

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА  
АКАДЕМІЯ»  
Факультет інформатики  
Кафедра інформатики

## **Магістерська робота**

Освітній ступінь: магістр

**На тему: «РОЗРОБКА СИСТЕМИ АНАЛІЗУ ТА ПОШУКУ  
НЕСТРУКТУРОВАНОЇ МУЛЬТИМЕДІЙНОЇ ІНФОРМАЦІЇ»**

Виконав: студент 2 року навчання

Спеціальності

121 Інженерія програмного забезпечення

Пухальський Богдан Мирославович

Керівник Глибовець А.М.

доктор технічних наук

Рецензент

Магістерська робота захищена

з оцінкою \_\_\_\_\_

Секретар ЕК С.А. Мелешенко

« \_\_\_ » \_\_\_\_\_ 2023

Київ 2023

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА  
АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

Зав.кафедри інформатики,  
к.ф.-м.н.

\_\_\_\_\_ С. С. Гороховський  
(підпис)

„\_\_\_\_\_” \_\_\_\_\_ 2023 р.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**

на магістерську роботу

студента 2 р.н. магістерської програми Комп'ютерні науки  
Пухальського Богдана Мирославовича

Розробка системи аналізу та пошуку неструктурованої мультимедійної  
інформації

Зміст текстової частини до магістерської роботи:

Зміст

Анотація

Вступ

1 Аналіз предметної області

2 Аналіз існуючих систем пошуку та обробки зображень та  
знаходження найбільш критичних проблем.

3 Розробка проекту системи

4 Порівняльна характеристика продуктивності розглянутих методів  
комп'ютерного зору

Висновки

Список літератури

Додатки

Дата видачі „\_\_\_\_\_” \_\_\_\_\_ 2023 р.

Керівник

(підпис)

Завдання отримав

(підпис)

**Тема:** “Розробка системи аналізу та пошуку неструктурованої мультимедійної інформації”

**Календарний план виконання роботи:**

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на дипломну роботу	01.11.2022	
2.	Огляд технічної літератури за темою роботи	15.11.2022	
3.	Провести аналіз предметної області	29.11.2022	
4.	Виконати аналіз існуючих систем пошуку та обробки мета інформації та знайти найбільш критичні проблеми.	27.12.2022	
5.	Розробити архітектуру системи аналізу та пошуку неструктурованої мультимедійної інформації	07.02.2023	
6.	Зібрати та проаналізувати методи розв'язання задач комп'ютерного зору, зробивши висновки, щодо продуктивності обраних підходів.	10.02.2023	
7.	Написання пояснювальної записки	12.05.2023	
8.	Створення слайдів для доповіді та написання доповіді	14.05.2023	
9.	Аналіз отриманих результатів з керівником, написання доповіді та попередній захист магістерської роботи	16.05.2023	
10.	Корегування роботи за результатами попереднього захисту	17.05.2023	
11.	Остаточне оформлення пояснювальної записки та слайдів	02.06.2023	
12.	Захист магістерської роботи (проекту)	13.06.2023	

**Студент Пухальський Богдан Мирославович**

**Керівник Глибовець Андрій Миколайович**

“ \_\_\_\_\_ ”

## ЗМІСТ

<i>ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ</i>	5
<i>АНОТАЦІЯ</i>	6
<i>ВСТУП</i>	7
<i>РОЗДІЛ 1. СИСТЕМИ ПОШУКУ ЗОБРАЖЕНЬ</i>	9
1.1. Огляд мультимедійних інформаційно-пошукових систем	9
1.2. Потреби користувача та його вимоги до систем пошуку зображень	12
1.3. Огляд існуючих систем отримання інформації із зображення	13
1.4. Переваги та недоліки	13
1.5. Запропоновані рішення для згаданих проблем	15
Висновки до розділу 1	15
<i>РОЗДІЛ 2. МЕТОДИ РОЗВ'ЯЗАННЯ ЗАДАЧ КОМП'ЮТЕРНОГО ЗОРУ</i>	17
2.1. Виявлення об'єктів	17
2.2. Розпізнавання облич	23
2.3. Текстовий опис зображень	25
Висновки до розділу 2	26
<i>РОЗДІЛ 3. ОГЛЯД РОЗРОБЛЕНОЇ СИСТЕМИ ПОШУКУ ТА АНАЛІЗУ ЗОБРАЖЕНЬ МЕТОДАМИ КОМП'ЮТЕРНОГО ЗОРУ</i>	27
3.1. Вимоги до системи пошуку та аналізу зображень	27
3.2. Опис програмних модулів	29
3.2.1. Контекст системи	31
3.2.2. Діаграма контейнерів	32
3.2.3. Діаграма компонентів	34
3.3. Принцип роботи розробленої системи	35
3.4. Порівняльна характеристика продуктивності розглянутих методів комп'ютерного зору	36
Висновки до розділу 3	38
<i>ВИСНОВКИ</i>	39
<i>СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ</i>	41

## **ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ**

QA - quality attributes (англ. атрибути якості).

REST API - representative state transfer application programming interface (англ. програмний інтерфейс застотунку репрезентативної передачі стану)

SVM - support vector machine (англ. векторна машина підтримки)

RPN - regional proposal network (англ. мережа пропозицій областей)

HTTP - hypertext transferring protocol (англ. протокол передачі гіпертексту)

SQL - structured query language (англ. структурована мова запитів)

mAP - mean average precision (англ. помірне середнє значення)

## АНОТАЦІЯ

Дана робота має на меті надати огляд систем пошуку зображень і розглянути методи, які використовуються для вирішення проблем комп'ютерного зору. Перша частина зосереджена на потребах користувачів і вимогах до систем пошуку зображень, після чого розглядаються існуючі системи, такі як Google Photos, Apple Photos, Amazon Photos, Microsoft Photos і Flickr, оцінюються їхні переваги та недоліки та обговорюються запропоновані шляхи вирішення виявлених обмежень. У другій частині розглядаються запропоновані методи комп'ютерного зору, включаючи виявлення об'єктів (облич) та текстовий опис зображень. У третьому розділі, представлено огляд розробленої системи пошуку та індексування зображень, що використовує методи комп'ютерного зору, висвітлюючи її принцип роботи, програмні модулі та порівняльну оцінку продуктивності серед розглянутих методів комп'ютерного зору.

**Ключові слова:** Системи пошуку зображень, Комп'ютерний зір, Виявлення об'єктів, Текстовий опис зображень

## ВСТУП

Швидке зростання цифрових зображень призвело до зростання потреби в ефективних системах пошуку зображень, які могли б ефективно задовольнити потреби користувачів.

В результаті, дослідження, представлене в цій дипломній роботі, має на меті надати повний огляд систем пошуку зображень і вивчити методи, які використовуються для вирішення проблем комп'ютерного зору для покращених можливостей пошуку та індексування.

Основною метою цього дослідження є аналіз потреб користувачів і вимог до систем пошуку зображень, визначення ключових аспектів, які сприяють ефективній та зручній роботі. Розуміючи ці вимоги, ми можемо оцінити існуючі системи, доступні на ринку і вивчити їхні сильні та слабкі сторони. Зокрема, ми зосередимося на добре відомих системах пошуку зображень, включаючи Google Photos, Apple Photos, Amazon Photos, Microsoft Photos і Flickr. Ці платформи представляють різноманітні підходи до пошуку зображень і пропонують цінну інформацію про поточний стан галузі.

Завдання дослідження передбачає проведення поглибленого вивчення згаданих систем, у тому числі їхніх характеристик, функціональних можливостей та взаємодії з користувачем. Крім того, ми запропонуємо рішення для вирішення виявлених обмежень або проблем, з якими стикаються ці системи, щоби покращити їх функціонал і задовольнити користувачів.

Для підтримки нашого дослідження буде використано різноманітні джерела, зокрема наукові статті та технічну документацію. Цей мульти-джерельний підхід надасть всебічне розуміння предмета та забезпечить надійність і обґрунтованість наших висновків.

Наукова новизна результатів полягає в комплексному аналізі потреб і вимог користувачів до систем пошуку зображень, оцінці існуючих систем і пропозиції рішень для підвищення їх функціональності. Крім того, дане дослідження сприятиме знаходженню нових способів застосування комп'ютерного зору шляхом вивчення різних методів вирішення проблем комп'ютерного зору, таких як виявлення об'єктів.

Результати цієї роботи можуть слугувати цінним ресурсом для дослідників, розробників і професіоналів галузі, які займаються розробкою та впровадженням систем пошуку зображень. Запропоновані рішення та висновки, отримані в результаті цього дослідження, можуть скеровувати майбутні досягнення в галузі, що зрештою призведе до підвищення ефективності та зручності для користувачів.



## **РОЗДІЛ 1. СИСТЕМИ ПОШУКУ ЗОБРАЖЕНЬ**

У першому розділі описується теоретична частина щодо пошуку зображень. Для того, щоб зрозуміти доцільність вдосконалення існуючих мультимедійних інформаційно-пошукових систем, у другому розділі описуються потреби кінцевих користувачів розглянутого програмного забезпечення. У третьому та четвертому розділах здійснено огляд таких систем як: Google photos, Apple photos, Amazon Photos, Microsoft Photos та Flickr, а у четвертому висвітлені їх переваги та недоліки. Для виправлення недоліків, у п'ятому розділі, поверхнево описуються запропоновані методи розв'язання задач комп'ютерного зору.

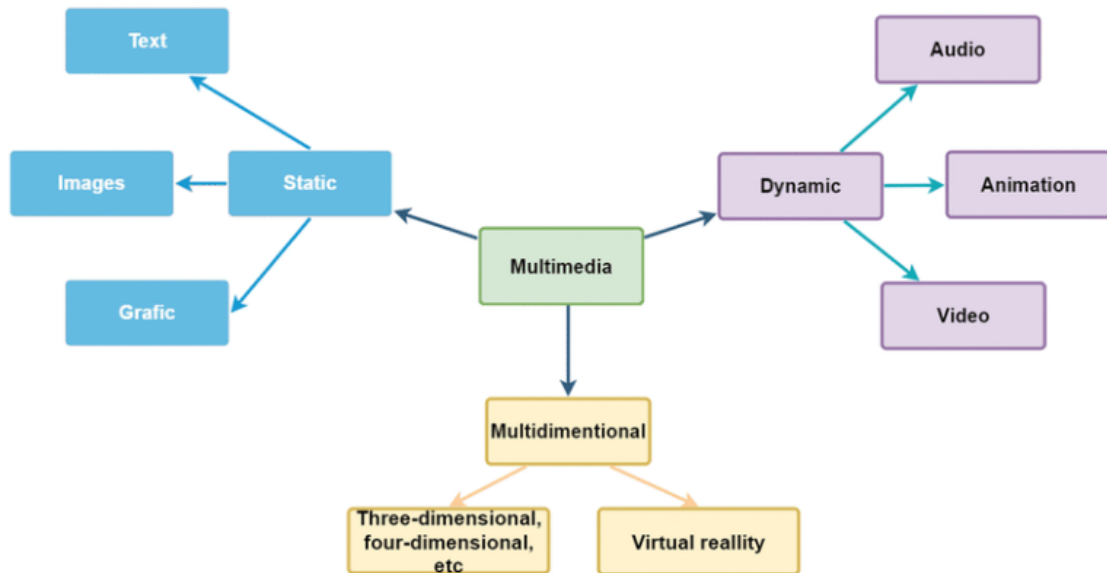
### **1.1.Огляд мультимедійних інформаційно-пошукових систем**

Мультимедія - це форма комунікації, що поєднує в собі такі види інформації як текст, аудіо, зображення, анімацію та відео[2]. Навідміну від "традиційних" засобів надання інформації, таких як газети, мультимедія надає засоби взаємодії користувача. Мультимедійна інформація репрезентується, обробляється та передається через численні сенсорні канали, такі як камери спостереження, датчики звуку, тощо. Поєднання різних типів медіа підвищує насиченість і ефективність спілкування, створюючи більш захоплюючий та інтерактивний досвід.

