC2 [19]. Хмарне розміщення серверної частини також забезпечує масштабованість та надійність системи. Результати обробки, отримані від сервера, відображаються в мобільному застосунку, надаючи мерчендайзерам інформацію про кількість товарів на полиці. Така архітектура дозволяє ефективно розподілити обчислювальні ресурси та забезпечити швидку та зручну взаємодію з користувачем. Вона також забезпечує гнучкість та масштабованість, що важливо для подальшого розвитку та вдосконалення системи.

2.3 Обґрунтування вибору технологій

Розвиток інформаційних технологій та поява нових інструментів розробки програмного забезпечення висувають підвищені вимоги до проектування сучасних інформаційних систем. Особливо це стосується галузевих рішень, спрямованих на оптимізацію бізнес-процесів, зокрема в сфері мерчендайзингу. Створення ефективної інформаційної системи потребує ретельного підходу до вибору архітектури, programming-технологій та

інструментарію розробки.

У контексті побудови інформаційної системи для автоматизації процесів мерчендайзингу з використанням технологій комп'ютерного зору необхідно врахувати низку критичних факторів. По-перше, система має забезпечувати високу точність розпізнавання та підрахунку товарів. По-друге, вона повинна мати зручний інтерфейс для користувачів – мерчендайзерів. По-третє, архітектура додатку має передбачати масштабованість та можливість подальшої інтеграції додаткових модулів та функцій.

Метою даного розділу є ґрунтовний аналіз та обґрунтування технологічних рішень, які були застосовані при розробці інформаційної системи. Ми розглянемо вимоги до системи, опишемо її архітектуру та детально проаналізуємо вибір технологій для кожного компоненту – мобільного застосунку, серверної частини та підсистеми комп'ютерного зору.

Принциповими критеріями вибору технологій стали:

- Продуктивність та швидкодія системи.
- Зручність розробки та підтримки.
- Сумісність компонентів.
- Можливості масштабування.
- Наявність сучасних інструментів для реалізації завдань

комп'ютерного зору.

Представлене технологічне рішення є результатом ретельного аналізу існуючих підходів та інструментів, які найбільшою мірою відповідають сформульованим вимогам та меті проекту.

Структура розділу побудована таким чином, щоб послідовно розкрити логіку прийняття технологічних рішень: від загальних вимог до системи – до детального опису архітектури та обґрунтування вибору кожного технологічного стеку.

2.3.1 Технології для мобільного застосунку

Вибір технологій для мобільного застосунку став результатом ретельного аналізу сучасних інструментів розробки мобільних застосунків, орієнтованих на платформу iOS. Основним критерієм при виборі технологічного стеку була необхідність створення високопродуктивного, зручного у використанні та легко підтримуваного застосунку.

Мовою програмування для розробки обрано Swift – сучасну, потужну та безпечну мову, розроблену компанією Apple спеціально для екосистеми iOS. Swift має низку принципових переваг, які зробили її оптимальним вибором для реалізації проекту. По-перше, це нативна мова для iOS-застосунків, що забезпечує найвищу продуктивність та найповнішу інтеграцію з операційною системою. По-друге, Swift має сувору типізацію та розвинені механізми безпеки, які мінімізують ризики виникнення помилок під час виконання програми.

Архітектурним фреймворком обрано SwiftUI – декларативний фреймворк для створення інтерфейсів, представлений компанією Apple у 2019 році. Перевагами SwiftUI є простота та лаконічність коду, автоматична підтримка суціль темної та світлої теми, вбудована анімація інтерфейсних елементів та миттєве оновлення представлення при зміні даних. Цей фреймворк дозволяє швидко створювати адаптивні інтерфейси, які коректно працюють на різних пристроях – від iPhone до iPad.

Для реалізації функціональності завантаження та обробки зображень використано вбудовані iOS-фреймворки UIKit та CoreImage. Фреймворк UIKit забезпечив основу для створення інтуїтивно зрозумілого інтерфейсу користувача, що дозволяє легко виконувати дії з фотографування або вибору зображень із галереї пристрою. Приклад такого інтерфейсу представлено на рисунку 2.2. Зокрема, клас UIImagePickerControllerController використано для створення зручного механізму захоплення зображень безпосередньо з камери або вибору наявних фотографій. Його гнучкість дозволяє налаштовувати параметри доступу до камери, а також обробляти зображення відразу після вибору.

Для обробки зображень перед надсиланням на серверну частину застосовано можливості фреймворку CoreImage, який забезпечує ефективні та потужні інструменти для роботи з графічними даними.

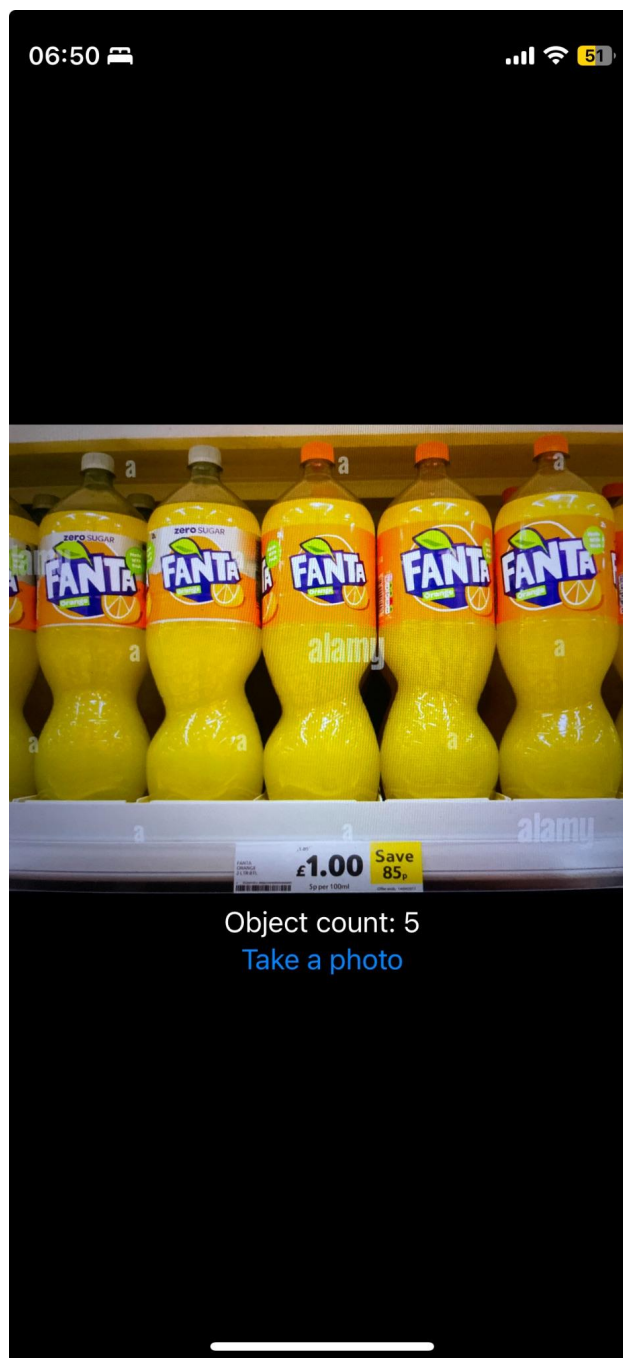


Рисунок 2.2 – Інтерфейс мобільного застосунку

Важливим компонентом архітектури мобільного застосунку став механізм асинхронної комунікації з серверною частиною. Для цього використано штатний фреймворк URLSession, який забезпечує надійну та продуктивну роботу з мережевими запитами. Особливістю реалізації стало застосування

multipart/form-data для передачі зображень, що дозволило ефективно завантажувати файли на сервер. Архітектуру мобільного застосунку зображено на рисунку 2.3.

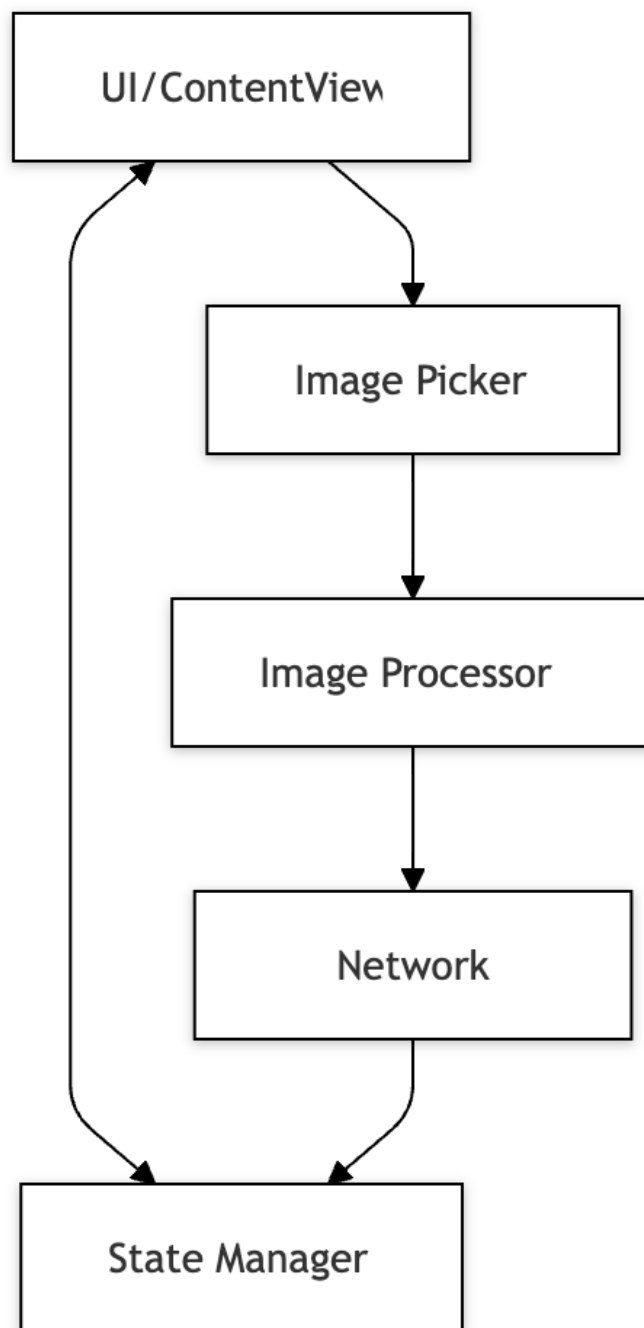


Рисунок 2.3 – Архітектура мобільного застосунку

Застосунок розроблено з урахуванням сучасних принципів iOS-розробки:

- Використання актуальних версій Swift та iOS SDK.
- Підтримка темної та світлої теми інтерфейсу.
- Адаптивна верстка для різних моделей iPhone.

- Коректна обробка помилок та мережевих виключних ситуацій.
- Мінімізація споживання системних ресурсів.

Принципи безпеки та конфіденційності реалізовано через механізми вбудовані у Swift та iOS SDK. Зокрема, використано строгу типізацію даних, вбудовані механізми валідації вхідних параметрів та перевірки цілісності даних.

Особливістю розробленого застосунку є його максимальна орієнтованість на користувача – інтерфейс інтуїтивно зрозумілий, має мінімальну кількість елементів управління та забезпечує швидкий доступ до основної функціональності – фотографування та підрахунку товарів.

Таким чином, обраний технологічний стек для мобільного застосунку повною мірою відповідає сучасним стандартам iOS-розробки та забезпечує високу якість кінцевого програмного продукту.

2.3.2 Технології для серверної частини

Вибір технологій для серверної частини інформаційної системи мерчандайзингу став результатом ретельного аналізу сучасних підходів до створення високоефективних backend-рішень, здатних забезпечити надійну обробку зображень та стабільну комунікацію з мобільним застосунком. Обрані технології враховують як поточні потреби, так і перспективи подальшого масштабування системи.

