

Міністерство освіти і науки України
 Національний технічний університет
 «Дніпровська політехніка»
 Інститут електроенергетики
 (інститут)
 факультет інформаційних технологій
 (факультет)
 Кафедра інформаційних технологій та комп'ютерної інженерії
 (повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
 кваліфікаційної роботи ступеня магістра
 (бакалавра, спеціаліста, магістра)
 студента Клеймьонова Аделіна Валентинівна
 (ПІБ)
 академічної групи 126м-22з-1
 (шифр)
 спеціальності 126 «Інформаційні системи та технології»
 (код і назва спеціальності)
 за освітньо-професійною програмою
 (за наявності)
 «Інформаційні системи та технології»
 (офіційна назва)
 на тему Автоматизована організація фотоколекцій з використанням інтелектуальних алгоритмів
 (назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Каштан В.Ю.			
розділів:				

Рецензент				
-----------	--	--	--	--

Нормоконтролер	проф. Коротенко Г.М.			
----------------	----------------------	--	--	--

Дніпро
2023

ЗАТВЕРДЖЕНО:

завідувач кафедри

інформаційних технологійта комп'ютерної інженерії

(повна назва)

Гнатушенко В.В.

(підпис)

(прізвище, ініціали)

« _____ » _____ 2023 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня магістр
 (бакалавра, спеціаліста, магістра)

студенту Клеймьонова А.В. академічної групи 126М-22з-1
 (прізвище та ініціали) (шифр)

спеціальності 126 «Інформаційні системи та технології»

за освітньою-професійною програмою _____
 (за наявності)

«Інформаційні системи та технології»

на тему Автоматизована організація фотоколекцій з використанням інтелектуальних алгоритмів.

затверджену наказом ректора НТУ «Дніпровська політехніка» від 09.10.2023 р. № 1228-с

Розділ	Зміст	Термін виконання
Розділ 1	Аналіз теми та постановка задачі	02.10.2023 – 23.10.2023
Розділ 2	Огляд методів автоматизованого управління транспортними потоками	24.10.2023 – 01.11.2023
Розділ 3	Експериментальна частина. Підготовка матеріалів для захисту роботи	02.11.2023 – 30.11.2023

Завдання видано

(підпис керівника)

В.Ю. Каштан

(прізвище, ініціали)

Дата видачі 02.10.2023 р.

Дата подання до екзаменаційної комісії

01.12.2022 р.

Прийнято до виконання

(підпис студента)

Клеймьонова А.В.

(прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 64 стор., 17 рис., 2 додатки, 19 джерел.

Об'єктом дослідження є цифрові фотоколекції, які зберігаються та управляються користувачами в ОС Windows та Linux.

Предметом дослідження є методи та інструменти для автоматизованої організації фотоколекцій з використанням інтелектуальних алгоритмів та технологій обробки зображень.

Метою дослідження є розробка та впровадження ефективних методів та інструментів для автоматизованої організації цифрових фотоколекцій з метою полегшення пошуку, сортування, відбору та подальшого використання фотографій. Головною метою є покращення користувацького досвіду в управлінні великими обсягами фотографій, а також зменшення необхідності в ручному втручанні та організації.

Для досягнення мети дослідження визначаються наступні задачі:

- аналіз існуючих методів та інструментів організації фотоколекцій із залученням інтелектуальних алгоритмів та технологій обробки зображень;
- розробка нових алгоритмів та інструментів для автоматизованої каталогізації та сортування фотографій, враховуючи їхню роздільну здатність, контент, час зйомки тощо;
- вивчення можливостей використання машинного навчання для покращення процесів організації фотоколекцій та пошуку фотографій;
- реалізація та тестування розроблених методів та інструментів на реальних фотоколекціях з метою оцінки їхньої ефективності та продуктивності.

Ключові слова: організація фотографій, цифрова фотографія, технології обробки зображень, сортування фотографій, роздільна здатність, каталогізація фотографій, користувацький досвід.

ABSTRACT

Explanatory note: 64 pages, 17 figures, 2 appendices, 19 sources.

The object of research is digital photo collections stored and managed by users in Windows and Linux.

The subject of the study is methods and tools for the automated organization of photo collections using intelligent algorithms and image processing technologies.

The purpose of the study is to develop and implement effective methods and tools for the automated organization of digital photo collections to facilitate the search, sorting, selection, and further use of photos. The main goal is to improve the user experience in managing large volumes of photos, as well as to reduce the need for manual intervention and organization.

To achieve the research goal, the following tasks are defined:

- analysis of existing methods and tools for organizing photo collections using intelligent algorithms and image processing technologies;
- development of new algorithms and tools for automated cataloging and sorting of photos, taking into account their resolution, content, time of shooting, etc;
- studying the possibilities of using machine learning to improve the processes of organizing photo collections and searching for photos;
- implementing and testing the developed methods and tools on real photo collections to evaluate their efficiency and productivity.