У контексті систем пошуку зображень мультимедія відіграє вирішальну роль у захопленні та представленні візуальної інформації. Розуміння основ мультимедія має важливе значення для розуміння складності та потенціалу систем пошуку зображень.

Мультимедія охоплює широкий діапазон типів медіа, кожен сприяє загальному досвіду користувача. Загалом вони поділяються на статичні

(текст, зображення), динамічні (аудіо, відео) та багатовимірні (віртуальна реальність)[3]. Схему класифікації надано нижче.



*Рисунок 1* Схема класифікації мультимедійної інформації [3]

Серед поширених мультимедійних типів:

1. Текст: текстова інформація, така як підписи, описи та метадані, надає контекстну інформацію та допомагає зрозуміти зміст мультимедійних об'єктів [7].
2. Зображення: зображення — це візуальні уявлення, які можуть передавати інформацію, викликати емоції та служити засобом спілкування [4].
3. Аудіо: аудіокомпоненти включають звукові доріжки, голос за кадром, фонову музику або будь-яку форму аудіозаписів, які супроводжують мультимедійні презентації [5].
4. Відео: відео поєднує в собі рухомі зображення та аудіо, що дозволяє знімати та відтворювати динамічні сцени, події чи оповідання [4].

5. Анімація: анімація передбачає послідовне відображення зображень або візуальних елементів для створення ілюзії руху або зміни [7].

Кожен із цих типів потребує спеціальних методів і алгоритмів для ефективного зберігання, індексування та пошуку.

Фотографії займають важливе місце в мультимедіа і їх ефективний пошук є ключовою метою систем пошуку зображень. Отримання відповідних фотографій із великих колекцій вимагає надійних механізмів індексування [6].

Індексація фотографій включає вилучення значущих характеристик із зображень, таких як колір, текстура, форма та просторова інформація [4]. Потім ці функції використовуються для створення компактних представлень або дескрипторів зображень, які можна ефективно зберігати та здійснювати пошук [7].

Щоб увімкнути подальший пошук та індексацію, ці дескриптори зображень організовуються та індексуються за допомогою відповідних структур даних, таких як інвертовані файли, хеш-таблиці або деревоподібні структури [6]. Цей процес індексування дозволяє здійснювати ефективний пошук на основі різних критеріїв, таких як схожість, релевантність або специфічні візуальні атрибути [4]. Пошук за дескрипторами, згаданими вище, широко використовується у різноманітних пошукових системах та не несе наукової новизни. У даній роботі будуть розглянуті такі дескриптори зображень як виявлення об'єктів та текстовий опис зображень, які будуть отримані засобами комп'ютерного зору та застосовані для подальшої індексації та пошуку. Деякі із згаданих вище засобів комп'ютерного зору використовуються у системах вилучення інформації із мультимедіа.

Система отримання інформації із мультимедіа відноситься до спеціалізованої системи, призначеної для пошуку, отримання та керування

мультимедійним вмістом, включаючи зображення, аудіо та відео. Вона використовує передові методи та алгоритми для аналізу та індексування мультимедійних даних, полегшення ефективного пошуку на основі запитів користувачів [7].

Система пошуку мультимедійних зображень поєднує в собі області пошуку інформації, комп'ютерного зору та обробки мультимедіа, щоб подолати проблеми пошуку та керування мультимедійним вмістом [5]. Вона охоплює різні компоненти, такі як вилучення функцій, індексування, рейтинг релевантності та користувацькі інтерфейси, щоб забезпечити інтегрований і зручний пошук[7].

Ці системи спрямовані на подолання семантичного розриву між низькорівневими візуальними функціями та високорівневими запитам користувача, забезпечуючи ефективний пошук і перегляд мультимедійного вмісту[5]. Вони відіграють життєво важливу роль у різних областях, включаючи цифрові бібліотеки, бази даних зображень, платформи соціальних медіа та особисті колекції фотографій.

## **1.2. Потреби користувача та його вимоги до систем пошуку зображень**

Зазвичай, користувачами подібних систем пошуку мультимедійної інформації є звичайні користувачі, яким потрібен зручний пошук у особистій галереї, журналісти, що публікують фотозвіти чи проводять розслідування, та інші. Серед потреб користувачів можна виділити наступні:

1. Знайти зображення за датою.
2. Знайти за геолокацією.
3. Знайти за об'єктами зображеними на фото.
4. Знайти за особами зображеними на фото.
5. Знайти за описом дій, що виконуються на фото.

Перші дві функції згадані вище реалізовані у більшість систем пошуку. Подібна інформація знаходиться у метаданих зображення, а перед тим наповнюється програмним забезпеченням встановленим на пристрої із якого робиться фотографія. Третя, четверта та п'ята функції здебільшого реалізовані у інформаційних системах пошуку зображень. Для вилучення необхідної інформації використовуються нейромережі, які викликаються при завантаженні фото у систему. Згодом, отримані результати записуються у певне сховище даних та індексуються для подальшого пошуку.

### 1.3.Огляд існуючих систем отримання інформації із зображення

Нижче наведено таблицю із розглянутими програмними продуктами та їх функціоналом, що стосуються предмета нашого дослідження.

Таблиця 1. Порівняльна характеристика функціоналу розглянутих систем

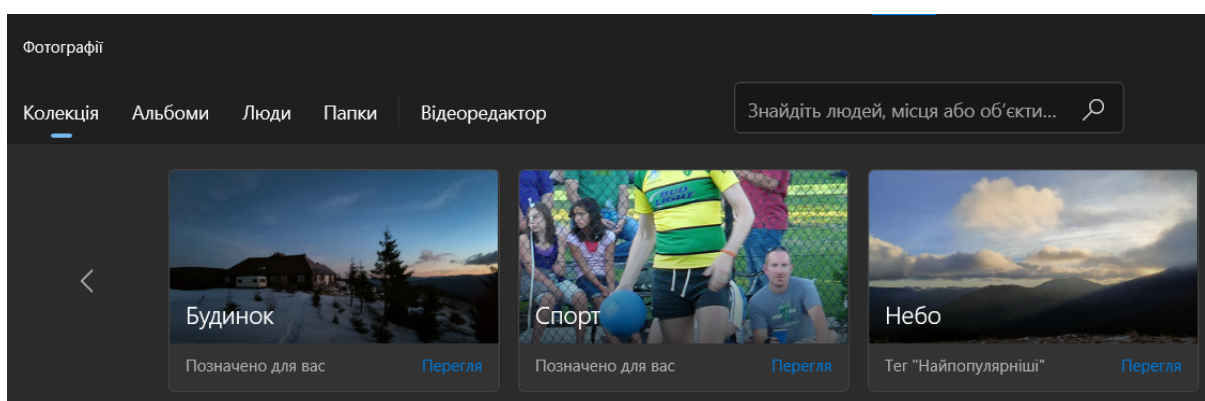
	Пошук за об'єктами	Пошук за обличчями	Пошук за діями
Google Photos	+	+	-
Apple Photos	+	+	-
Amazon Photos	+	+	-
Microsoft Photos	+	+	-
Flickr	+	+	-

### 1.4. Переваги та недоліки

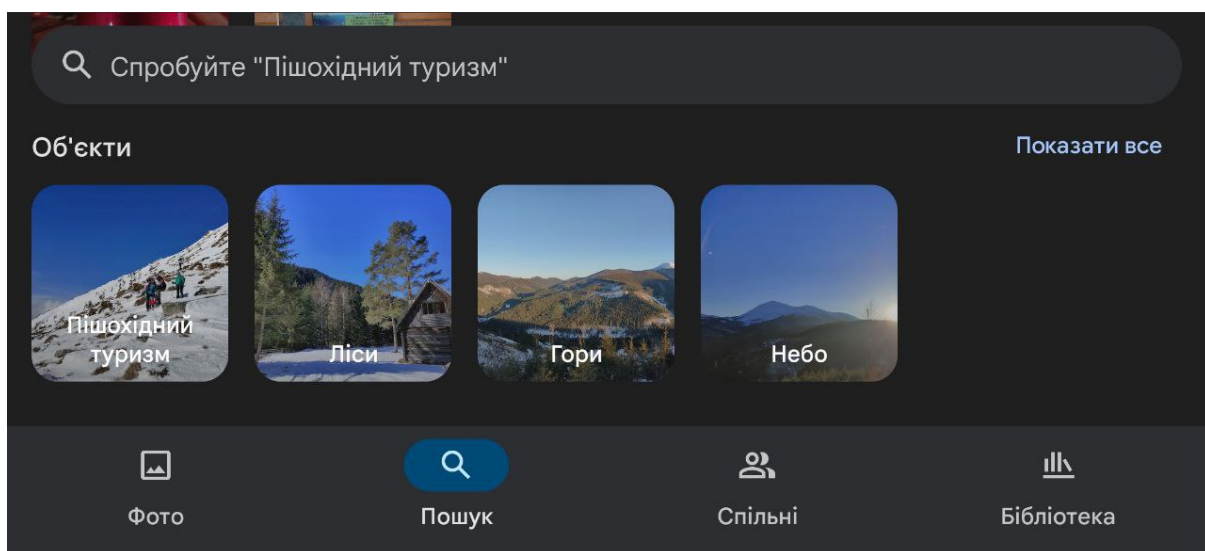
У наданій вище таблиці зазначено, що у кожному продукті реалізований функціонал розпізнавання об'єктів, в тому числі й облич. За

допомогою жодного розглянутого застосунку не можна знайти фото за описом.

Один і той самий функціонал, в нашому випадку - це визначення об'єктів, може виконуватись за допомогою різних нейромереж. Після тестування вищезгаданих застосунків було виявлено різні набори об'єктів на одному і тому ж наборі даних. Результати перевірки зображено на рисунках нижче.



1



2

*Рисунок 2 Результати обробки зображень на наявність об'єктів: 1 - у Microsoft Photos; 2 - у Google Photos*

У відкритих джерелах можна знайти їх порівняльну характеристику точності, але дані щодо продуктивності, на жаль, відсутні.