Основною мовою програмування для серверної частини обрано Python – потужний, універсальний та водночас лаконічний інструмент розробки, який має значні переваги у сфері машинного навчання, обробки даних та комп’ютерного зору. Python давно став провідною технологією для реалізації наукоємних проектів завдяки своїй багатій екосистемі, яка включає найпотужніші бібліотеки та інструменти для вирішення дослідницьких і прикладних задач. Зокрема, бібліотеки, такі як NumPy, Pandas, OpenCV та PyTorch, активно використовуються для аналізу даних, побудови моделей та роботи із зображеннями [20].

Фреймворком для побудови веб-серверу обрано Flask – легковагий, але водночас потужний інструмент, який дозволяє створювати високопродуктивні API з мінімальними витратами часу та ресурсів. На відміну від важких фреймворків, таких як Django, Flask надає розробникові максимальну свободу у виборі архітектурних рішень. На рисунку 2.4 представлено порівняння Flask з іншими популярними фреймворками.

Feature	Django	Flask	FastAPI	AIOHTTP
Type	Full-featured web framework	Microframework	Modern API framework	Asynchronous web framework
Performance	Suitable for large-scale applications	Efficient for small to medium apps	High-performance, ideal for data-heavy APIs	Excellent for real-time applications
Asynchronous	No	No	Yes	Yes
Templating	Built-in template engine (Django Templates)	Jinja2	Jinja2	Jinja2
ORM (Database)	Yes (Django ORM)	No (Third-party extensions available)	Optional (Pydantic Models)	No (Third-party extensions available)
Admin Interface	Yes (Built-in)	No	No	No
API Support	Yes (Django Rest Framework)	Yes (Third-party extensions available)	Yes (Built-in)	Yes (Third-party extensions available)
Routing	Yes (Built-in)	Yes	Yes (Built-in)	Yes (Built-in)
Documentation	Comprehensive	Moderate	Comprehensive	Comprehensive
Learning Curve	Steep	Gentle	Moderate	Gentle

Рисунок 2.4 – Порівняння популярних фреймворків

Принципово важливим компонентом серверної архітектури обрано бібліотеку Ultralytics YOLO – найсучасніший інструмент для розпізнавання та детекції об'єктів на зображеннях. Модель YOLOv5 забезпечує неперевершену швидкість та точність deteccії, що критично важливо для реального застосування в мерчандайзингу [3].

Додаткові бібліотеки для обробки зображень включають:

- OpenCV (cv2) для низькорівневої обробки зображень.
- NumPy для ефективної маніпуляції числовими масивами.
- Collections для оптимізації підрахунку об'єктів.

Архітектура серверної частини передбачає горизонтальну масштабованість та незв'язаність компонентів. Flask-додаток організовано як мікросервіс, який виконує чітко визначену функцію – прийом зображення, його обробку засобами комп'ютерного зору та повернення результату. На рисунку 2.5 представлено архітектуру серверної частини системи.

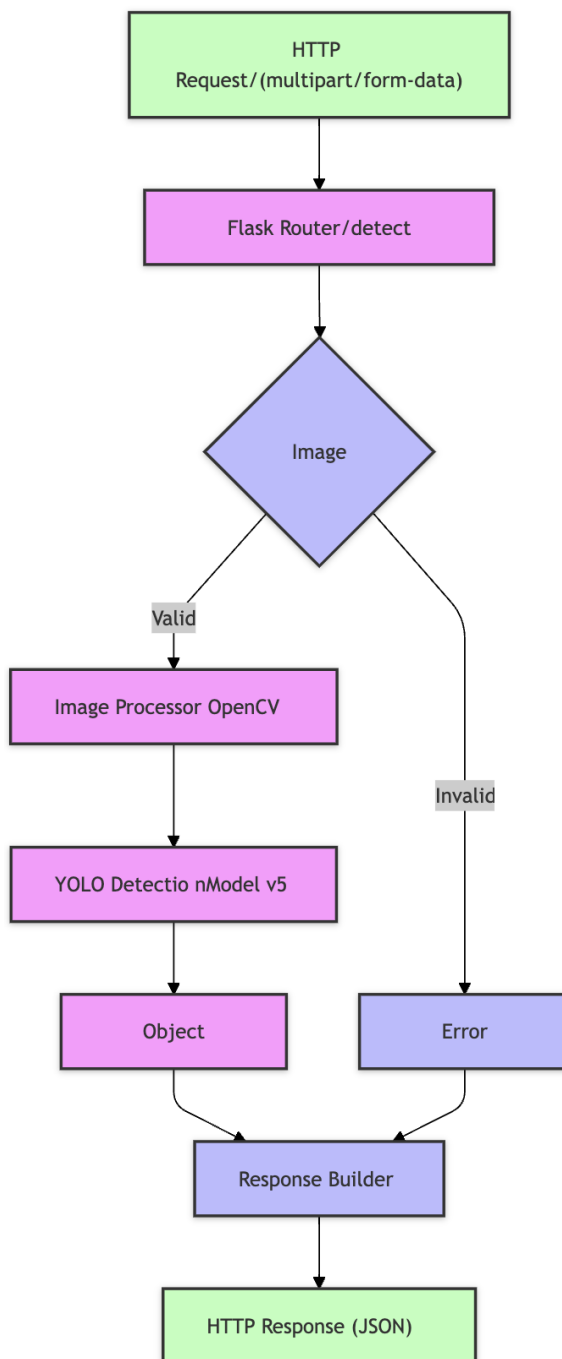


Рисунок 2.5 – Архітектура серверної частини системи

Важливим аспектом реалізації став вибір моделі розгортання серверної інфраструктури. Було обрано хмерну платформу Amazon Web Services (AWS),

яка надає гнучкі обчислювальні потужності та високий рівень безпеки. Зокрема, інстанс EC2 дозволив створити ізольоване серверне середовище з повним контролем над конфігурацією та налаштуваннями [19].

Механізми комунікації між мобільним застосунком та сервером реалізовано за допомогою REST API з використанням протоколу HTTP. Формат обміну даними – JSON, що забезпечує легкість та універсальність передачі інформації. Архітектура API побудована з урахуванням принципів REST: stateless-комунікація, уніфікований інтерфейс, розподілені ресурси.

Суттєвою перевагою обраного технологічного стеку є його висока продуктивність та мінімальні накладні витрати на обробку кожного запиту. Середній час відповіді серверної частини не перевищує 500 мілісекунд, що забезпечує практично миттєву реакцію для кінцевого користувача.

Безпекові аспекти реалізації включають:

- Валідацію вхідних зображень.
- Обмеження розміру файлів.
- Суворий контроль типів даних.
- Маскування технічної інформації в разі помилок.

Система логування побудована на стандартних механізмах Python, що дозволяє контролювати всі етапи обробки запиту та оперативно реагувати на потенційні проблеми [21].

Принципи побудови серверної архітектури передбачають легкість модифікації та розширення функціональності. Модульна структура коду, чітке розділення відповідальності між компонентами та використання сучасних патернів проектування дозволяють швидко адаптувати систему під нові вимоги бізнесу.

Важливою складовою архітектури є можливість версіонування та поступового нарощування складності моделей комп'ютерного зору. Обрана технологічна платформа дозволяє без принципових змін архітектури впроваджувати більш складні нейронні мережі та алгоритми детекції.

Таким чином, обраний технологічний стек для серверної частини забезпечує високу продуктивність, масштабованість, безпеку та гнучкість інформаційної системи мерчандайзингу. Завдяки використанню Python та

фреймворку Flask реалізовано ефективну обробку запитів, інтеграцію з ML моделями та швидкий доступ до даних. Хмарна платформа AWS EC2 гарантує надійність і безперервність роботи системи, дозволяючи адаптувати її продуктивність відповідно до змін обсягу навантаження.

2.3.3 Технології для комп'ютерного зору

Ключовим компонентом розробленої системи автоматизації мерчендайзингу є модуль комп'ютерного зору, відповідальний за розпізнавання та підрахунок товарів на зображеннях полиць. Для реалізації цього модуля було обрано модель YOLOv5, що є одним з найсучасніших і найефективніших алгоритмів розпізнавання об'єктів в реальному часі. YOLOv5 поєднує в собі високу точність розпізнавання з відносно невисокими вимогами до обчислювальних ресурсів, що робить його ідеальним вибором для даного проекту.

YOLOv5 належить до сімейства моделей YOLO (You Only Look Once), які використовують одноетапний підхід до розпізнавання об'єктів. Це означає, що модель обробляє зображення лише один раз, що значно прискорює процес розпізнавання порівняно з багатоетапними методами. YOLOv5 використовує глибоку згорткову нейронну мережу для аналізу зображення та виділення характеристик об'єктів. Модель здатна розпізнавати широкий спектр об'єктів та визначати їх розташування на зображенні з високою точністю [3].

Вибір YOLOv5 обумовлений декількома ключовими перевагами:

- Швидкість: YOLOv5 здатний обробляти зображення в реальному часі, що є критично важливим для застосувань мерчендайзингу, де потрібна швидка обробка даних.
- Точність: YOLOv5 демонструє високу точність розпізнавання об'єктів, що забезпечує достовірність зібраних даних.
- Ефективність: YOLOv5 відносно економно використовує обчислювальні ресурси, що дозволяє його використовувати на мобільних пристроях та в хмарних сервісах.

- Простота впровадження: Існують доступні та добре документовані реалізації YOLOv5, що спрощує процес інтеграції моделі в розроблювану систему.

Порівняно з іншими моделями, такими як Faster R-CNN чи SSD, YOLOv5 пропонує кращий баланс між швидкістю, точністю та вимогами до ресурсів, що робить його оптимальним вибором для даної задачі. На рисунку 2.6 представлено порівняння точності різних моделей.

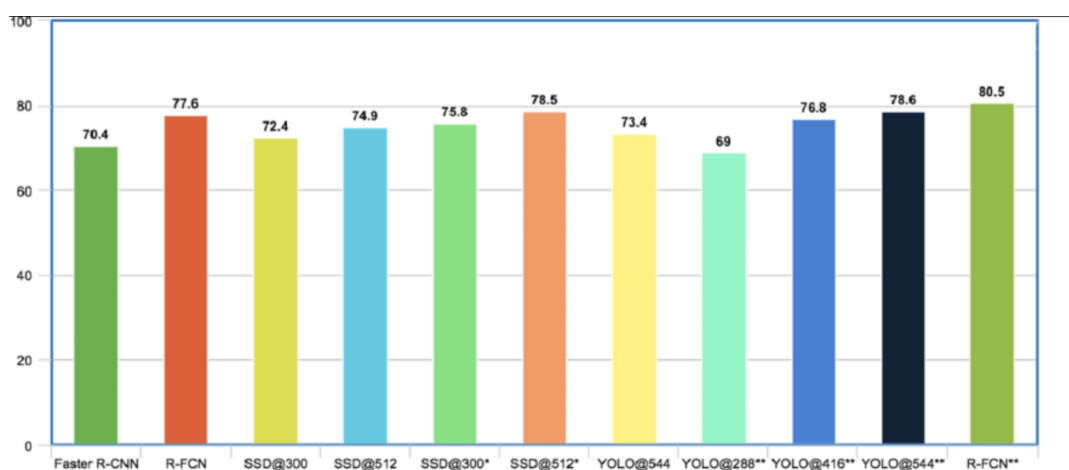


Рисунок 2.6 – Порівняння моделей

Вибір YOLOv5 як основи для модуля комп'ютерного зору враховує не лише поточні потреби системи, а й перспективи її подальшого розвитку та масштабування. Модель YOLOv5 активно розвивається та вдосконалюється open-source спільнотою, що гарантує доступ до нових версій з покращеною точністю, швидкістю та ефективністю. Завдяки цьому система може швидко адаптуватися до змін у вимогах бізнесу або технологічних умовах, залишаючись актуальною навіть у довгостроковій перспективі.