Продуктивність є важливим атрибутом якості інформаційних систем. Надмірне очікування є неприпустимим для користувача системи [27]. Крім того, продуктивність тісно пов'язана із масштабованістю, оскільки такі показники як пропускна здатність та час очікування відповіді визначають момент для розширення обчислювальних потужностей системи.

### **1.5. Запропоновані рішення для згаданих проблем**

Реалізацію розглянутого функціоналу можна розглянути як розв'язання типових задач комп'ютерного зору. Існує 4 основних класи задач комп'ютерного зору: класифікація зображень, виявлення об'єктів, сегментація зображень та текстовий опис зображень[13]. Для пошуку за об'єктами, в тому числі і обличчями, буде використовуватись задача виявлення об'єктів, а для пошуку за діями - задача текстового опису зображень.

Виявлення об'єктів - це поєднання задач локалізації об'єктів та класифікації зображень, де класифікація виконується окремо для кожного об'єкта. Важливою підзадачею є виявлення людей та їх облич[14].

Текстовий опис зображень – це в основному виявлення об'єктів + підписи[13]. Дану задачу можна розглядати як розширення задачі виявлення об'єктів згаданої раніше.

Дані задачі детальніше будуть розглядатися у наступному розділі.

### **Висновки до розділу 1**

У розділі 1 наукової праці подано огляд систем пошуку зображень. Він розпочався з обговорення важливості мультимедійної інформації та пошукових систем та їхнього значення для пошуку зображень. Потім у розділі було розглянуто потреби користувачів і вимоги до ефективного пошуку зображень, враховуючи фактори, які сприяють задоволенню та досвіду користувачів. Крім того, було проведено комплексне дослідження

існуючих систем пошуку зображень, включаючи Google Photos, Apple Photos, Amazon Photos, Microsoft Photos і Flickr, з оцінкою їхніх особливостей, функцій, переваг і недоліків. У п'ятому підрозділі були перераховані задачі комп'ютерного зору, які потрібно вирішити для додавання потрібного функціоналу системи.

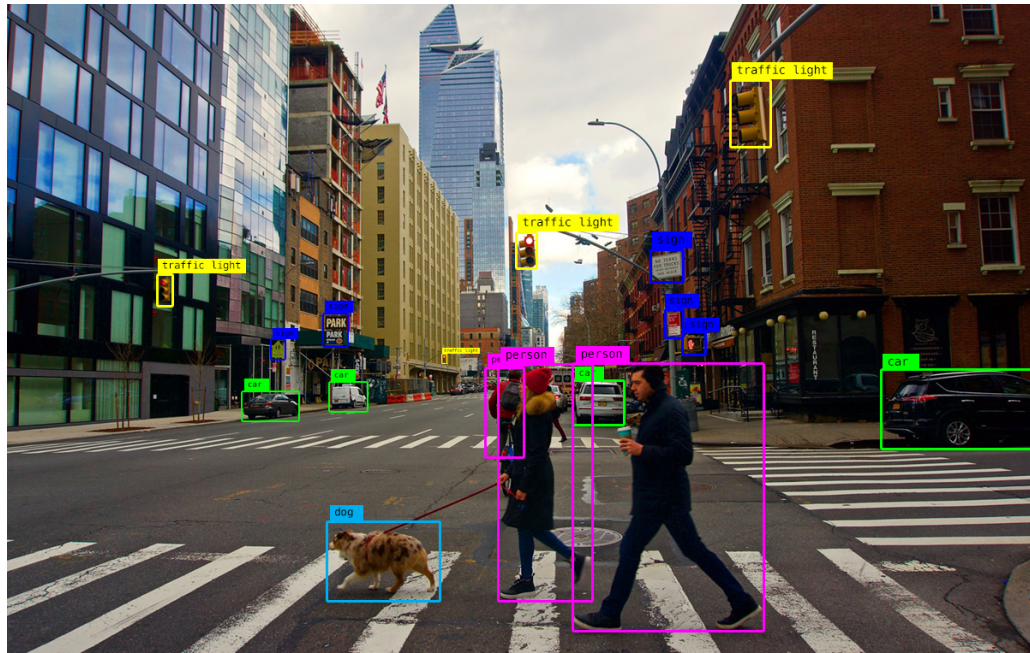


## **РОЗДІЛ 2. МЕТОДИ РОЗВ'ЯЗАННЯ ЗАДАЧ КОМП'ЮТЕРНОГО ЗОРУ**

На даний момент існує п'ять найпопулярніших задач комп'ютерного зору, а саме: класифікація зображень, визначення об'єктів (в тому числі і облич), сегментація зображень, текстовий опис зображень. У даному розділі такі задачі як класифікація зображень та сегментація зображень розглядатися не будуть. Перша задача вже розв'язується у більшості пошукових систем, а друга не є доцільною для функціоналу подібних систем. Тому дана частина присвячена лише задачам виявленням об'єктів (облич) та текстовому опису зображень.

### **2.1 Виявлення об'єктів**

Виявлення об'єктів — це важливе завдання комп'ютерного зору, яке має справу з виявленням екземплярів візуальних об'єктів певного класу (таких як люди, тварини чи автомобілі) у цифрових зображеннях. Метою виявлення об'єктів є розробка обчислювальних моделей і методів, які надають одну з основних частин інформації, необхідної для програм комп'ютерного зору, тобто, які об'єкти де знаходяться [15]. Демонстрацію виконання цієї задачі зображено нижче.



*Рисунок 3 Демонстрація задачі виявлення об'єктів [3]*

Розв'язок задачі виявлення об'єктів має застосування у різноманітних сферах. У наш час люди використовують системи вилучення інформації із мультимедіа, такі як: Google Photos, Microsoft Photos та інші для зберігання особистих великих колекцій фотографій та швидкого пошуку по об'єктах, які зображені на цих фото.

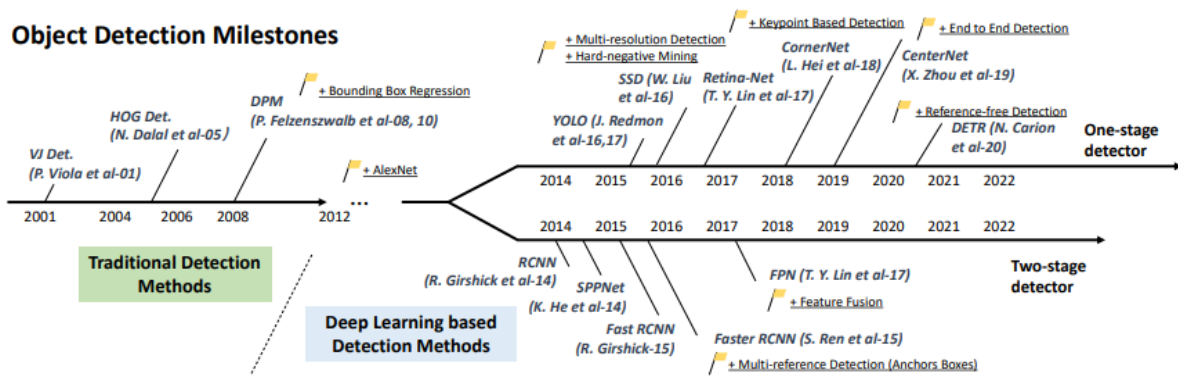
Часто, на місце подій прибувають репортери для того щоб відзняти фото чи відео матеріал. Перед публікацією їм потрібно відібрати відповідні фотографії. Система, за допомогою якої можна знайти зображення за об'єктами, суттєво спрощує фільтрування фотографій перед самою публікацією. Досить часто журналісти проводять OSINT розслідування на основі відкритих джерел[16]. Серед таких джерел може бути і мультимедіа, в тому числі і зображення. Як і у випадку із фоторепортажами, пошук за об'єктами суттєво спрощує підготовку результатів розслідування.

Застосування виявлення об'єктів у сфері охорони здоров'я зробили революцію в медичній візуалізації та діагностиці, дозволивши

автоматизовано виявляти та локалізувати анатомічні структури, ураження та аномалії. Ці програми використовують методи комп'ютерного зору, алгоритми глибокого навчання, і передові технології візуалізації для підвищення точності та ефективності виявлення захворювань, діагностики та планування лікування. Завдяки автоматизації процесу виявлення об'єктів медичні працівники можуть заощадити час, покращити результати лікування пацієнтів і сприяти більш персоналізованому та цілеспрямованому втручанню[17].

Виявлення об'єктів у військовій сфері стало критично важливою технологією, яка революціонізувала операції оборони та безпеки. Використовуючи методи комп'ютерного зору та алгоритми глибокого навчання, системи виявлення об'єктів дозволяють автоматизовано ідентифікувати та відстежувати необхідні об'єкти, такі як транспортні засоби, зброя чи жива сила. Ці технології покращують ситуативну обізнаність, допомагають у розпізнаванні цілей, підтримують місії спостереження та допомагають у виявленні загроз і реагуванні на них[18].

Шлях розвитку рішень для задачі виявлення об'єктів можна умовно поділити на 2 періоди. Перший період - це традиційні методи рішення задачі, розвиток яких закінчився у 2014 році. Дані методи є алгоритмами, що базуються на ознаках об'єкта заданих вручну. Із розвитком нейронних мереж та глибокого навчання розпочався другий період, методи якого базуються на алгоритмах машинного навчання. Розвиток у цьому напрямку триває і досі[19]. Хронологія віх у розвитку рішень задачі виявлення об'єктів зображено нижче.



*Рисунок 4 Історія розвитку способів рішення задачі виявлення об'єктів[19]*

Традиційні методи розв'язання задачі виявлення об'єктів поділяються на ті, що засновані на особливостях, ті, що засновані на шаблонах та ті, що засновані на зовнішньому вигляді.

Відповідно до технік, які базуються на особливостях, знаходяться риси притаманні певному об'єкту. У випадку із обличчям це очі, ніс, рот і тому подібні. Наступним кроком є перевірка на правдоподібне географічне розташування виявлених рис.

Техніки, що базуються на шаблонах здатні виявляти об'єкт у різноманітних позиціях та вираженнях. Однак, вони вимагають належної ініціалізації та їх швидкодія не є високою.

Згідно із технікою, що базується на зовнішньому вигляді, зображення поділяється на прямокутні зони, які можуть перекривати одна одну. Згодом, у цих зонах, знаходяться кандидати на об'єкт, який потрібно виявити. Після цього, його можна уточнити використовуючи каскад більш складних, але селективних алгоритмів виявлення. Більшість підходів, заснованих на зовнішньому вигляді, сильно покладаються на навчання класифікаторів з використанням наборів мічених об'єктів і не об'єктних фрагментів.

Серед “традиційних” детекторів найвідомішими є Viola Jones Detector, HOG Detector, та Deformable Part-based Model (DPM). У розробленій системі пошуку та аналізу неструктурованої мультимедійної інформації будуть застосовуватись та порівнюватись актуальні методи глибинного навчання для виявлення об’єктів.