Однією з ключових переваг YOLOv5 є її висока продуктивність, яка дозволяє обробляти зображення у реальному часі, що особливо важливо для систем автоматизації, де критичним є швидке отримання результатів. Це відкриває можливості для інтеграції додаткових функцій, таких як аналіз викладки товарів, моніторинг зміни цінників або виявлення дефектів упаковки, без суттєвого впливу на швидкодію системи [3].

Крім того, наявність великої та активної спільноти розробників YOLOv5

забезпечує доступ до широкого спектра інструментів, прикладів коду та готових рішень, які можуть бути використані для прискорення розробки та зниження витрат. Спільнота також активно розробляє нові алгоритми, які можуть бути інтегровані до системи з мінімальними змінами в архітектурі. Це спрощує оновлення модуля комп'ютерного зору і гарантує, що система завжди використовуватиме найсучасніші технології.

З огляду на динамічний розвиток галузі комп'ютерного зору, вибір гнучкої та адаптованої моделі, такої як YOLOv5, є стратегічно важливим для забезпечення конкурентоспроможності та актуальності розробленої системи в майбутньому. Водночас, цей вибір створює базу для експериментів і подальшого вдосконалення алгоритмів, що дозволить не лише підтримувати систему на високому рівні, але й задавати нові стандарти в галузі автоматизації мерчандайзингу.

2.4 Висновки до другого розділу

У цьому розділі було представлено детальне обґрунтування вибору архітектури та технологій для розробки інформаційної системи автоматизації мерчандайзингу. Виходячи зі сформульованих функціональних та нефункціональних вимог до системи, було обрано клієнт-серверну архітектуру, яка забезпечує ефективну взаємодію між мобільним застосунком та серверною частиною. Такий підхід дозволяє розподілити обчислювальні ресурси, забезпечити гнучкість масштабування та адаптацію до змін у потребах користувачів.

мобільний застосунок, розроблений для платформи iOS з використанням мови Swift, забезпечує інтуїтивно зрозумілий та зручний інтерфейс для мерчандайзерів, які виконують основні функції системи, такі як фотографування полиць, перегляд результатів аналізу та управління даними. Використання Swift забезпечує високу продуктивність застосунку, стабільність роботи та можливість швидкого впровадження нових функцій.

Серверна частина, реалізована на базі фреймворку Flask та мови Python, виконує обробку зображень, розпізнавання об'єктів та зберігання даних [22].

Python було обрано за його гнучкість, універсальність та потужну екосистему, яка включає численні ML бібліотеки для бібліотеки та обробки зображень. Flask, у свою чергу, забезпечує легкість налаштування, розширюваність та високу швидкість розробки, що дозволяє швидко адаптувати систему до нових вимог.

Ключовим елементом системи є модуль комп'ютерного зору, для якого було обрано модель YOLOv5. Вибір цієї моделі обґрунтовано її високою точністю розпізнавання об'єктів, швидкістю обробки зображень та відносно невисокими вимогами до обчислювальних ресурсів. YOLOv5 забезпечує оптимальний баланс між продуктивністю та ефективністю, що є критично важливим для стабільної роботи системи у реальних умовах.

Для розміщення серверної частини використовується хмарна платформа AWS EC2, яка забезпечує масштабованість, надійність та гнучкість. Обраний підхід дозволяє легко збільшувати обчислювальні потужності в разі збільшення обсягів даних або розширення функціональності системи. Крім того, AWS EC2 надає інструменти для моніторингу, управління безпекою та резервного копіювання, що підвищує загальну надійність рішення.

Аналіз альтернативних технологій та архітектур показав, що обрані рішення є найбільш оптимальними для досягнення поставленої мети. Зокрема, використання сучасних мов програмування Swift і Python дозволяє знизити витрати на розробку, забезпечуючи при цьому високу якість програмного забезпечення. Flask надає гнучкість і простоту у впровадженні серверної частини, а платформа AWS EC2 забезпечує інфраструктурну підтримку для реалізації системи в реальних умовах [19].

Таким чином, вибір архітектури та технологій для розробленої системи є обґрунтованим, відповідає вимогам до продуктивності, надійності, безпеки та зручності використання, а також створює міцну основу для подальшого розвитку та масштабування системи.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ МЕРЧАНДАЙЗИНГУ

Наступним важливим етапом у розробці інформаційної системи є безпосередня реалізація та тестування всіх складових системи. Цей процес включає розробку мобільного застосунку, серверної частини, інтеграцію моделі комп'ютерного зору YOLOv5, а також всебічне тестування ефективності та надійності системи. У даному розділі буде детально описано кожен з етапів цього процесу, що дозволить краще зрозуміти особливості реалізації проекту.

Процес реалізації був організований відповідно до вимог, визначених на попередніх етапах розробки. Першим кроком стало створення мобільного застосунку для платформи iOS. Для цього була обрана мова програмування Swift та фреймворк UIKit, оскільки вони дозволяють швидко розробити інтуїтивно зрозумілий та зручний інтерфейс для кінцевих користувачів — мерчендайзерів. Особливу увагу було приділено забезпеченню високої зручності користування додатком, щоб кінцеві користувачі могли ефективно здійснювати всі необхідні операції, такі як фотографування товарів та перегляд результатів обробки зображень.

Паралельно з розробкою мобільного застосунку здійснювалась розробка серверної частини на базі мови Python та фреймворку Flask [21]. Це дозволило реалізувати потужний бекенд для прийому зображень від мобільного застосунку, їх обробку за допомогою моделі комп'ютерного зору YOLOv5 та повернення результатів користувачам. Серверна частина також відповідала за обробку запитів та збереження результатів, що є важливим для подальшої аналітики і оптимізації роботи системи.

Інтеграція моделі комп'ютерного зору YOLOv5, яка є основним компонентом для розпізнавання товарів на зображеннях, вимагала особливого підходу. Необхідно було адаптувати модель до специфічних задач мерчендайзингу, щоб забезпечити точне і швидке розпізнавання товарів на полицях. Це включало не лише налаштування самої моделі, а й коригування алгоритмів для обробки даних у реальному часі, що є критичним для ефективності системи.

Після завершення етапу розробки мобільного застосунку, серверної частини та інтеграції YOLOv5 було проведено комплексне тестування системи. Тестування було організоване як багатоетапний процес, що дозволило ретельно оцінити працездатність, стабільність та відповідність системи заданим вимогам. Першим етапом тестування була перевірка функціональності основних компонентів, що включає оцінку коректності виконання всіх функцій, від фотографування до відображення результатів. Було перевірено, як система обробляє запити, як коректно працюють алгоритми розпізнавання, а також чи немає помилок в передачі і відображенні результатів користувачам [23].

Також важливою частиною тестування став аналіз юзабіліті мобільного застосунку. Оскільки кінцеві користувачі — це мерчендайзери, інтерфейс повинен бути простим та інтуїтивно зрозумілим. Для цього проводились тести на зручність навігації, швидкість виконання завдань та зручність взаємодії з застосунком. Результати тестування дозволили виявити потенційні проблеми в інтерфейсі та зробити необхідні корективи.

Оцінка продуктивності системи включала вимірювання часу обробки зображень, затримок в передачі даних та загальної стабільності роботи під різним навантаженням. Для цього використовувались різноманітні навантажувальні тести, що дозволили оцінити, як система справляється з великим обсягом запитів та чи не виникають збої при високих навантаженнях.

Результати тестування були ретельно задокументовані, що дозволило виявити недоліки системи та визначити напрямки для її вдосконалення. На основі отриманих даних було розроблено план коригування, який включав зміни до коду, алгоритмів та налаштувань моделі YOLOv5, щоб підвищити точність та швидкість обробки зображень. Це дозволило значно покращити роботу системи та досягти високої точності у реальному застосуванні.

Цей розділ надає повний опис кожного етапу розробки та тестування, а також детальний аналіз отриманих результатів. У результаті тестування були отримані кількісні показники ефективності системи, які дозволяють оцінити її практичну цінність та можливості для подальшого використання в автоматизації процесів мерчендайзингу. Також в розділі представлена інформація про використані інструменти, бібліотеки та методи тестування, що

дозволяє читачеві об'єктивно оцінити всі аспекти розробленої системи.

3.1 Розробка мобільного застосунку

Розробка мобільного застосунку для платформи iOS є ключовим етапом у створенні інформаційної системи автоматизації мерчендайзингу, оскільки саме цей застосунок забезпечує безпосередню взаємодію користувачів із системою. Мобільний застосунок виконує роль клієнта, що відповідає за збір даних у вигляді фотографій полиць із товарами, передачу цих даних на сервер для подальшої обробки та відображення результатів аналізу. Важливим аспектом розробки стало забезпечення інтуїтивно зрозумілого інтерфейсу, який дозволяє користувачам легко виконувати основні дії, а також високої швидкодії та стабільності роботи застосунку, що є критично важливим для щоденного використання в реальних умовах [16].

Для реалізації мобільного застосунку було обрано мову програмування Swift та фреймворк UIKit, які є стандартними інструментами для розробки під платформу iOS. Swift, як сучасна та потужна мова програмування, поєднує високу продуктивність із простотою написання коду, що суттєво спрощує процес розробки. Крім того, Swift забезпечує додаткову безпеку за рахунок запобігання багатьом поширеним помилкам програмування, що сприяє стабільності роботи застосунку. Використання UIKit дозволило створити зручний, адаптивний і функціональний графічний інтерфейс, завдяки широкому набору готових компонентів, які відповідають стандартам дизайну iOS.

Процес розробки мобільного застосунку був організований у кілька етапів, серед яких одним із ключових стало створення інтерфейсу користувача (UI). Інтерфейс було розроблено з урахуванням специфіки роботи мерчендайзерів, з акцентом на простоту та інтуїтивність. Центральним елементом інтерфейсу став екран камери, який надає можливість швидко та зручно фотографувати полиці з товарами (рисунки 3.1). Екран керування камерою містить основні елементи, необхідні для виконання задач користувачів: кнопку для активації або вимкнення спалаху, елементи керування зумом, кнопку виходу з екрану камери, а також кнопку створення фотографії.

Цей підхід дозволив забезпечити максимальну зручність для користувачів, мінімізуючи час, необхідний для виконання завдань.