Способи рішення періоду глибинного навчання розділяються на 2 види, які розвивалися паралельно: виявлення у 2 етапи та виявлення у один етап. Перший спосіб визначає виявлення як процес “від грубого до тонкого”, тоді як другий є виявленням за один крок.

Найвідомішими двоетапними нейронними мережами є RCNN, SPPNet, FastRCNN, FasterRCNN. Короткий опис кожної із них надано нижче.

1. RCNN - Regions with Convolutional Neural Networks (англ. області та згорткові нейронні мережі). Першим етапом роботи цієї нейромережі є дістати пропозиції об’єктів. Потім кожна пропозиція масштабується до фіксованого розміру та передається до CNN моделі натренованої на ImageNet, щоб дістати особливості. І наостанок, використовуються SVN класифікатори, щоб визначити наявність об’єкта у кожній області зображення та розпізнати категорії об’єкта.
2. SPPNet - Spatial Pyramid Pooling Networks (англ. мережі просторової піраміди) Навідміну від попередніх нейромереж, дана мережа містить шар просторової піраміди, що дозволяє репрезентацію фіксованої довжини незалежно від розміру вхідного зображення чи області без масштабування.
3. FastRCNN - дозволяє одночасно тренувати детектор та регресор границь області при незмінній конфігурації мережі.
4. FasterRCNN - є першим наскрізним та близьким до реального часу детектором. Головною особливістю є запровадження шару RPN,

який забезпечує не ресурсозатратні припущення областей. Крім того, окремі блоки системи виявлення об'єктів, такі як: виявлення припущення, вилучення особливостей, регресія границь області, та інші, були поступово інтегровані в уніфікований наскрізний фреймворк навчання.

Серед одноетапних нейронних мереж для виявлення об'єктів варто згадати: YOLO, SSD, Retina Net. Нижче коротко описана кожна із них.

1. You Only Look Once (YOLO) - перший одноетапний детектор ери глибинного навчання. По своєму принципу роботи ділить зображення на області, визначає границі та ймовірності для них.
2. Single Shot Multibox Detector (SSD) - основною відмінністю є запровадження методів виявлення із декількома посиленнями та з різною роздільною здатністю, що значно підвищує точність виявлення одноетапного детектора, особливо для деяких малих об'єктів.
3. Retina Net - оскільки одноетапні детектори не вирізняються точністю роботи у Retina Net було введено нову функцію витрат під назвою «фокальна витрата», яка змінює стандартні витрати перехресної ентропії, щоб під час навчання детектор більше зосереджувався на складних, неправильно класифікованих прикладах.

Визначається, що по своєму принципу, двоетапні нейронні мережі дають точніший результат, тоді як одноетапні швидший. При виборі нейронної мережі, що буде застосовуватись у прикладному програмному забезпеченні, потрібно маневрувати між точністю та продуктивністю. Під час планування розробки нашою системи було прийнято інтегрувати одну двоетапну та одну одноетапну нейронну мережу та порівняти результати роботи. Серед одноетапних ми обрали нейронну мережу YOLOv4. Через відсутність додаткових функцій підвищення точності роботи,

припускається, що дана мережа є найшвидшою, а це саме та метрика, яка цікавить нас у даному дослідженні. Серед двоетапних ми обрали Faster-RCNN. Оскільки, завдяки своїм архітектурним особливостям, припускається, що вона є найшвидшою нейронною мережею серед одноетапних, що підвищує інтерес для порівняння із швидкими одноетапними мережами.

## 2.2 Розпізнавання облич

Окремо варто зазначити таку задачу комп'ютерного зору як розпізнавання облич, яка є окремим випадком задачі виявлення об'єктів. Протягом всього часу свого існування, технологія знайшла застосування у автоматизованому відслідковуванні дотримання правил дорожнього руху, у розслідуванні кримінальних справ, у виявленні емоцій і накладання масок у мобільних застосунках та у інших сферах. Приклад застосування у автофокусі фотокамери зображено нижче. Хоч і виявлення облич є підкласом більш загальної задачі, вона є дещо специфічною та має свої складнощі та виклики.

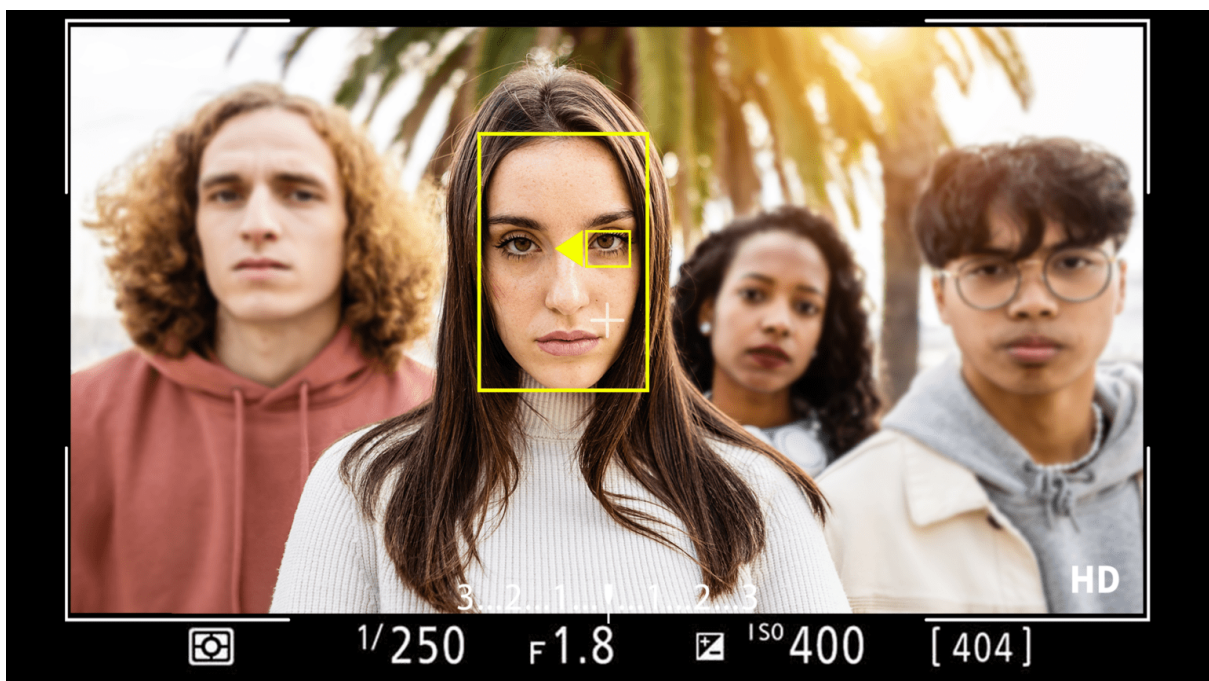


Рисунок 5 Виявлення облич у автофокусі фотокамери[20]

При виявленні облич часто можна зіткнутися із такими складнощами як: варіативність, перекриття, виявлення у різних масштабах та виявлення у реальному часі [20]. Людське обличчя здатне виражати багато емоцій, причому кожна людина виражає їх по своєму, того ж існує велика варіативність кольору та відтінку шкіри, що ускладнює процес виявлення. Обличчя можуть бути частково закритими різноманітними об'єктами та аксесуарами, такими як: волосся, головні убори, окуляри, макіяж, маска та інші. Масштаб фотографій облич також різниться. Лице може бути сфотографована, як зблизька, здалеку так і в натовпі. Виявлення у реальному часі обличчя, яке містить досить багато характерних рис вимагає залучення швидких обчислювальних потужностей, особливо коли мова йде про мобільні пристрої.

Згаданий раніше детектор Віола-Джонс був одним із перших спроб рішення цієї проблеми та став основою розвитку пізніших детекторів [21].

### **2.3 Текстовий опис зображень**

Текстовий опис зображень є відносно новою проблемою в області комп'ютерного зору. Визначення проблеми полягає у генеруванні опису, який є найбільш доречним до заданого зображення. Опис містить об'єкти, що зображені на заданому мультимедійному файлі та зв'язки між цими об'єктами. Серед зв'язків можуть бути дії, що виконують знайдені об'єкти, сполучники, прийменники та інші частини мови, що забезпечують цілісність опису. Не існує вимог щодо структури опису. Він може бути наданим у довільній формі[22].

Можна знайти багато застосувань вирішенням даної проблеми. Найбільш очевидне - це генерація опису всього, що відбувається на вихідному зображенні для підвищення ефективності пошуку самого зображення. Це дозволяє нам забезпечити посередництво між людиною та її високорівневими запитами та системою аналізу та пошуку зображень та



її низькорівневими функціями, що суттєво підвищує зручність використання подібних систем.

Наступною цікавою галуззю застосування даної функції можна зазначити інклюзивність. Люди із вадами зору, які не здатні чітко бачити мультимедійний вміст, отримують текстовий опис зображення зручним для них шрифтом.

По своїй суті текстовий опис зображень є комбінацією виявлення об'єктів згаданих раніше сам опис[13]. Протягом останніх майже десяти років, виявлення виконується методами глибокого навчання згаданими раніше. Опис здобувається такими нейронними мережами як: RNN (Recurrent Neural Net) та LSTM (Long Short Term Memory), яка є розширеною версією RNN. Існує також модель Show and Tell, яка поєднує у собі два вищезгаданих рішення із використанням LSTM [23]. На рисунку нижче можна ознайомитись зі структурою даної моделі.

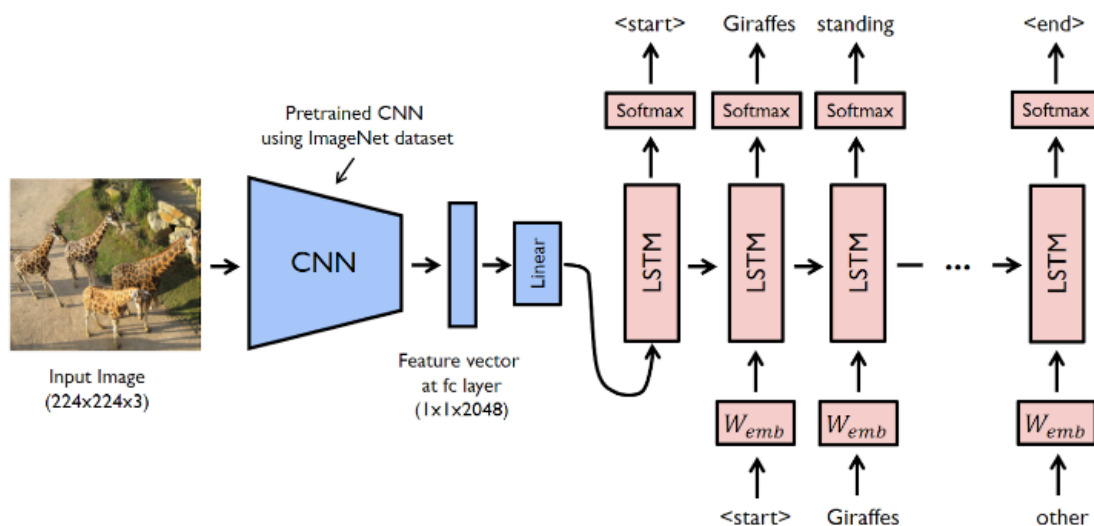


Рисунок 6 Схема моделі Show and Tell [24]

При розробці системи аналізу та пошуку зображень буде використовуватись даний підхід із конфігураціями шарів для виявлення об'єктів.

## **Висновки до розділу 2**

У розділі 2 було досліджено різні методи вирішення проблем комп'ютерного зору. Акцент був зосередженим на виявленні об'єктів, зокрема на розпізнаванні облич, і використанні текстового опису у пошуку зображень. У цьому розділі було надано уявлення про методи, які використовуються для розпізнавання обличчя та застосування текстового опису зображень, створення основи для подальшого розвитку системи пошуку та індексування зображень. У відкритих джерелах можна знайти порівняльну характеристику розглянутих способів. На жаль, у них представлена єдина метрика – точність. Для розробки прикладного програмного забезпечення нас цікавлять атрибути якості розробленого сервісу, особливо продуктивність. У наступному розділі на прикладі розробленої системи будуть здійснені виміри, а саме пропускну здатність та час отримання відповіді.

## **РОЗДІЛ 3. ОГЛЯД РОЗРОБЛЕНОЇ СИСТЕМИ ПОШУКУ ТА АНАЛІЗУ ЗОБРАЖЕНЬ МЕТОДАМИ КОМП'ЮТЕРНОГО ЗОРУ**

У цьому розділі ми проводимо комплексний огляд розробленої системи пошуку та індексування зображень, яка використовує методи комп'ютерного зору. Система була розроблена для підвищення ефективності та точності завдань пошуку зображень. У даній секції проведено заглиблення в принцип роботи та архітектуру розробленої системи, забезпечуючи розуміння її основних механізмів. Крім того, описано модулі програмного забезпечення, які складають систему, підкреслюючи їх функціональні можливості та внесок у загальний процес пошуку та індексування. В останньому розділі проведено порівняльну характеристику розглянутих методів рішення задач комп'ютерного зору у

вигляді таблиці. Включено такі властивості як точність моделі, а також, пропускну здатність та час отримання відповіді, що були виміряні на розробленій системі.

### **3.1 Вимоги до системи пошуку та аналізу зображень**

З точки зору архітектури програмних систем, вимога - це додатне для використання представлення потреби, яке допомагає визначити який вид цінності потрібно надати, щоб її задовольнити. Вимоги описують умови або здатність, яким має відповідати система. Вони впливають безпосередньо з потреб користувача або викладені в контракті, стандарті, специфікації чи іншому офіційно встановленому документі.

Архітектурна вимога, у свою чергу, — це будь-яка вимога, яка впливає на архітектуру, незалежно від того, чи є це значення явним чи неявним.[27]

Класифікація вимог визначена міжнародним інститутом бізнес аналізу у праці “Посібник із сукупності знань з бізнес-аналізу”[28]. Високорівнево, вимоги поділяються на бізнес вимоги, вимоги зацікавлених сторін, вимоги до розвитку та вимоги рішень. Перші три класи вимог формулюються та збираються на ранніх етапах розробки програмного продукту. У нашому випадку, при розробці практичної частини дослідження, доцільним є лише формулювання вимог рішення, які у свою чергу, поділяються на функціональні та нефункціональні.

Вимоги рішення описують можливості та якості рішення, яке відповідає вимогам зацікавлених сторін. Вони використовуються групою розробників для розробки системи. Наприклад, дозволити до 200 одночасних користувачів або вимагати надійні паролі.

Функціональні вимоги зображують поведінкові та інформаційні можливості, які будуть керуватися розробленим рішенням. Наприклад, відкрити можливість користувачу замовляти товари лише після реєстрації.

Нефункціональні вимоги це вимоги до системи, які можуть бути оцінені з точки зору кінцевого користувача та не описують специфічну поведінку та є джерелом атрибутів якості, наприклад, конфіденційні дані захищені від несанкціонованого доступу.

При розробці системи аналізу та пошуку неструктурованої інформації були зформульовані наступні вимоги:

Функціональні:

1. У користувача повинна бути можливість завантажити зображення через REST API.
2. Із зображення по чергово дістаються такі дані як об'єкти та текстовий опис зображень на зображенні
3. Отримані дані та вихідне зображення записуються у базу для подальшого пошуку
4. У користувача повинна бути можливість знаходити зображення за описом та об'єктами
5. Якщо особи, зображеної на фото, не існує у базі даних, то у користувача повинна бути можливість надати їй ім'я або залишити присвоєний ідентифікатор

Нефункціональні:

1. Кількість підтримуваних онлайн-користувачів становить 50 000

### **3.2 Опис програмних модулів**

Кожна зацікавлена сторона має погляд та точку зору. Погляд - це представлення одного або кількох аспектів архітектури, яке ілюструє, як архітектура вирішує проблеми, які мають одна або більше зацікавлених сторін. Точка зору - це колекція патернів, шаблонів і конвенцій побудови одного типу погляду. Вона визначає зацікавлених сторін, чий занепокоєння відображаються в точці зору [37]. Кожна зацікавлена сторона бачить та

розуміє архітектуру по своєму та є зацікавленою у вирішенні питань, які стосуються саме її. Опис архітектури повинен бути зрозумілим для всіх однаково. Саме тому були розроблені архітектурні нотації: порядок правил для опису архітектури програмних систем. Існують 4 найбільш вживані архітектурні нотації: 4+1, C4, SEI, Rozanski and Woods. Детальніше про них описано нижче.

1. 4+1: містить в собі логічний погляд, точку зору розробки, процесуальний погляд, фізичний погляд та сценарій

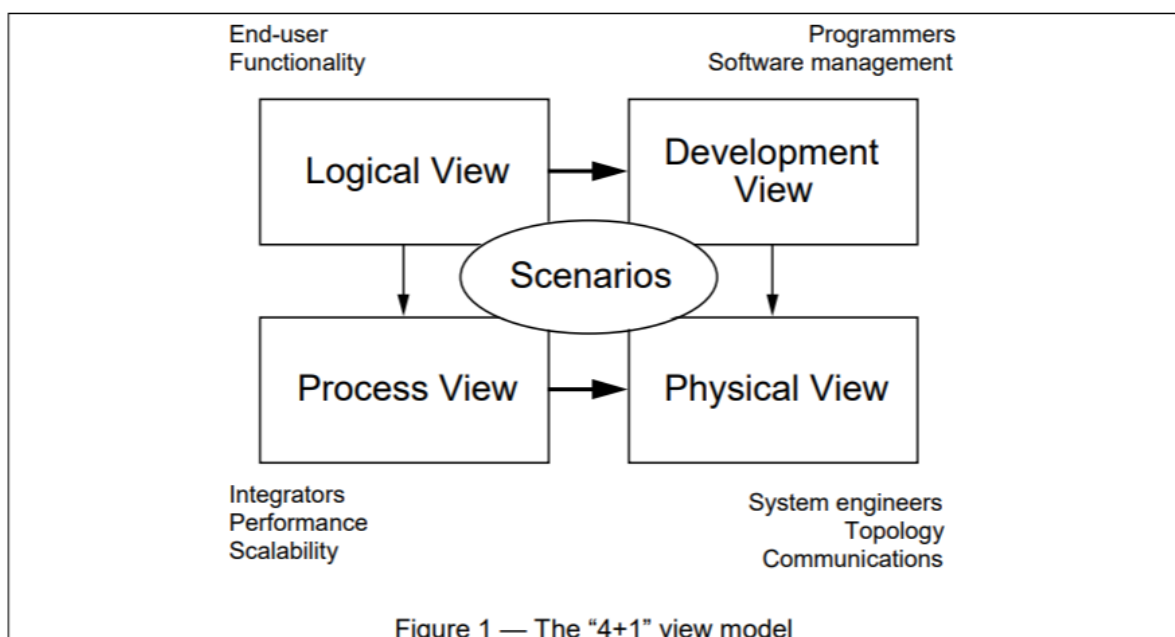


Рисунок 7 Архітектурна нотація 4+1. [33]

2. C4: Містить в собі діаграму контексту системи, діаграму контейнерів, діаграму компонентів та діаграму коду [34]
3. SEI: Архітектурна нотація розроблена Software Engineering Institute (англ. Інститут інженерії програмного забезпечення). Базується на загальноновизначених принципах декомпозиції та розподілу. Містить в собі: Позначення для демонстрації існування інтерфейсів, позначення для передачі синтаксичної інформації та позначення для передачі семантичної інформації. [36]

4. Rozanski and Woods - це архітектурний опис, набір артефактів, які документують архітектуру таким чином, щоб її зацікавлені сторони могли зрозуміти, і продемонструвати, що архітектура вирішила їхні проблеми. Складається з поглядів, точок зору та перспектив, що демонструють атрибути якості.[37]

Під час опису архітектури розробленої системи була обрана нотація C4. Вона описує точку зору для кожної зацікавленої сторони у зручній для неї формі та, на мою думку, не містить надлишкової інформації. Нижче описано 3 діаграми даної нотації: контекст, контейнери та компоненти.

Хоча діаграма коду також входить у четвірку діаграм C4, але вона не є обов'язковою. Діаграма коду, зазвичай, являє собою UML діаграму класів, діє в межах одного компоненту, згаданого на діаграмі на рівень вище та є доцільною лише для опису складної бізнес логіки, що буде реалізовуватись у розглянутому компоненті[35]. У нашому випадку випадку бізнес логіка та функціональні вимоги не є досить складними, тому додавати цей рівень потреби немає. Принцип роботи системи детальніше буде розглянуто у розділі 3.3.

### **3.2.1 Контекст системи**

На наведеній нижче схемі представлена високорівнева структура всієї системи. Вона описує основні внутрішні компоненти, які розміщені в межах організації, включаючи основну програмну систему, яка проектується. На ній також описано, як система під'єднується до зовнішніх нейронних мереж.