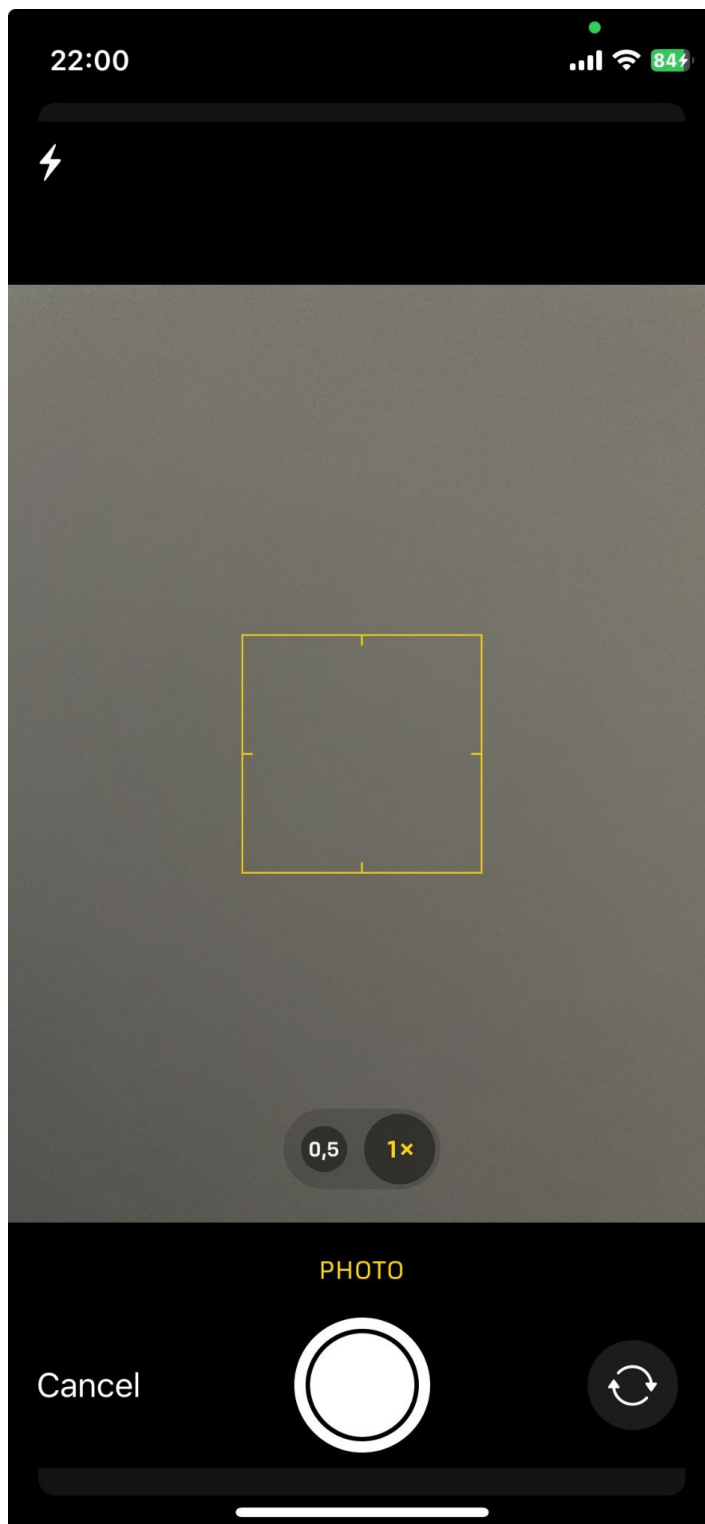


Рисунок 3.1 – Інтерфейс екрану камери

Після того, як знімок було зроблено, користувач отримує можливість вибору подальших дій. Він може надіслати фото на аналіз для обробки системою, або, у разі потреби, зробити знімок повторно, якщо попередній

результат не відповідає очікуванням чи вимогам (рисунок 3.2). Такий підхід дозволяє забезпечити гнучкість і зручність у роботі з застосунком, надаючи користувачеві можливість самостійно контролювати якість зроблених фотографій.

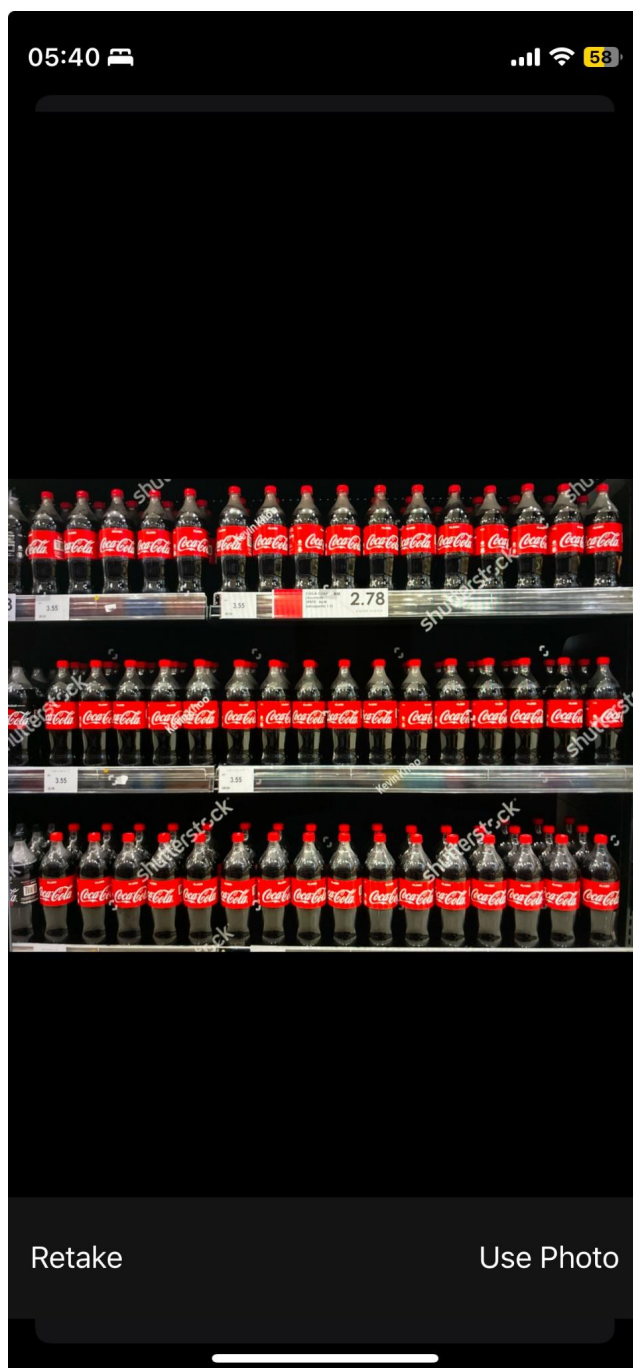


Рисунок 3.2 – Екран перегляду фото

Важливою частиною користувацького інтерфейсу є екран відображення результатів розпізнавання, на якому чітко представлено інформацію про кількість розпізнаних товарів. Цей екран забезпечує зручний і зрозумілий

спосіб отримання результатів аналізу зображення, що дозволяє користувачеві швидко оцінити ситуацію та прийняти відповідні рішення. Приклад відображення результатів розпізнавання наведено на рисунку 3.3.



Рисунок 3.3 – Результат розпізнавання

Реалізація роботи з камерою: Для забезпечення можливості фотографування полиць, в додатку реалізовано функціонал доступу до камери

пристрою з використанням фреймворку AVFoundation. Цей функціонал дозволяє отримувати зображення з камери, а також налаштовувати параметри зйомки, такі як фокус, експозиція та інші (див. лістинг 3.1).

Лістинг 3.1 – Програмний код для роботи з камерою

```
struct ImagePicker: UIViewControllerRepresentable {
    @Binding var image: UIImage?
    var sourceType: UIImagePickerController.SourceType
    func makeUIViewController(
        context: Context) -> UIImagePickerController {
        let picker = UIImagePickerController()
        picker.delegate = context.coordinator
        picker.sourceType = sourceType
        if sourceType == .camera { // Only apply scaling for
camera
            let screenSize = UIScreen.main.bounds.size
            let cameraAspectRatio = CGFloat(4.0 / 3.0)
            let screenAspectRatio = screenSize.width /
screenSize.height
            let scale = screenAspectRatio > cameraAspectRatio
                ? screenSize.height / (screenSize.width *
cameraAspectRatio)
                : screenSize.width / (screenSize.height /
cameraAspectRatio)
            picker.modalPresentationStyle = .fullScreen
        }
        return picker
    }

    func updateUIViewController(
        _ viewController: UIImagePickerController,
        context: Context,
    ) {
        // No updates needed
    }

    func makeCoordinator() -> Coordinator {
        Coordinator(self)
    }
}

class Coordinator: NSObject, UINavigationControllerDelegate,
UIImagePickerControllerDelegate {
    let parent: ImagePicker

    init(_ parent: ImagePicker) {
        self.parent = parent
    }

    func imagePickerController(
        _ picker: UIImagePickerController,
        didFinishPickingMediaWithInfo:
        [UIImagePickerController.InfoKey: Any]) {
```

```

        if let uiImage = info[.originalImage] as? UIImage {
            parent.image = uiImage
        }
        picker.dismiss(animated: true)
    }
}

```

Логіка передачі даних на сервер: Після отримання зображення з камери, застосунок має відправити його на сервер для подальшої обробки. Для цього використовувався HTTP-запит з методом POST, де зображення надсиалося у форматі multipart/form-data.

Важливим аспектом розробки мобільного застосунку є оптимізація процесу передачі зображень на сервер, що включає зменшення їх розміру перед відправкою. Оскільки необроблені зображення з камери мобільного пристрою можуть мати великий розмір, їх передача може займати значний час та споживати велику кількість трафіку, особливо при нестабільному інтернет-з'єднанні. Для вирішення цієї проблеми в мобільному застосунку було реалізовано функціонал зменшення розміру зображення перед його відправкою на сервер.

Для зменшення розміру зображення використовувався метод масштабування (resizing) з використанням фреймворку UIKit. Перед масштабуванням зображення стискається за допомогою JPEG-компресії, що дозволяє зменшити розмір файлу без значної втрати якості. Для масштабування використовується алгоритм, що забезпечує пропорційну зміну розмірів зображення, щоби уникнути його спотворення. При цьому задається максимальний розмір зображення, наприклад, ширина 800 пікселів, а висота пропорційно підлаштовується (див. лістинг 3.2).

Лістинг 3.2 – Програмний код для оптимізації розміру зображень

```

func resizeImage(image: UIImage, targetSize: CGSize) -> UIImage? {
    let size = image.size
    let widthRatio  = targetSize.width  / size.width
    let heightRatio = targetSize.height / size.height
    var newSize: CGSize
    if(widthRatio > heightRatio) {
        newSize = CGSize(
            width: size.width * heightRatio,

```

```

        height: size.height * heightRatio,
    )
} else {
    newSize = CGSize(
        width: size.width * widthRatio,
        height: size.height * widthRatio,
    )
}

let rect = CGRect(
    x: 0, y: 0,
    width: newSize.width,
    height: newSize.height,
)
UIGraphicsBeginImageContextWithOptions(newSize, false, 1.0)
image.draw(in: rect)
let newImage = UIGraphicsGetImageFromCurrentImageContext()
UIGraphicsEndImageContext()

return newImage
}

```

Реалізація процесу зменшення зображення виконується в декілька етапів:

- **Отримання зображення з камери:** Після того, як мерчендайзер робить фотографію за допомогою камери пристрою, зображення зберігається в оперативній пам'яті у вигляді об'єкта UIImage.
- **Стискання зображення:** Отримане зображення стискається за допомогою JPEG-компресії з заданим рівнем якості. Це дозволяє зменшити розмір файлу без значної втрати якості. Рівень якості стискання може варіюватися залежно від налаштувань, але для забезпечення балансу між розміром та якістю зазвичай використовується значення в діапазоні 0.5-0.7.
- **Масштабування зображення:** Стиснуте зображення масштабується до заданого максимального розміру. При цьому зберігаються пропорції зображення, щоби уникнути спотворення. (Можна додати фрагмент коду Swift, що ілюструє процес стискання та масштабування зображення).
- **Передача зменшеного зображення:** Зменшене зображення передається на сервер за допомогою HTTP-запиту, як було описано раніше.

Використання процесу зменшення зображення дозволило значно зменшити час передачі даних на сервер та споживання трафіку, особливо при використанні мобільного інтернету. Це сприяє покращенню продуктивності та зручності використання мобільного застосунку для мерчендайзерів.

Логіка отримання та відображення результатів: Після надсилання зображення на сервер, застосунок очікує на відповідь з результатами обробки, що містить інформацію про кількість розпізнаних товарів. Отримавши відповідь, застосунок відображає результати на екрані користувача. Для цього використовується UIKit, що дозволяє динамічно оновлювати інтерфейс та відображати отриману інформацію.

Обробка помилок: Застосунок має обробляти можливі помилки, такі як помилки мережі або помилки сервера, і виводити повідомлення про них користувачу.

Процес розробки мобільного застосунку також включав в себе тестування кожного етапу, що дозволило виявити недоліки та виправити їх на ранніх стадіях розробки [24]. мобільний застосунок є важливим компонентом розробленої системи, і його зручність, надійність та функціональність є ключовими факторами успішного впровадження системи в цілому.