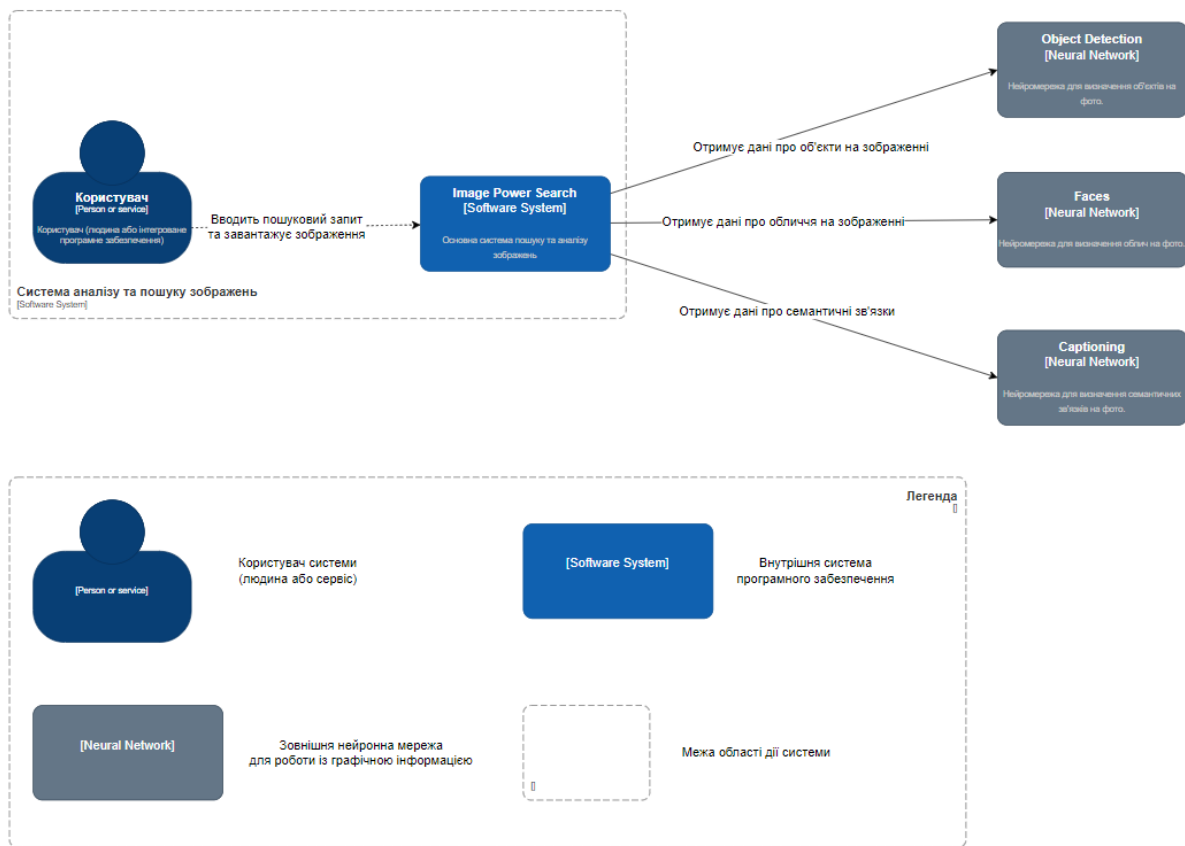


Рисунок 8 Діаграма контексту системи

Відповідно до наданого сценарію, рішення являє собою платформу пошуку та аналізу неструктурованої мультимедійної інформації. На діаграмах нижче показано, як система розділена на компоненти. Система інтегрується із неймережами, які розв'язують задачі комп'ютерного зору.

### 3.2.2 Діаграма контейнерів

Діаграма контейнерів представляє декомпозицію елементів, згаданих у попередній діаграмі. Вона також визначає технології, які використовуються для кожного елемента, і зв'язок між контейнерами.

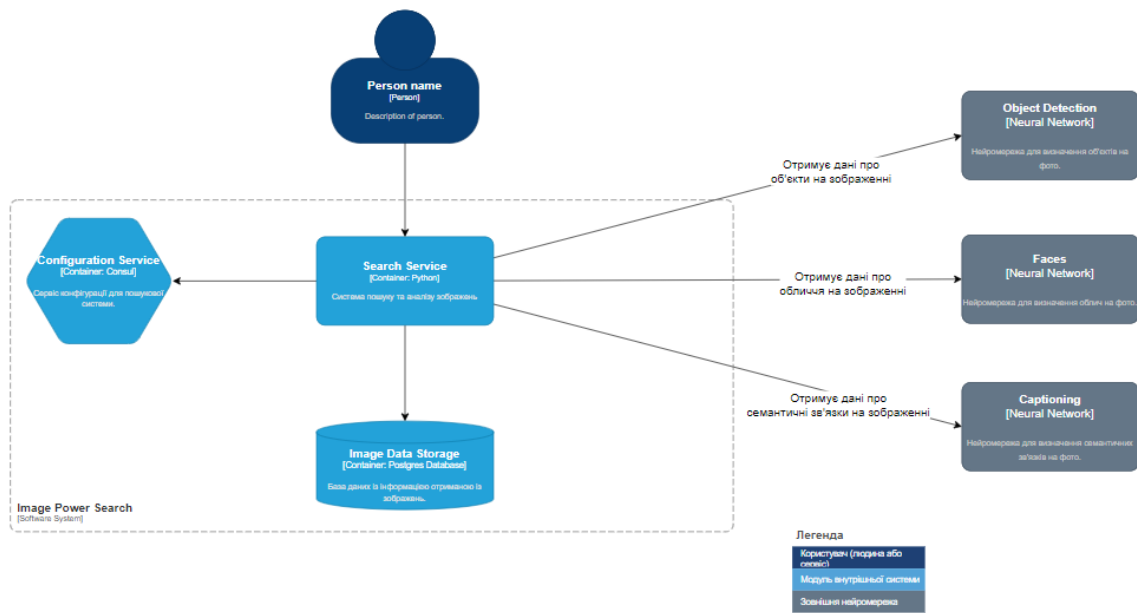


Рисунок 9 Діаграма контейнерів системи

Архітектура моноліту була застосована для рішення Image Power Search. Така архітектура дозволяє нам легко реалізувати систему відповідно до заданих вимог, легко зрозуміти із точки зору архітектури та легко масштабувати при збільшенні потоку користувачів[Understanding]. Елементи, зображені на схемі, описані нижче.

### 3.2.2.1 Search Service

Опис	Система пошуку та аналізу неструктурованої мультимедійної інформації
Технологічний стек	Python, FastAPI, psycopg2

### 3.2.2.2 Image Data Storage

Опис	База даних із такими сутностями як об'єкт на фото, персона, опис зображення, вихідне зображення.
Технологічний стек	PostgreSQL



### 3.2.2.3 Configuration service

Опис	Містить конфігурацію сервісів, що використовують нейронні мережі для задач комп'ютерного зору. Дозволяє оновлювати конфігурацію без повного розгортання системи.
Технологічний стек	Consul

### 3.2.3 Діаграма компонентів

Наведена нижче діаграма компонентів показує компоненти контейнера пошукової програми. Усі вони є класами Python, які мають тип контролера (забезпечує RESTful API), сервіси (містять бізнес-логіку) і репозиторії (рівень доступу до даних).

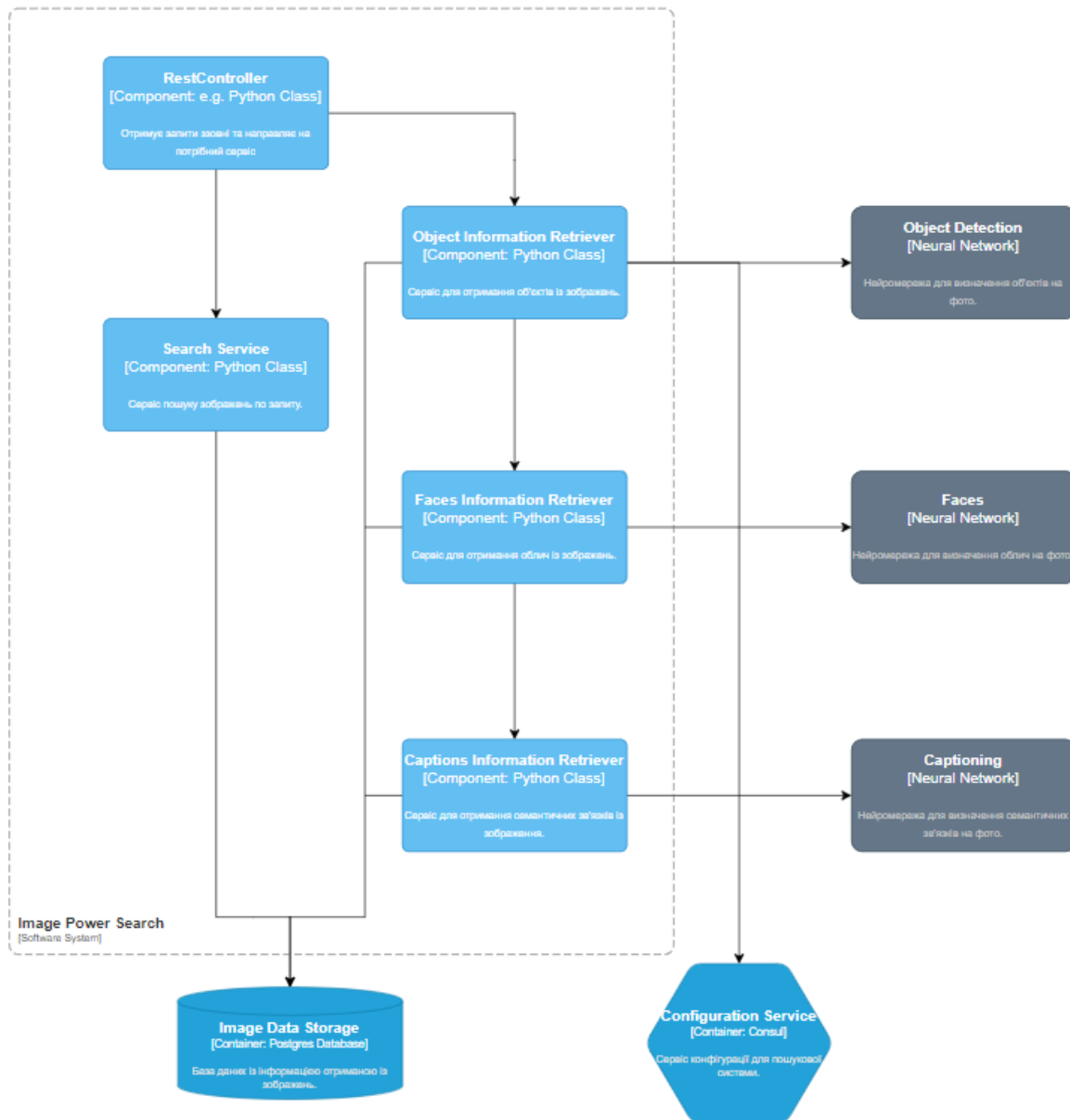


Рисунок 10 Діаграма компонентів системи пошуку та аналізу зображень

### 3.3 Принцип роботи розробленої системи

Дана система містить дві точки входу: для завантаження зображення в систему для його подальшого аналізу та для пошуку зображення за запитом користувача. Відповідно до вимог, система має програмний інтерфейс REST API, який надає доступ по протоколу HTTP як для користувача, так і для програмних продуктів, у які може інтегруватись

розроблена система. Фрагмент вихідного тексту контролера, що відповідає за обробку запитів надано у додатку А.