3.2 Розробка серверної частини

Розробка серверної частини є критично важливим етапом у створенні інформаційної системи автоматизації мерчендайзингу, оскільки саме вона відповідає за обробку зображень, розпізнавання товарів, обчислення їх кількості та повернення результатів мобільному застосунку. Серверна частина виступає в ролі ядра системи, що забезпечує обробку даних, інтеграцію з моделлю комп'ютерного зору та взаємодію з базою даних, якщо така використовується. Процес розробки був організований з урахуванням вимог до продуктивності, надійності, масштабованості та безпеки

Для розробки серверної частини було обрано мову програмування Python та фреймворк Flask, що є популярними інструментами для створення веб-додатків. Python є простою та гнучкою мовою програмування з великою кількістю бібліотек, а Flask є легким та гнучким фреймворком, що дозволяє швидко розробляти веб-сервіси. Процес розробки включав в себе наступні етапи:

Реалізація API для взаємодії з мобільним застосунком. Серверна частина надає API, що дозволяє мобільному застосунку надсилати зображення та отримувати результати обробки. API реалізовано з використанням HTTP-запитів з методом POST для надсилання зображень та методом GET для отримання результатів. При реалізації API враховано необхідність обробки помилок та забезпечення безпечного обміну даними між мобільним застосунком та сервером [22].

Логіка отримання та обробки зображень. Серверна частина приймає зображення, відправлені мобільним застосунком, та обробляє їх для подальшого розпізнавання (див. лістинг 3.3).

Лістинг 3.3 – Програмний код для обробки зображень

```
@app.route('/detect', methods=['POST'])
def detect():
    if 'file' not in request.files:
        return jsonify({'error': 'No file part'}), 400

    file = request.files['file']

    if file.filename == '':
        return jsonify({'error': 'No selected file'}), 400

    # Read the uploaded image as a numpy array
    img_array = np.frombuffer(file.read(), np.uint8)
    image = cv2.imdecode(img_array, cv2.IMREAD_COLOR)

    if image is None:
        return jsonify({'error': 'Invalid image file'}), 400

    # Detect and count objects
    object_count = detect_and_count(image)

    # Return the result as JSON
    return jsonify({'object_count': object_count})
```

Логіка виклику моделі YOLOv5. Для ефективного розпізнавання товарів на зображенні серверна частина взаємодіє з моделлю комп'ютерного зору YOLOv5. Модель отримує зображення, передане сервером, та виконує процес розпізнавання, визначаючи координати кожного виявленого об'єкта, а також його клас (тип товару). Цей процес дозволяє точно ідентифікувати товари на зображенні з високою точністю.

Логіка підрахунку та повернення результатів. Після того як модель YOLOv5 повертає координати та класи розпізнаних об'єктів, серверна частина обробляє ці дані, підраховуючи кількість товарів кожного типу. Далі формує відповідь, яка надсилається в мобільний застосунок. Ця відповідь містить загальну кількість розпізнаних товарів. Приклад формату такої відповіді наведено в лістингу 3.4.

Лістинг 3.4 – Відповідь від сервера

```
{  
  "object_count": 69  
}
```

Обробка помилок. Серверна частина має обробляти можливі помилки, такі як помилки обробки зображень або помилки зв'язку з моделлю YOLOv5, і повертати відповідні повідомлення про помилки.

Розгортання серверної частини на AWS EC2. Після завершення розробки серверна частина була розгорнута на хмарному сервісі AWS EC2. Це забезпечує надійну та стабільну роботу сервера, а також можливість масштабування при зростанні навантаження. AWS EC2 надає необхідні обчислювальні ресурси для ефективної роботи серверної частини [19].

Процес розробки серверної частини також включав в себе тестування кожного етапу, що дозволило виявити недоліки та виправити їх на ранніх стадіях розробки. Серверна частина є важливим компонентом розробленої системи, і її продуктивність, надійність та масштабованість є ключовими факторами успішної роботи системи в цілому.

3.3 Тестування системи

Тестування розробленої системи автоматизації мерчендайзингу є важливим етапом у циклі її створення, який спрямований на оцінку її працездатності, надійності та відповідності встановленим вимогам. У процесі тестування було реалізовано кілька типів перевірок, кожна з яких мала на меті

забезпечити високий рівень якості кінцевого продукту. Зокрема, функціональне тестування дозволило оцінити відповідність системи заданим функціональним вимогам, перевіривши її здатність коректно виконувати всі основні операції.

Тестування зручності використання проводилося для забезпечення інтуїтивності інтерфейсу, що є критично важливим для ефективної роботи користувачів із системою. Продуктивність перевірялася за допомогою аналізу швидкості обробки зображень, передачі даних та загальної стабільності роботи під навантаженням. Кожен етап тестування мав на меті виявити недоліки системи, усунути їх і забезпечити найвищу якість продукту перед його впровадженням у реальне використання.

Функціональне тестування. Цей вид тестування був спрямований на перевірку правильності роботи основних функцій системи. Зокрема, тестувалася функціональність мобільного застосунку, що включала в себе фотографування полиць з товарами, відправку зображень на сервер та отримання результатів обробки. Також перевірялася працездатність API серверної частини, а саме можливість прийому зображень, виклику моделі YOLOv5 та повернення результатів розпізнавання. Результати функціонального тестування показали, що основні функції системи працюють коректно та відповідають вимогам.

Юзабіліті-тестування. Цей вид тестування був спрямований на оцінку зручності та інтуїтивності інтерфейсу мобільного застосунку. В тестуванні приймали участь потенційні користувачі системи - мерчендайзери. Тестувальникам було запропоновано виконати ряд типових завдань, таких як фотографування полиць, надсилання зображень на сервер та перегляд результатів розпізнавання. В процесі тестування оцінювалася легкість навігації, зрозумілість елементів інтерфейсу та загальна зручність використання застосунку. Юзабіліті-тестування дозволило виявити деякі недоліки інтерфейсу та внести необхідні покращення [25].

Тестування продуктивності мало на меті всебічну оцінку швидкості та ефективності роботи системи, включаючи аналіз її здатності функціонувати стабільно за різних умов навантаження. Основна увага приділялася вимірюванню швидкості передачі зображень з мобільного застосунку на сервер,

тривалості обробки зображень моделлю комп'ютерного зору YOLOv5 та часу повернення результатів користувачам. Окрім цього, проводилося навантажувальне тестування, під час якого імітувалася одночасна робота великої кількості користувачів, щоб перевірити масштабованість системи. Результати тестування подані в таблиці 3.1.

Таблиця 3.1 – Результати навантажувального тестування

Запитів/хв	Загальний час (с)	Обробка на сервері (с)	Обробка моделлю (с)
10	0.8	0.7	0.6
25	1.0	0.9	0.7
50	1.3	1.2	0.8
75	1.7	1.6	0.8
100	1.9	1.8	0.8

Результати тестування засвідчили, що система демонструє високу швидкість обробки, стабільність у роботі та здатність витримувати значні навантаження, що підтверджує її відповідність вимогам до продуктивності та надійності.

3.4 Висновки до третього розділу

У цьому розділі було детально описано процес реалізації та тестування інформаційної системи автоматизації мерчендайзингу з використанням комп'ютерного зору. Розробка системи передбачала створення мобільного застосунку для платформи iOS, серверної частини на основі Python та Flask, а також інтеграцію моделі YOLOv5 для розпізнавання об'єктів на зображеннях. Розроблений мобільний застосунок надає зручний та інтуїтивно зрозумілий інтерфейс для мерчендайзерів, що дозволяє їм швидко робити фотографії товарних полиць та відправляти їх на сервер для подальшої обробки. Серверна частина, розміщена на хмарній платформі AWS EC2, відповідає за ефективну обробку зображень, визначення товарів та підрахунок їх кількості.

У процесі розробки особлива увага приділялася оптимізації продуктивності та надійності системи. Одним із ключових аспектів стала реалізація механізмів стиснення зображень перед їх відправкою на сервер. Це дозволило зменшити навантаження на мережу та значно пришвидшити обробку зображень. Для інтеграції моделі YOLOv5 у серверну частину враховувалась необхідність забезпечення обробки зображень у реальному часі, що вимагало високої продуктивності та масштабованості серверної частини. Завдяки розгортанню на AWS EC2 вдалося досягти стабільності та надійності системи навіть при великому обсязі запитів.

Після розробки та інтеграції системи було проведено комплексне тестування, яке охоплювало кілька основних етапів. Функціональне тестування підтвердило коректну роботу основних компонентів системи. Були перевірені такі функції, як фотографування товарних полиць, відправка зображень на сервер, обробка зображень, точність розпізнавання товарів, а також підрахунок їх кількості. Юзабіліті-тестування дозволило виявити деякі недоліки в інтерфейсі мобільного застосунку, що не дозволяли користувачам працювати максимально зручно. Після проведення тестувань було внесено необхідні покращення для забезпечення більш зручного використання додатку [26].

Тестування продуктивності продемонструвало, що система здатна ефективно обробляти зображення без значних затримок, навіть при високому навантаженні. Це важливо для забезпечення своєчасної обробки даних та збереження високої швидкості роботи системи в умовах великої кількості запитів.

Загалом результати реалізації та тестування продемонстрували високу ефективність створеної системи. Вона здатна автоматизувати процеси мерчендайзингу та значно зменшити час, витрачений на збір даних про наявність товарів на полицях. Крім того, система підвищує точність отриманих даних, що дозволяє забезпечити більш точну інформацію для подальшого аналізу та ухвалення рішень. мобільний застосунок зроблений максимально зручним, що дозволяє мерчендайзерам швидко та ефективно виконувати завдання. Обробка зображень на сервері за допомогою моделі YOLOv5 гарантує високу точність розпізнавання та підрахунку товарів.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

4.1.1 Шкідливі та небезпечні фактори при користуванні персональними комп'ютерами та смартфонами

Одним з найважливіших елементів будь-якого підприємства є робоче місце, де конкретний працівник виконує цілеспрямовану діяльність (тобто роботу). Робоче місце - це частина виробничого простору одного або групи працівників, оснащена основним і допоміжним технологічним обладнанням, інвентарем, інструментом, робочими меблями, необхідними для виконання певного виду робіт. З розвитком виробничих процесів та інформаційних технологій робочі місця мерчендайзерів все більше оснащуються електронними пристроями. Однак їх використання загостило проблеми зі здоров'ям. Це вимагає вдосконалення існуючих і розробки нових підходів до організації робочого місця, профілактичних заходів для запобігання розвитку негативних наслідків для здоров'я працівників мерчендайзингової компанії.

Враховуючи той факт, що робота мерчендайзера пов'язана з використанням комп'ютерної техніки, необхідно створити безпечні умови праці при використанні ПК та смартфонів. Особливу увагу слід також приділяти електробезпеці.

З урахуванням вимог ДСанПіН 3.3.2.007-98 та НПАОП 0.00-7.15-18 необхідно виявити небезпечні та шкідливі фактори, які впливають на користувачів ПК та смартфонів під час роботи, вивчити ці фактори, розглянути їх вплив, принципи їх нормування та шляхи запобігання їх шкідливому впливу на людину [32].

Основними шкідливими та небезпечними виробничими факторами, пов'язаними з використанням ПК та смартфонів, є наступні:

- електромагнітне випромінювання радіочастотного діапазону;
- випромінювання оптичного діапазону (ультрафіолетове,
- інфрачервоне і випромінювання видимого діапазону);

- електростатичне поле;
- недостатня освітленість робочої зони;
- значна напруга зорових органів і пов'язане з цим перевтомлення користувача ПК;
- можливість ураження електричним струмом;

Відповідно до ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин», площа приміщень, де розміщуються відеотермінали, визначається з розрахунку на одне робоче місце — не менше 6,0 квадратних метрів, об'єм — не менше 20,0 кубічних метрів, з урахуванням максимальної кількості осіб, які одночасно працюють у зміні. Це забезпечує комфортні умови для роботи, дозволяючи мінімізувати негативний вплив на здоров'я працівників при тривалому використанні візуальних дисплейних терміналів.

Загальне освітлення приміщення повинно бути організоване так, щоб максимально відповідати вимогам щодо комфорту та безпеки. Для цього слід передбачити розміщення безперервних або переривчастих ліній світильників, які розташовуються збоку від робочого місця, зазвичай зліва, і паралельно лінії зору працівників. Це дозволяє уникнути прямого потрапляння світла в очі та забезпечує рівномірне освітлення робочої зони. Як джерело світла для штучного освітлення рекомендується використовувати люмінесцентні лампи типу ЛБ, які забезпечують м'яке, приємне для очей освітлення без додаткових теплових втрат. Водночас допускається використання ламп розжарювання в світильниках місцевого освітлення для конкретних завдань, таких як підсвічування окремих ділянок робочого місця.

Рівень освітленості на робочому столі в зоні розташування документів має бути в межах 300-500 лк. Це гарантує достатню яскравість для комфортного читання та роботи з документами. Якщо загальна система освітлення не здатна забезпечити такий рівень освітленості, допускається використання додаткових ламп місцевого освітлення. Проте важливо, щоб при цьому не було відблисків на поверхні екрана, а рівень освітленості екрана не перевищував 300 лк, оскільки це може призвести до перевантаження зору і викликати дискомфорт у працівників [33].

При дотриманні цих вимог, виконання всіх видів робіт на персональних комп'ютерах та смартфонах у сфері мерчандайзингу є безпечним з точки зору охорони праці. Така організація робочих місць не тільки підвищує ефективність роботи, але й значно знижує ризики для здоров'я працівників, забезпечуючи комфортні та безпечні умови для тривалої роботи з візуальними дисплейними терміналами.

4.2 Безпека в надзвичайних ситуаціях

4.2.1 Підвищення стійкості роботи об'єктів господарської діяльності у воєнний час

Підвищення стійкості об'єктів господарської діяльності, в тому числі підприємств роздрібної торгівлі, в яких працюють представники мерчандайзингових компаній, у воєнний час є одним з головних завдань цивільного захисту України.

Стійкість бізнесу визначається як здатність об'єкта господарської діяльності реалізовувати продукцію в запланованих обсягах в умовах надзвичайної ситуації, а також відновлювати виробництво в найкоротші терміни у разі незначних або помірних пошкоджень чи перебоїв у постачанні товарів.

Для того, щоб забезпечити нормальну роботу об'єктів господарської діяльності промислових підприємств та зменшити можливі матеріальні втрати у воєнний час, необхідно здійснити комплекс різноманітних заходів, які б забезпечили їхнє функціонування [34]. Стабільна робота суб'єкта господарювання залежить від захисту та нормального функціонування його основних елементів:

- Виробничий персонал.
- Будинки та споруди з технологічним устаткуванням.
- Система постачання енергією, водою, паливом, устаткуванням та ремонтною базою.
- Система виробничих і кооперативних зв'язків з іншими об'єктами.

На стійкість роботи об'єктів господарської діяльності впливають наступні фактори:

- Надійність захисту робітників від дії вражаючих факторів.
- Здатність промислового комплексу протистояти дії вражаючих факторів.
- Стійкість системи управління виробництвом та цивільною обороною в надзвичайних ситуаціях.
- Готовність об'єкта до проведення рятувальних дій або робіт по відновленню виробництва.
- Захищеність об'єкта від дії вторинних вражаючих факторів.

При вирішенні проблеми підвищення стійкості об'єктів господарської діяльності, як і інших об'єктів національної економіки, застосовуються ті ж самі фундаментальні положення:

- Завчасне проведення заходів цивільного захисту, спрямованих на зниження можливих втрат та руйнувань у разі застосування з боку противника зброї масового ураження і на створення умов для швидкого відновлення виробництва після часткового руйнування.
- Комплексний підхід в розробці і здійсненні заходів забезпечення стійкості роботи для всіх напрямків діяльності підприємства.
- Узгодження заходів з територіальними і військовими органами управління.

Заходи з підвищення стійкості планують з урахуванням місцевих умов, ступеня важливості об'єкта, його географічного розташування та економічної доцільності заходів. Крім того, при реалізації заходів цивільного захисту слід враховувати внутрішні фактори, що впливають на стійкість:

- розмір виробництва;
- чисельність працівників;
- рівень їх дисциплінованості і компетентності;
- системи постачання підприємства технічною і питною водою;

Існують наступні шляхи і способи підвищення стійкості роботи підприємств приладобудівної галузі:

- Забезпечення надійного захисту робітників і службовців: укриття робітників і службовців, які продовжують роботу на об'єкті у воєнний час.
- Проведення евакуації робітників, службовців і членів їх сімей та забезпечення їх життєдіяльності.
- Захист об'єкта від ураження, підвищення стійкості будівель, споруд проти впливу ударної хвилі чи світлового випромінювання.
- Укриття найбільш уразливого обладнання в захисних пристроях (шатрах, камерах, конусах).
- Підготовка паливно-енергетичного господарства до роботи у воєнний час.
- Розосередження запасів найбільш уразливого обладнання, приладів, сировини.
- Встановлення виробничих контактів з дублерами постачальниками, необхідних для безперебійної роботи об'єкта.
- Проведення підземних кабельних ліній зв'язку до всіх елементів об'єкта.
- Підготовка до виконання робіт по відновленню об'єкта у воєнний час, планування відновлювальних робіт за кількома варіантами.

Підвищення стійкості об'єкта господарської діяльності у воєнний час є надзвичайно важливим для забезпечення безперервності роботи ключових підприємств та організацій. Це досягається за рахунок ретельно спланованого комплексу інженерно-технічних, технологічних та організаційних заходів, спрямованих не лише на підготовку до можливих воєнних загроз, а й на створення умов для оперативного відновлення роботи об'єкта після їх виникнення.

Основними напрямками таких заходів є забезпечення захисту від впливу зброї масового ураження (в тому числі хімічного, біологічного, радіаційного та термічного впливу), мінімізація можливих пошкоджень інфраструктури, створення резервів ресурсів та технічних засобів для швидкого реагування на наслідки у разі їх виникнення. Крім того, важливим аспектом є організація ефективної роботи в умовах обмеженого доступу до ресурсів та при можливому пошкодженні ліній зв'язку [35].

Необхідно також забезпечити готовність персоналу до дій у надзвичайних ситуаціях, зокрема шляхом проведення регулярних тренувань і навчань з ліквідації наслідків надзвичайних ситуацій. Важливим елементом забезпечення стійкості є створення планів евакуації та налагодження співпраці з органами влади та іншими підприємствами, які можуть бути залучені до надання допомоги.

Завдяки комплексному підходу до цього процесу об'єкт господарської діяльності може суттєво знизити ризики та наслідки можливих військових дій, забезпечивши стабільну та ефективну роботу навіть у найскладніших умовах.

4.3 Висновки до четвертого розділу

Було проаналізовано основні шкідливі та небезпечні фактори, пов'язані з використанням електронних пристроїв, та наведено рекомендації щодо їх мінімізації відповідно до чинних нормативних документів. Визначено заходи щодо забезпечення ергономіки робочих місць, оптимального освітлення та електробезпеки, що сприяють зниженню ризиків для здоров'я працівників і підвищенню продуктивності їхньої роботи.

Окремо розглянуто безпеку в надзвичайних ситуаціях, зокрема підвищення стійкості об'єктів господарської діяльності у воєнний час. Визначено основні фактори, що впливають на стабільність роботи підприємств, та запропоновано комплекс організаційно-технічних заходів для забезпечення безперервності функціонування та оперативного відновлення роботи об'єктів у разі виникнення надзвичайних ситуацій. Реалізація зазначених заходів дозволяє мінімізувати втрати та ризики, забезпечуючи стабільну роботу підприємств навіть у найскладніших умовах [36].

ВИСНОВКИ

У кваліфікаційній роботі було представлено дослідження, розробку та тестування інформаційної системи для автоматизації процесів мерчендайзингу з використанням технологій комп'ютерного зору. Метою роботи була розробка системи, що дозволить підвищити ефективність роботи мерчендайзерів та якість збору даних про товари на полицях магазинів. В процесі дослідження було проведено аналіз існуючих рішень та методів комп'ютерного зору, обґрунтовано вибір архітектури системи та технологій, розроблено мобільний додаток та серверна частина, проведено тестування системи, визначено перспективи подальшого розвитку та оцінено вплив на процес мерчендайзингу.

В результаті виконання дипломної роботи було досягнуто поставленої мети та виконано наступні завдання:

- Проведено аналіз існуючих рішень та методів комп'ютерного зору для розпізнавання та підрахунку об'єктів на зображеннях. Було вивчено різні підходи до розпізнавання об'єктів, зокрема традиційні методи та сучасні методи на основі глибокого навчання.
- Обґрунтовано вибір архітектури та технологічного стеку для розробки системи, включаючи вибір моделі комп'ютерного зору (YOLOv5), фреймворку для серверної частини (Flask) та хмарного провайдера (AWS EC2). Було продемонстровано, чому обрані технології є оптимальними для вирішення поставлених завдань.
- Розроблено мобільний застосунок для iOS, що дозволяє робити фотографії полиць з товарами та надсилати їх на сервер для обробки. Додаток забезпечує зручний та інтуїтивно зрозумілий інтерфейс для користувача.
- Розроблено серверну частину, що отримує зображення, обробляє їх за допомогою моделі YOLOv5 та повертає кількість розпізнаних товарів. Серверна частина ефективно обробляє дані та забезпечує необхідну швидкість розпізнавання.
- Проведено тестування розробленої системи, включаючи оцінку швидкодії та стабільності роботи. Тестування підтвердило стабільну роботу

системи та її здатність ефективно обробляти зображення, а також дозволило виявити та усунути деякі недоліки.

- Визначено перспективи подальшого розвитку та вдосконалення системи, враховуючи можливості розпізнавання брендів, аналізу цінників та інтеграції з іншими системами. Запропоновано напрямки для подальшого розвитку системи.

- Оцінено вплив розробленої системи на процес мерчендайзингу та ефективність роботи мерчендайзерів. Зроблено висновок, що розроблена система сприяє автоматизації процесу збору даних, підвищує точність обліку товарів та дозволяє мерчендайзерам більш ефективно виконувати свою роботу.

Результати дослідження підтверджують, що використання технологій комп'ютерного зору є перспективним напрямком для автоматизації процесу мерчендайзингу. Розроблена система може бути ефективно впроваджена в роздрібних мережах, що дозволить підвищити ефективність роботи мерчендайзерів, оптимізувати витрати та покращити якість обслуговування клієнтів.

СПИСОК ДЖЕРЕЛ

1. Business process management. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Business_process_management
2. П. Панцир, "Імплементация цифрових інструментів в структуру бізнес- процесів організації," in Інформаційні системи та технології: матеріали VII наук.- техн. конф., Тернопіль, Україна, 2019, p. 81.
3. J. Mitric, I. Radulovic, T. Popovic, Z. Scekcic and S. Tinaj, "AI and Computer Vision" in 2024 28th International Conference on Information Technology (IT), Zabljak, Montenegro, 2024, pp. 1-4, doi: 10.1109/IT61232.2024.10475738.
4. Amazon Web Services - Wikipedia. URL: https://uk.wikipedia.org/wiki/Amazon_Web_Services (дата звернення: 02.02.2024)
5. A. Kharchenko, I. Bodnarchuk, and V. Yatcysyn, "The Method for Comparative Evaluation of Software Architecture with Accounting of Trade-offs," American Journal of Information Systems, vol. 2, no. 1, pp. 20-25, 2014.
6. I. Strutynska, H. Kozbur, L. Dmytrotsa, O. Sorokivska, and L. Melnyk, "Influence of Digital Technology on Roadmap Development for Digital Business Transformation" in Advanced Computer Information Technologies, ACIT 2019 : materials of the 9th International Conference, Ceske Budejovice, Czech Republic, 2019, pp. 333–337.
7. А. Г. Микитишин, М. М. Митник, П. Д. Стухляк, and В. В. Пасічник, Комп'ютерні мережі. Книга 1 [навчальний посібник]. Львів: «Магнолія 2006», 2013.
8. А. Г. Микитишин, М. М. Митник, П. Д. Стухляк, and В. В. Пасічник, Комп'ютерні мережі. Книга 2. [навчальний посібник]. Львів: "Магнолія 2006", 2014.
9. О. Харченко and В. Яцишин, "Розробка та керування вимогами до програмного забезпечення на основі моделі якості," Вісник ТДТУ, vol. 14, no. 1, pp. 201-207, 2009.