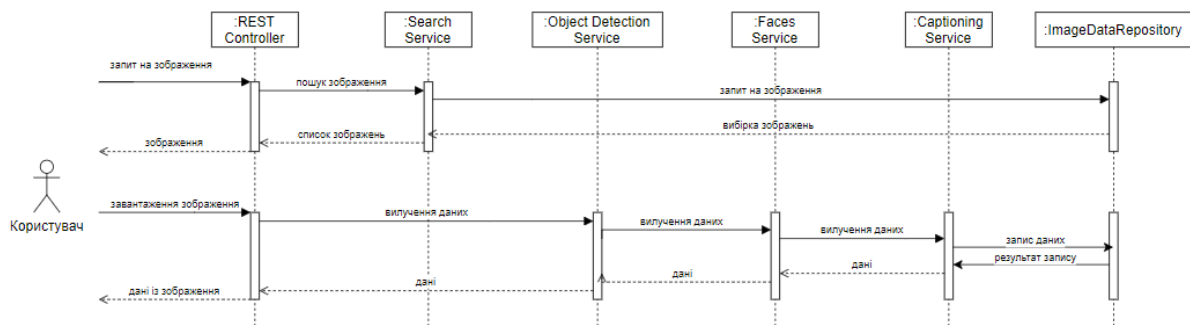


Рисунок 11 Use-case діаграма розробленої системи

При пошуку зображення за запитом користувача у довільній формі, запит скеровується на сервіс, відповідальний за пошук мультимедіа. Після цього, формується SQL запит у базу даних, у якій зберігаються зображення та вся інформація про них здобута методами комп’ютерного зору. Вибірка із зображень, що відповідають запиту, повертаються користувачу у тілі відповіді. Всі операції, що стосуються запитів до бази даних виокремлено у окремий репозиторій, частина коду якого надано у додатку Б.

Для додавання нового зображення у базу даних користувачу потрібно лише надіслати зображення у тілі запиту. Отримане зображення по черзі обробляється ObjectInformationRetriever, FacesInformationRetriever та CaptionsInformationRetriever. Коли інформація вже здобута, формується сутність та відбувається запис у базу даних. Усі сервіси, що відповідають за вилучення текстової інформації із зображення знаходяться у окремому модулі sv. Фрагменти коду продемонстровані у додатку В.

### 3.4 Порівняльна характеристика продуктивності розглянутих методів комп’ютерного зору

При порівнянні використаних моделей для розв’язання задач комп’ютерного зору були використані метрики точності та продуктивності.

Для демонстрації того, на скільки точно модель передбачає результат ми отримали метрику mAP із відкритих джерел. Для демонстрації швидкодії були використані метрики пропускну здатність та час очівання відповіді.

Mean average precision (англ. помірنا середня точність, далі mAP) є метрикою для вимірювання точності передбачення для всіх категорій об'єктів, які здатна розпізнати модель [42]. На практиці вимірюється у відсотках. Згідно із формулою, обчислюється сума середніх точностей для кожного класу та ділиться на кількість класів.

$$mAP = \frac{\sum_{k=1}^{k=n} AP_k}{n}$$

, де  $n$  - кількість класів,  $AP$  - середня точність окремого класу.

Пропускна здатність - це метрика, що показує скільки запитів чи транзакцій може безперебійно опрацювати система за одиницю часу [43]. В залежності від особливостей роботи системи, вимірюється у запитах за секунду чи хвилину (tps, rpm) або у транзакціях за секунду чи хвилину (tps, rpm).

Час очікування відповіді - це метрика, що показує час, за який запит проходить від користувача через усі компоненти інформаційної системи до сховища даних та у зворотній бік [43]. Вимірюється у секундах.

Під час проведення дослідження було проведено тестування навантаженням за допомогою інструмента Apache JMeter. При налаштуванні було додано такі слухачі як: «Підсумковий звіт» та «Графік часу отримання відповіді». Результати тестування у вигляді метрики часу отримання відповіді надано нижче.

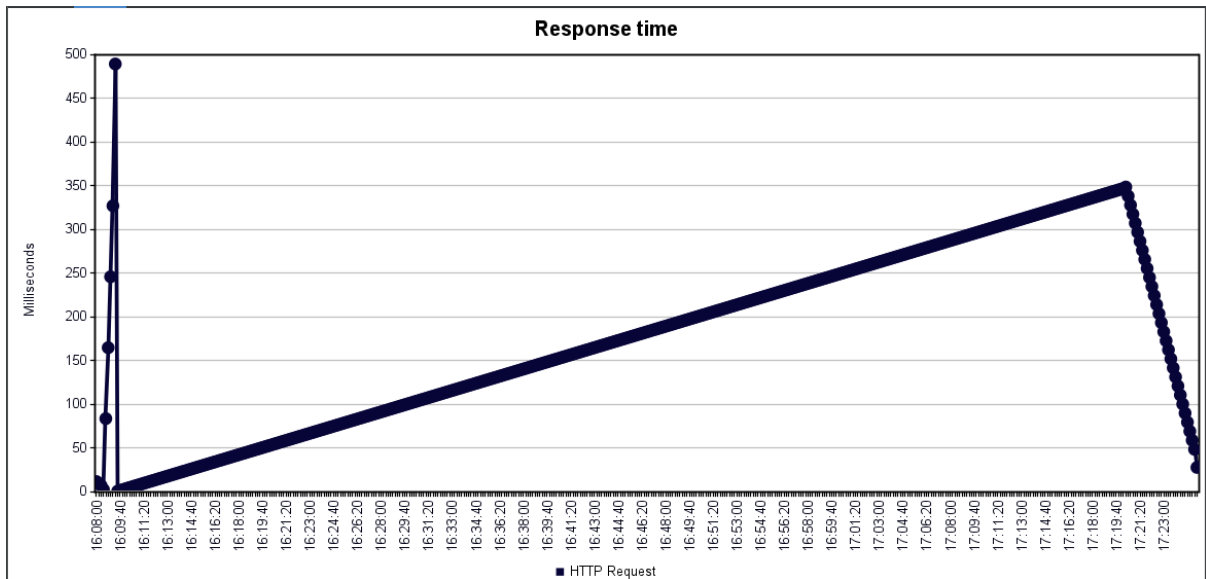


Рисунок 12 Графік часу отримання відповіді при використанні моделі YOLOv4

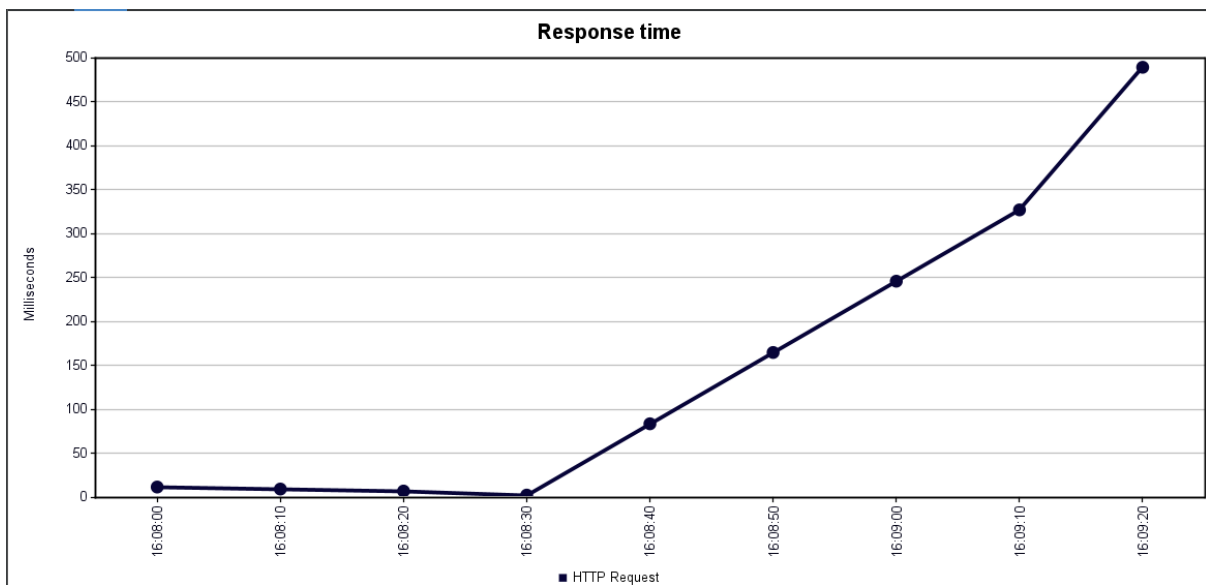


Рисунок 13 Графік часу отримання відповіді при використанні моделі Faster-RCNN

Окрім вимірювання пропускної здатності системи при зростаючому навантаженні, ми також вказали і значення помірної середньої точності передбачення моделі, отриману із відкритих джерел. При виборі моделі комп'ютерного зору завжди постає питання у виборі між точністю та швидкістю роботи в залежності від поставлених вимог.

**Таблиця 2. Порівняльна характеристика точності та продуктивності розглянутих методів виявлення об'єктів**

Назва методу	Точність	Пропускна здатність(95%)	Час отримання відповіді
YOLOv4	52.7%	28.9 fps	350 ms
Faster-RCNN	73.2%	20.4 fps	500 ms

### **Висновки до розділу 3**

У цьому розділі був проведений всебічний огляд розробленої системи пошуку та індексування зображень, яка використовує передові методи комп'ютерного зору. Досліджено принцип роботи та архітектуру системи, надаючи уявлення про її основні механізми. Система складається з різних програмних модулів, кожен з яких служить певній меті та сприяє загальному процесу пошуку та індексування. Завдяки огляду, проведеному в останньому підрозділі, отримано глибше розуміння того, як методи комп'ютерного зору інтегровані в систему, підвищують ефективність і точність завдань пошуку зображень.

### **ВИСНОВКИ**

Дана робота була зосереджена на дослідженні систем пошуку зображень і використання методів комп'ютерного зору для ефективного пошуку та індексування. Дослідження розпочалося із огляду мультимедійної інформації та систем пошуку, підкреслюючи важливість пошуку зображень у ширшому мультимедійному контексті. Згодом були досліджені потреби та вимоги користувачів до систем пошуку зображень,

що дало розуміння факторів, які сприяють ефективній взаємодії з користувачем.

Робота продовжилась дослідженням існуючих систем пошуку зображень, включаючи Google Photos, Apple Photos, Amazon Photos, Microsoft Photos і Flickr. Кожна система була проаналізована, враховуючи її характеристики, функціональні можливості та досвід користувача. Було визначено переваги та недоліки цих систем, що створило основу для запропонованих рішень для вирішення існуючих проблем.

Згодом було досліджено методи вирішення проблем комп'ютерного зору, особливо зосереджуючись на техніках виявлення об'єктів і використання текстового опису для пошуку зображень. Ці методи комп'ютерного зору забезпечили основу для подальшої розробки та перегляду нової системи пошуку та індексування зображень.

Розроблена система, що об'єднує методи комп'ютерного зору, пройшла комплексний огляд. Її принцип роботи, архітектура та програмні модулі були детально розглянуті, демонструючи, як методи комп'ютерного зору підвищують ефективність і точність завдань пошуку зображень.

Крім того, було проведено порівняльний аналіз різних методів розв'язання задач комп'ютерного зору та оцінка їхніх характеристик продуктивності. Ця оцінка дала цінну інформацію для практиків і дослідників у галузі комп'ютерного зору.

На закінчення, цією працею ми зробили внесок у розуміння та вдосконалення систем пошуку зображень. Запропоновані рішення, отримані знання та розроблена система пошуку та індексування зображень пропонують практичну інформацію для покращення функціональності, ефективності та взаємодії з користувачами систем пошуку зображень.

Висновки та рекомендації, представлені в цій роботі, є цінними ресурсами для дослідників, розробників, та професіоналів галузі, які беруть участь у розробці та впровадженні систем пошуку зображень.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Sapino, M.L., Candan, K.S. (2008). Multimedia Information Systems. In: Furht, B. (eds) Encyclopedia of Multimedia. Springer, Boston, MA.
2. Multimedia Електронний ресурс. Доступ 23.05.2023.  
<https://en.wikipedia.org/wiki/Multimedia>
3. What is Multimedia Data? Електронний ресурс. Доступ 23.05.2023.  
<https://dev.to/dimanikulin/what-is-multimedia-data-111f>
4. Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys
5. Lew, M. S., Sebe, N., Djeraba, C., & Jain, R. (2006). Content-based multimedia information retrieval: State of the art and challenges. ACM Transactions on Multimedia Computing, Communications, and Applications
6. Sivic, J., & Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In Proceedings of the Ninth IEEE International Conference on Computer Vision
7. Smeulders, A. W., Worring, M., Santini, S., Gupta, A., & Jain, R. (2000). Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence
8. Beginner's guide to Google Photos Електронний ресурс. Доступ 24.05.2023. <https://www.androidauthority.com/google-photos-937968/>
9. Перегляд фото в програмі «Фотографії» на iPhone Електронний ресурс. Доступ 24.05.2023.  
<https://support.apple.com/uk-ua/guide/iphone/iph3d267610/ios>
10. How to Use Amazon Photos: A Complete Guide Електронний ресурс. Доступ 24.05.2023.  
<https://www.makeuseof.com/how-to-use-amazon-photos/>



11. Microsoft Photos Review Електронний ресурс. Доступ 24.05.2023.  
<https://www.pcmag.com/reviews/microsoft-photos-for-windows-10>
12. Flickr Review Електронний ресурс. Доступ 24.05.2023.  
<https://www.pcmag.com/reviews/flickr>
13. An analysis on computer vision problems Електронний ресурс. Доступ 26.05.2023.  
<https://medium.com/deep-dimension/an-analysis-on-computer-vision-problems-6c68d56030c3>
14. Richard Szeliski, “Computer Vision: Algorithms and Applications 2nd Edition”, Final draft, September 30, 2021
15. Object Detection – Part 1: Introduction Електронний ресурс. Доступ 26.05.2023. <https://appliedsingularity.com/2021/05/18/object-detection/>
16. Як працює OSINT-розвідка? Від бізнес-аналізу до оборони України. Електронний ресурс. 26.05.2023.  
<https://www.issp.training/post/yak-pratsyuyeye-osint-rozvidka-vid-biznes-analizu-do-oborony-ukrayiny>
17. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis.
18. Bashir, F., & Mateen, A. (2020). Object detection and tracking in military operations: A review. *International Journal of Advanced Computer Science and Applications*
19. Zhengxia Zou, Keyan Chen, Zhenwei Shi, Member, IEEE, Yuhong Guo, and Jieping Ye, Fellow, IEEE (2022) “Object Detection in 20 Years: A Survey”
20. What is Face Detection the Ulitmate Guilde Електронний ресурс. Доступ 26.05.2023.  
<https://learnopencv.com/what-is-face-detection-the-ultimate-guide/>

21. Cheney, Jordan & Klein, Ben & Jain, Anil & Klare, Brendan. (2015). Unconstrained Face Detection: State of the Art Baseline and Challenges.
22. Image Captioning Электронный ресурс. Доступ 27.05.2023. <https://paperswithcode.com/task/image-captioning>
23. Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan. (2015) “Show and Tell: A Neural Image Caption Generator”
24. Image Captioning Электронный ресурс. Доступ 27.05.2023. <http://shikib.com/captioning.html>
25. Jeff Tyree, and Art Akerman (2005) “Architecture Decisions: Demystifying Architecture”
26. “Systems and software engineering — Architecture description” (ISO/IEC/ IEEE 42010, ), First edition, 2011
27. Len Bass, Paul Clements, and Rick Kazman (2012) “Software Architecture in Practice”, Third Edition
28. International Institute of Business Analysis, “A Guide to the Business Analysis Body of Knowledge”, Version 2.0, 2005
29. Philippe Kruchten, “Common Misconceptions about Software Architecture”, in *Rational Software*, 2001
30. William G. Wood, “A Practical Example of Applying Attribute-Driven Design (ADD), Version 2.0”, Carnegie Mellon University, 2007
31. “Systems and software engineering — Recommended practice for architectural description of software-intensive systems” (ISO/IEC 42010, ) First edition, 2007
32. Philippe Kruchten. (1995) “Architectural Blueprints—The “4+1” View Model of Software Architecture”
33. 4+1 Architectural View Model. Электронный ресурс. Доступ 29.05.2023 <https://codeopinion.com/41-architectural-view-model/>

34. Andrea Vázquez-Ingelmo, Alicia García-Holgado, Francisco J. García-Peñalvo. (2020) “C4 model in a Software Engineering subject to ease the comprehension of UML and the software”
35. C4 Model Электронный ресурс. Доступ 29.05.2023.  
<https://c4model.com>
36. Felix Bachmann, Len Bass, Paul Clements, David Garlan, James Ivers, Reed Little, Robert Nord, Judith Stafford. (2002) “Documenting Software Architecture: Documenting Interfaces”
37. Nick Rozanski and Eoin Woods. (2005) “Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives”
38. Rick Kazman, and Len Bass. (2005) “Categorizing Business Goals for Software Architectures”
39. Paul Clements, Len Bass. (2010) “Relating Business Goals to Architecturally Significant Requirements for Software Systems”
40. Len Bass, Robert Nord, William Wood, David Zubrow. (2006) “Risk Themes Discovered Through Architecture Evaluations”
41. Understanding Monolithic Architectures: Benefits, Scaling, and Pain Points. Электронный ресурс. Доступ 30.05.2023.  
<https://benjiv.com/understanding-monolithic-architecture/>
42. Breaking Down mAP Электронный ресурс. Доступ 30.05.2023.  
<https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52>
43. Martin Fowler. (2003) “Patterns of Enterprise Application Architecture”

## ДОДАТКИ

Додаток А Вихідний текст REST контролера

```
from fastapi import FastAPI, File
from typing import Annotated

import cv
import repository

app = FastAPI()

@app.post("/upload-image")
def upload_image(image: Annotated[bytes, File()]):
    return cv.retrieve_information(image)

@app.patch("/who-is")
def name_a_person(person_id: int, name: str):
    return repository.alter_person(person_id, name)

@app.get("/search")
def search_image(query: str):
    return repository.search(query)

@app.get("/image/{image_id}")
def get_image(image_id: int):
    return repository.get_image(image_id)
```

Додаток Б Фрагмент вихідного тексту репозиторія

```
def save_information(image, image_model):
    # insert image
    insert_image(image)

    # get id of the image
    image_id = get_last_image_id()

    # insert captions with image id
    insert_caption(image_id, image_model)

    # insert objects
    insert_objects(image_model)
    # get ids of inserted objects
    object_ids = get_last_object_ids(image_model)

    # insert objects ids with image id
    insert_into_objects_images(object_ids, image_id)
```

```

# insert faces
insert_faces(image_model)

# get faces ids
face_ids = get_last_face_ids(image_model)
print("FACE_IDS" + str(face_ids))

# insert faces ids with image id
insert_into_faces_images(face_ids, image_id)

cursor.close()
conn.close()
return None

def alter_person(person_id, name):
    update_query = 'UPDATE "faces" SET "person"=%s WHERE "id"=%s;'
    data = (name, person_id)
    cursor.execute(update_query, data)

    conn.commit()
    cursor.close()
    conn.close()
    return {"Result": "The " + name + " was assigned to person #" + str(person_id)}

def search(query):
    data = ('%' + query + '%',)

    captions_search_query = "SELECT \"images\".\"id\" FROM \"captions\" |
        \"INNER JOIN \"images\" ON \"captions\".\"image_id\" = \"images\".\"id\" |
        |
        \"WHERE \"captions\".\"caption\" LIKE %s;"

    cursor.execute(captions_search_query, data)
    conn.commit()
    image_ids = cursor.fetchall()

    objects_search_query = "SELECT \"images\".\"id\" FROM \"objects\" |
        \"INNER JOIN \"objects_images\" ON \"objects\".\"id\" =
|\"objects_images\".\"object_id\" |
        \"INNER JOIN \"images\" ON \"objects_images\".\"image_id\" =
|\"images\".\"id\" |
        \"WHERE \"objects\".\"object\" LIKE %s;"

    cursor.execute(objects_search_query, data)
    conn.commit()
    image_ids.extend(cursor.fetchall())

```

```

faces_search_query = "SELECT \"images\".\"id\" FROM \"faces\" |
                    \"INNER JOIN \"faces_images\" ON \"faces\".\"id\" =
\"faces_images\".\"face_id\" |
                    \"INNER JOIN \"images\" ON \"faces_images\".\"image_id\" =
\"images\".\"id\" |
                    \"WHERE \"faces\".\"person\" LIKE %s;"

cursor.execute(faces_search_query, data)
conn.commit()
image_ids.extend(cursor.fetchall())

image_ids = [image_id[0] for image_id in image_ids]

image_ids = list(set(image_ids))

image_links = ["http://localhost:8000/image/" + str(image_id) for image_id in image_ids]

return {
    "results": image_links
}

```

```

def get_image(image_id):
    get_query = "SELECT \"binary\" FROM \"images\" WHERE \"id\"=%s;"
    data = (image_id,)

    cursor.execute(get_query, data)
    conn.commit()
    image = cursor.fetchone()[0]

    return Response(content=image.tobytes(), media_type="image/jpeg")

```

### Додаток В Фрагмент вихідного тексту модуля cv

```

class CaptionsInformationRetriever(BaseInformationRetriever):
    name = 'captions'

    def retrieve(self, image_file: UploadFile):
        try:
            model = hub.load("https://tfhub.dev/google/show_and_tell/1")

            image = cv2.imread(image_file)
            image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

            image_tensor = tf.convert_to_tensor(image)
            image_tensor = tf.expand_dims(image_tensor, axis=0)

            captions = model(image_tensor)
            caption = captions['captions'][0].numpy().decode()

```

```
    results = {caption}
except:
    results = []
```

```
return results
```

```
retrievers = {ObjectInformationRetriever(), FacesInformationRetriever(),
CaptionsInformationRetriever()}
```

```
def retrieve_information(image):
    image_model = ImageModel()
    for retriever in retrievers:
        setattr(image_model, retriever.name, retriever.retrieve(image))

    repository.save_information(image, image_model)

    return image_model
```