10. С. Шпилик, "Інтернет як ефективний маркетинговий інструмент сучасного підприємства," Галицький економічний вісник, no. 2, pp. 212–223, 2015.
11. A. Kharchenko, I. Raichev, I. Bodnarchuk, and O. Matsiuk, "The Survey of Global Software Design Processes," in 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology, PIC S and T 2021-Proceedings, 2021, pp. 291-294.
12. М. Р. Петрик and О. Ю. Петрик, Моделювання програмного забезпечення : наук.-метод. посіб. Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2015.
13. О. Гузеляк, Ю. Шевчук, Б. М. Береженко, and І. О. Боднарчук, "Програмна архітектура в розподілених командах гнучких проєктів," in Матеріали Х науково-технічної конференції „Інформаційні моделі, системи та технології “Тернопільського національного технічного університету імені Івана Пулюя, 2022, pp. 110-112.
14. В. Волович, Б. М. Береженко, and І. О. Боднарчук, "Задача проєктування програмної архітектури в процесах забезпечення якості," in Матеріали Х науково-технічної конференції „Інформаційні моделі, системи та технології “Тернопільського національного технічного університету імені Івана Пулюя, 2022, pp. 104-106.
15. В. П. Бугай and І. О. Боднарчук, "Оцінювання програмної архітектури при гнучких методах розробки програмних систем," in Матеріали Міжнародної науково-технічної конференції „Фундаментальні та прикладні проблеми сучасних технологій “до 60-річчя з дня заснування Тернопільського національного технічного університету імені Івана Пулюя та 175-річчя з дня народження Івана Пулюя, 2020, pp. 152-153.
16. I. Bodnarchuk et al., "Adaptive method for assessment and selection of software architecture in flexible techniques of design," in 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), vol. 1, 2018.

17. S. Easterbrook, J. Singer, M. A. Storey, and D. Damian, "Selecting empirical methods for software engineering research," in *Guide to advanced empirical software engineering*, 2008, pp. 285-311.
18. K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *12th international conference on evaluation and assessment in software engineering (EASE)*, 2008.
19. L. Wieclaw, V. Pasichnyk, N. Kunanets, O. Duda, O. Matsuik, and P. Falat, "Cloud computing technologies in “smart city” projects," in *Proceedings of the 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Bucharest, Romania, 2017, pp. 339–342.
20. О. Васильєв, Програмування мовою Python, В.: Навчальна книга - Бог- дан, 2019.
21. П. Ендрюс, Програмування на Python: від початківця до професіонала, М.: Видавництво "Просвітництво", 2018.
22. K. R. Srinath, "Python–the fastest growing programming language," *International Research Journal of Engineering and Technology*, vol. 4, no. 12, pp. 354-357, 2017.
23. I. Hooda and R. S. Chhillar, "Software test process, testing types and techniques," *International Journal of Computer Applications*, vol. 111, no. 13, 2015.
24. “Тестування програмного забезпечення: типи, види та застосування” FoxmindEd, foxminded.ua/. Дата доступу 14 травня 2024.
25. P. Dhore et al., "Brief Review On Different Manual Software Testing Approaches & Procedure," *Journal of Pharmaceutical Negative Results*, pp. 455-464, 2023.
26. I. Feshchuk, “Manual Testing Guide for Beginners.” QA Madness, www.qamadness.com/manual-testing-guide-for-beginners/. Дата доступу 14 травня 2024.
27. K. Sneha and G. M. Malle, "Research on software testing techniques and software automation testing tools," in *2017 international conference on energy, communication, data analytics and soft computing (ICECDS)*, 2017.

28. A. Contan, C. Dehelean, and L. Miclea, "Test automation pyramid from theory to practice," in 2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), 2018.
29. D. Dimoski et al., "Testing RESTful APIs—Use Case: RESTful API for Solving Multidimensional Time-Independent Schrödinger Equation," 2022.
30. Д. В. Котлярчук, О. В. Романюк, "Аналіз методів тестування інтерфейсу користувача," ВНТУ, 2020.
31. "Most Important Software Test Metrics." QAwerk, qawerk.com/blog/most- important-software-test-metrics. Дата доступу 14 травня 2024.
32. В. І. Голінько, "Контроль умов праці," 2018.
33. Ю. А. Гасило et al., Охорона праці в галузі та цивільний захист: навч. посіб. Кам'янське: ДДТУ, 2017.
34. В.С. Стручок, Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання. Тернопіль: ФОП Паляниця В. А., 2022.
35. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Офіційний веб портал парламенту України. URL: <https://zakon.rada.gov.ua/rada/show/v0007282-98#Text>.
36. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. Офіційний веб портал парламенту України. URL: <https://zakon.rada.gov.ua/laws/show/z0508-18#Text>.
37. М. П. Гандзюк, Є. П. Желібо, М. О. Халімовський, Основи охорони праці, К.: Каравела, 2008.
38. Про затвердження Правил охорони праці під час експлуатації електронно-обчислювальних машин. Офіційний веб-портал парламенту України. URL: <https://zakon.rada.gov.ua/laws/show/z0293-10#Text>.
39. Г. Г. Гогіташвілі, В. М. Лапін, Основи охорони праці, К.: Знання, 2008.

40. В.І. Голінько, М.Ю. Іконніков, Я.Я. Лебедєв, Охорона праці в галузі інформаційних технологій, Д.: НГУ, 2015.
41. Природне і штучне освітлення: ДБН В.2.5-28-2006.
42. І.Я. Бедрій, В.Я. Нечай, Безпека життєдіяльності, Львів: Манголія 2006, 2007.
43. Основи охорони праці, К.: Основа, 2006.
44. Вимоги безпеки до робочого місця. Ергономічні вимоги. Pidru4niki. URL:
https://pidru4niki.com/16850303/bzhd/vimogi_bezpeki_robochogo_mistsya_ergonomic_hni_vimogi.
45. К. Н. Ткачук, В. В. Зацарний, Д. В. Зеркалов, О. І. Полукаров, В. С. Коз'яков, Л. О. Мітюк, Ю. О. Полукаров, and Т. Є. Луц, Основи охорони праці, К.: Основа, 2014.
46. В.І. Шевченко, Правові питання охорони праці, Харків: ХНАМГ, 2004.
47. Закон України «Про охорону праці» від 21.11.2002р. №229-IV.
48. В.С. Стручок, Методичні вказівки до практичного заняття і самостійної роботи з курсу «Техноекологія та цивільна безпека» частина «Цивільна безпека» на тему «Шляхи і способи підвищення стійкості роботи промислового об'єкта», Тернопіль: ТНТУ імені Івана Пулюя, 2023.
49. В. Г. Грибан, Охорона праці, Київ: Центр учбової літератури, 2009.
50. Г.П. Демиденко, Безпека життєдіяльності, К.: НТУУ «КШ», 2008.

ДОДАТКИ

Тези конференції

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
імені ІВАНА ПУЛЮЯ
НАУКОВЕ ТОВАРИСТВО ім. ШЕВЧЕНКА

МАТЕРІАЛИ

ХІІ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»



18-19 грудня 2024 року

ТЕРНОПІЛЬ
2024

В. Федорієнко, О. Жук

ПІДХІД ДО МОНІТОРИНГУ ІНФОРМАЦІЙНОГО ПРОСТОРУ В ІНТЕРЕСАХ СИЛ
ОБОРОНИ УКРАЇНИ

92

В. Р. Цвігун, М. В. Деркач,

АНАЛІЗ БЕЗПЕКИ МОБІЛЬНИХ ДОДАТКІВ ЗА ДОПОМОГОЮ MOBSF

V. R. Tsvihun, M. V. Derkach

SECURITY ANALYSIS OF MOBILE APPLICATIONS USED BY MOBSF

93

В. Фецак, Є. Тиш

ДОСЛІДЖЕННЯ МЕТОДІВ ІНТЕГРАЦІЇ СИСТЕМ АДАПТИВНОГО УПРАВЛІННЯ
МІКРОКЛІМАТОМ ТА ОПТИМІЗАЦІЇ ЕНЕРГОВИТРАТ У РОЗУМНОМУ БУДИНКУ

V. Fetsak, I. Tysh

RESEARCH OF ALGORITHMS FOR ADAPTIVE CONTROL OF MICROCLIMATE
QUALITY AND ENERGY CONSUMPTION OPTIMIZATION IN A SMART HOME

94

В. Фецак, Є. Тиш

ДОСЛІДЖЕННЯ МЕТОДІВ ІНТЕГРАЦІЇ МУЛЬТИСЕНСОРНИХ СИСТЕМ ДЛЯ
АДАПТИВНОГО МОНІТОРИНГУ ТА ОПТИМІЗАЦІЇ МІКРОКЛІМАТУ В
РОЗУМНОМУ БУДИНКУ

V. Fetsak, I. Tysh

RESEARCH OF MULTI-SENSOR SYSTEM INTEGRATION METHODS FOR
ADAPTIVE MONITORING AND MICROCLIMATE OPTIMIZATION IN A SMART
HOME

96

А. Хом'як

ВИКОРИСТАННЯ СИНТЕТИЧНИХ СИГНАЛІВ ДЛЯ ПОКРАЩЕННЯ
КЛАСИФІКАЦІЇ МЕГ СИГНАЛІВ

A. Khomiak

IMPROVEMENT OF MEG SIGNAL CLASSIFICATION VIA THE USAGE OF
SYNTHETIC SIGNALS

98

І. С. Цимбрак

РЕАЛІЗАЦІЯ КРИПТОАЛГОРИТМУ ХЕШУВАННЯ SHA256

I. S. Tsymbrak

IMPLEMENTATION OF THE SHA256 HASHING CRYPTO ALGORITHM

99

А. Б. Чекановський, К. Б. Швирло

СТРАТЕГІЇ ПОКРАЩЕННЯ ПРОДУКТИВНОСТІ РЕЛЯЦІЙНИХ СИСТЕМ
КЕРУВАННЯ БАЗАМИ ДАНИХ

A. B. Chekanovskyi, K. B. Shvyrlo

STRATEGIES FOR IMPROVING PERFORMANCE OF RELATIONAL DATABASE
MANAGEMENT SYSTEMS

100

І. О. Баран, Ю. Р. Чорна

ТИПИ АНОМАЛІЙ

I. O. Baran, Yu. R. Chorna

TYPES OF ANOMALIES

101

О. Чорновол

ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ МЕРЧАНДАЙЗИНГУ З ВИКОРИСТАННЯМ
ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ

O. Chornovol

INFORMATION SYSTEM FOR MERCHANDISING USING COMPUTER VISION
TECHNOLOGIES

102

К. Чурбаков

ВИКОРИСТАННЯ МОВИ LUA ДЛЯ РОЗШИРЕННЯ ФУНКЦІОНАЛЬНИХ
МОЖЛИВОСТЕЙ IDS/IPS

K. Churbakov

USING LUA LANGUAGE TO EXTEND THE FUNCTIONALITY OF IDS/IPS

103

УДК 658.8

О. Чорновол

(Тернопільський національний університет імені Івана Пулюя, Україна)

Інформаційна система для мерчандайзингу з використанням технологій комп'ютерного зору

UDC 658.8

O. Chornovol

Information System for Merchandising Using Computer Vision Technologies

Мерчандайзинг — це важливий елемент маркетингу, спрямований на підвищення привабливості товару на полицях магазинів та стимулювання продажів. Він передбачає раціональне розміщення продукції, щоб вона виглядала максимально помітною для покупця. У сучасному світі, де конкуренція серед брендів надзвичайно висока, якість представлення товару на полицях може стати ключовим фактором успіху [1].

Проте оцінка розміщення товару залишається складним і часозатратним завданням, яке залежить від людського фактора. Мерчендайзери повинні вручну перевіряти полиці магазинів, що забирає багато часу та не гарантує високої точності. Особливо це актуально для великих торговельних мереж, де потрібно аналізувати велику кількість магазинів.

Для вирішення цієї проблеми у проєкті була розроблена інформаційна система для автоматизації оцінки якості представлення товару. Основною функцією цієї системи є обчислення кількості одиниць товару на фотографії. Мобільний додаток, створений у рамках проєкту, дозволяє мерчендайзерам швидко отримувати дані про наявність продукції та оцінювати її представлення, а також порівнювати з конкурентами.

Для реалізації системи використовувалися сучасні технології, які забезпечують її ефективність та точність. Основними інструментами стали мови програмування Python і Swift. Python використовувався для серверної частини системи та реалізації алгоритмів обробки зображень, завдяки своїй гнучкості та багатству бібліотек. Бібліотека комп'ютерного зору OpenCV дозволила реалізувати основні етапи обробки зображень, включаючи попередню підготовку даних і фільтрацію. Для розпізнавання об'єктів застосовувалася одна з найсучасніших моделей — YOLO (You Only Look Once), яка забезпечує високу точність і швидкість виявлення товарів на зображеннях [2].

Розроблена система спрощує роботу мерчендайзерів, зменшує ймовірність помилок та дозволяє швидко приймати управлінські рішення на основі отриманих даних. Вона є перспективним інструментом для підвищення ефективності мерчандайзингу та створення конкурентної переваги для брендів.

Література.

1. Котлер Ф., Армстронг Г. “Принципи маркетингу” 2021 - 101с.
2. Szeliski, R. “Computer Vision: Algorithms and Applications” 2010 - 69с.

УДК 658.8

О. Чорновол

(Тернопільський національний університет імені Івана Пулюя, Україна)

Хмарні технології як основа для масштабування інтелектуальних систем у ритейлі

UDC 658.8

O. Chornovol

Cloud technologies as a basis for scaling intelligent systems in retail

У сучасному ритейлі автоматизація та впровадження інтелектуальних систем є ключовими факторами для підвищення ефективності бізнес-процесів і забезпечення конкурентних переваг. Особливу роль у цьому відіграють хмарні технології, які забезпечують необхідну інфраструктуру для обробки великих обсягів даних, масштабування систем та інтеграції нових рішень.

Інтелектуальні системи, такі як системи комп'ютерного зору, штучного інтелекту (ШІ) та машинного навчання, активно використовуються у ритейлі для таких задач, як автоматизація обліку товарів, моніторинг заповненості полиць, аналіз поведінки покупців та прогнозування попиту. Проте реалізація цих систем вимагає значних обчислювальних ресурсів, що часто перевищує можливості локальної інфраструктури. Хмарні платформи, такі як AWS, Google Cloud та Microsoft Azure, пропонують оптимальне рішення для цих викликів [1].

Хмарні технології забезпечують масштабованість інтелектуальних систем, дозволяючи легко збільшувати обчислювальну потужність у пікові періоди або при обробці великих обсягів даних. Наприклад, використання AWS Lambda дає можливість запускати функції на вимогу без необхідності постійного утримання серверів. Це значно знижує витрати на інфраструктуру та підвищує гнучкість системи [2].

Однією з важливих переваг хмарних технологій є можливість інтеграції з інструментами штучного інтелекту. Такі сервіси, як Google Cloud Vision або Amazon Rekognition, дозволяють швидко впроваджувати моделі комп'ютерного зору для аналізу зображень без необхідності самостійної розробки складних алгоритмів. Це особливо важливо для ритейлу, де точність і швидкість розпізнавання об'єктів, таких як товари на полицях, відіграють ключову роль.

Таким чином, хмарні технології є невід'ємною складовою сучасних інтелектуальних систем у ритейлі. Вони забезпечують високу продуктивність, гнучкість і масштабованість, необхідні для обробки великих обсягів даних та швидкої адаптації до змін ринку. Впровадження цих технологій відкриває нові можливості для автоматизації бізнес-процесів, підвищення ефективності та забезпечення кращого досвіду для клієнтів.

Література.

1. Марін Г. "Архітектура хмарних обчислень: основи та практика" 2020 - 256с
2. Szeliski, R. "Computer Vision: Algorithms and Applications" 2019 - 101с.

Лістинг коду серверної частини системи

```

from flask import Flask, request, jsonify
from ultralytics import YOLO
import cv2
import numpy as np
from collections import Counter

app = Flask(__name__)

model = YOLO('yolov5nu.pt') # Load YOLOv5 pre-trained model

def detect_and_count(image):
    """
    Detect all objects in an image using YOLOv5 and return the
    object with the largest count.
    :param image: Input image (numpy array).
    :return: Count of objects detected.
    """
    # Run the model on the image
    results = model(image)

    # Get the class names and predictions
    detected_objects = results[0].boxes.data.numpy()
    class_names = model.names

    if len(detected_objects) == 0:
        return 0 # No objects detected

    # Extract class IDs and corresponding class names from the
    detections
    class_ids = [int(obj[5]) for obj in detected_objects]

    # Count the occurrences of each object class
    class_count = Counter(class_ids)

    # Identify the class ID with the highest count
    most_common_class_id, count = class_count.most_common(1)[0]
    return count

@app.route('/detect', methods=['POST'])
def detect():
    """
    Handle image upload and object detection, then return the
    count in JSON format.
    :return: JSON response with the object count.
    """
    if 'file' not in request.files:
        return jsonify({'error': 'No file part'}), 400

    file = request.files['file']

    if file.filename == '':

```

```
        return jsonify({'error': 'No selected file'}), 400

    # Read the uploaded image as a numpy array
    img_array = np.frombuffer(file.read(), np.uint8)
    image = cv2.imdecode(img_array, cv2.IMREAD_COLOR)

    if image is None:
        return jsonify({'error': 'Invalid image file'}), 400

    # Detect and count objects
    object_count = detect_and_count(image)

    # Return the result as JSON
    return jsonify({'object_count': object_count})

if __name__ == '__main__':
    app.run(host="0.0.0.0", debug=True)
```

Лістинг коду мобільного застосунку

```
//
// ContentView.swift
// ObjectCounter
//

import SwiftUI

struct ContentView: View {
    @State private var showingImagePicker = false
    @State private var inputImage: UIImage?
    @State private var objectCount: Int? = nil
    @State private var showingAlert = false
    @State private var alertMessage = ""
    @State private var isLoading = false

    var body: some View {
        VStack {
            if let image = inputImage {
                Image(uiImage: image)
                    .resizable()
                    .scaledToFit()
            }

            if isLoading {
                ProgressView()
            } else if let objectCount = objectCount {
                Text("Object count: \(objectCount)")
            }

            Button("Take a photo") {
                showingImagePicker = true
            }
        }
        .sheet(isPresented: $showingImagePicker) {
            ImagePicker(image: $inputImage, sourceType: .camera)
                .onDisappear {
                    if let image = inputImage {
                        uploadImage(image)
                    }
                }
        }
        .alert(isPresented: $showingAlert) {
            Alert(title: Text("Error"), message:
Text(alertMessage), dismissButton: .default(Text("OK")))
        }
    }

    func uploadImage(_ image: UIImage) {
```

```

        isLoading = true
        objectCount = nil

        guard image.jpegData(compressionQuality: 0.1) != nil else
        { // Image quality, experiment with this!
            alertMessage = "Could not compress image."
            showingAlert = true
            return
        }

        let resizedImage = resizeImage(image: image, targetSize:
CGSize(width: 800, height: 600)) // Adjust target size as needed

        guard let resizedImageData =
resizedImage?.jpegData(compressionQuality: 0.5) else {
            alertMessage = "Could not compress image."
            showingAlert = true
            return
        }

        guard let url = URL(string: "http://host/detect") else {
            alertMessage = "Invalid URL"
            showingAlert = true
            return
        }

        var request = URLRequest(url: url)
        request.httpMethod = "POST"

        let boundary = "Boundary-\(UUID().uuidString)"
        request.setValue("multipart/form-data;
boundary=\(boundary)", forHTTPHeaderField: "Content-Type")

        var body = Data()
        body.append("--\((boundary)\r\n".data(using: .utf8)!)
        body.append("Content-Disposition: form-data;
name=\"file\"; filename=\"image.jpg\""\r\n".data(using: .utf8)!)
        body.append("Content-Type: image/jpeg\r\n\r\n".data(using:
.utf8)!)
        body.append(resizedImageData) // Use resized image data
        body.append("\r\n".data(using: .utf8)!)
        body.append("--\((boundary)--\r\n".data(using: .utf8)!)

        request.httpBody = body

        let task = URLSession.shared.dataTask(with: request) {
data, response, error in
            if let error = error {
                DispatchQueue.main.async {
                    alertMessage = "Upload failed:
\((error.localizedDescription)"
                    showingAlert = true
                }
            }
            return
        }
    }

```

```

        if let data = data {
            do {
                let json = try
JSONSerialization.jsonObject(with: data, options: []) as? [String:
Any]
                if let count = json?["object_count"] as? Int {
                    DispatchQueue.main.async {
                        self.objectCount = count
                        self.isLoading = false
                    }
                }
            } catch {
                DispatchQueue.main.async {
                    alertMessage = "Error decoding JSON:
\\(error.localizedDescription)"
                    showingAlert = true
                    self.objectCount = nil
                    self.isLoading = false
                }
            }
        }

        task.resume()
    }

    func resizeImage(image: UIImage, targetSize: CGSize) ->
UIImage? {
        let size = image.size

        let widthRatio = targetSize.width / size.width
        let heightRatio = targetSize.height / size.height
        var newSize: CGSize
        if(widthRatio > heightRatio) {
            newSize = CGSize(width: size.width * heightRatio,
height: size.height * heightRatio)
        } else {
            newSize = CGSize(width: size.width * widthRatio,
height: size.height * widthRatio)
        }

        let rect = CGRect(x: 0, y: 0, width: newSize.width,
height: newSize.height)

        // Actually do the resizing to the calculated size while
respecting the context's scale
        UIGraphicsBeginImageContextWithOptions(newSize, false,
1.0)
        image.draw(in: rect)
        let newImage = UIGraphicsGetImageFromCurrentImageContext()
        UIGraphicsEndImageContext()

        return newImage
    }
}

```

```

// Image Picker (Helper struct)
struct ImagePicker: UIViewControllerRepresentable {
    @Binding var image: UIImage?
    var sourceType: UIImagePickerController.SourceType

    func makeUIViewController(context: Context) ->
    UIImagePickerController {
        let picker = UIImagePickerController()
        picker.delegate = context.coordinator
        picker.sourceType = sourceType

        if sourceType == .camera { // Only apply scaling for
camera
            let screenSize = UIScreen.main.bounds.size
            let cameraAspectRatio = CGFloat(4.0 / 3.0) // Standard
camera aspect ratio
            let screenAspectRatio = screenSize.width /
screenSize.height

            let scale = screenAspectRatio > cameraAspectRatio
? screenSize.height / (screenSize.width *
cameraAspectRatio)
: screenSize.width / (screenSize.height /
cameraAspectRatio)

            picker.modalPresentationStyle = .fullScreen
        }

        return picker
    }

    func updateUIViewController(_ viewController:
UIImagePickerController, context: Context) {
        // No updates needed
    }

    func makeCoordinator() -> Coordinator {
        Coordinator(self)
    }

    class Coordinator: NSObject, UINavigationControllerDelegate,
UIImagePickerControllerDelegate {
        let parent: ImagePicker

        init(_ parent: ImagePicker) {
            self.parent = parent
        }

        func imagePickerController(_ picker:
UIImagePickerController, didFinishPickingMediaWithInfo info:
[UIImagePickerController.InfoKey: Any]) {
            if let uiImage = info[.originalImage] as? UIImage {

```

```
        parent.image = uiImage
    }
    picker.dismiss(animated: true)
}
}
```