Keywords: photo organization, digital photography, image processing technologies, photo sorting, resolution, photo cataloging, user experience.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Опис предметної області	9
1.2 Виклики автоматичної організації фотографій	10
1.3 Ієрархічна організація фотоколекцій	11
1.4 Аналіз проблематики	13
1.5 Постановка задачі	15
1.6 Висновки до розділу 1	16
РОЗДІЛ 2. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ НА ОСНОВІ ІНТЕЛЕКТУАЛЬНИХ АЛГОРИТМІВ ДЛЯ ОРГАНІЗАЦІЇ ФОТОКОЛЕКЦІЙ	18
2.1 Програмні інструменти організації колекцій фотографій	18
2.2 Методи кластеризації зображень	20
2.3 Тегування фотографій	23
2.4 Автоматична кластеризація фотографій	26
2.5 Інтелектуальні алгоритми групування фотографій	29
2.6 Висновки до розділу 2	33
3 РЕАЛІЗАЦІЯ АЛГОРИТМУ АВТОМАТИЗОВАНОЇ ОРГАНІЗАЦІЇ ФОТОКОЛЕКЦІЙ	34
3.1 Опис алгоритму	34
3.2 Програмна реалізація алгоритму	40
3.3 Інструкція користувача	41
3.3.1 Встановлення програми	42
3.3.2 Функціональні можливості програми	42
3.3.3 UML-діаграма взаємодії користувача з програмою	43
3.3.4 Обмеження роботи з програмою	45
3.3.5 Завершення роботи з програмою	45
3.4 Експериментальні результати	45
ВИСНОВКИ	53
ПЕРЕЛІК ПОСИЛАНЬ	54
Додаток А. Фрагмент лістингу програми	56
Додаток Б. Фрагменти тестування програми	62

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ОС – операційна система.

OpenCV – Open Source Computer Vision.

DBSCAN – Density-Based Spatial Clustering of Applications with Noise.

.

.

ВСТУП

Сучасний світ зазнає стрімкого розвитку в галузі інформаційних технологій, що призводить до великого обсягу цифрової інформації, яку генерують та зберігають користувачі. Величезна кількість фотографій, які сьогодні зафіксовані, створює потребу в ефективному управлінні та організації цих фотоколекцій. Процеси зберігання, пошуку та аналізу цифрових фотографій стають більш важливими, і вимагають інноваційних підходів для автоматизації та поліпшення користування фотографічними ресурсами.

Велика кількість фотографій ускладнює виконання цих завдань. Наприклад, пошук конкретної фотографії серед тисяч інших може бути часозатратним та вимагати значних зусиль від користувача. Аналіз та відбір фотографій для спільного використання також може стати непростю задачею, оскільки люди надають перевагу мінімальним, але значущим наборам фотографій. Існує потреба в розвитку інтелектуальних алгоритмів, які допоможуть вирішити ці завдання та покращити організацію фотоколекцій.

У даній кваліфікаційній роботі розглядається проблема автоматизованої організації фотоколекцій з використанням інтелектуальних алгоритмів. Головною метою роботи є створення ефективного та інноваційного інструменту для автоматичного каталогізування та управління цифровими фотографіями з метою полегшення пошуку та використання цих фотографій.

У процесі дослідження розглядаються сучасні підходи та методи, пов'язані з організацією фотоколекцій, використовуючи інтелектуальні алгоритми. Також вивчаються технології обробки зображень, комп'ютерного бачення та машинного навчання, що можуть бути застосовані для вдосконалення процесів управління фотографічними даними.

Висвітлюються практичні аспекти застосування розробленого інструменту в реальних умовах та проводяться експерименти для оцінки

ефективності та продуктивності системи. Результати дослідження вказують на можливість покращення організації фотоколекцій за допомогою інтелектуальних алгоритмів та вказують на перспективи подальшого розвитку даного напрямку досліджень.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Фотоколекції представляють собою сукупності цифрових фотографій, які зберігаються та управляються на електронних пристроях, комп'ютерах чи в хмарних сховищах. Фотоколекції можуть включати зображення різних форматів та тематики, і їхній обсяг може значно зростати з часом. Організація фотоколекцій стає важливою задачею для забезпечення доступу до потрібних фотографій та підвищення користувацького комфорту [1].

Сучасний розвиток технологій цифрової фотографії дозволяє користувачам легко створювати та зберігати велику кількість цифрових фотографій. Смартфони, цифрові камери та інші пристрої стали дуже доступними, і це сприяє збільшенню кількості фотографій, які зберігаються користувачами. Зростаюча кількість фотографій викликає потребу в ефективній організації цих фото колекцій.

Організація фотоколекцій грає важливу роль у забезпеченні доступу до потрібних зображень та полегшенні використання фотографій. Вона дозволяє користувачам швидко знаходити фотографії за темою, датою, місцем тощо. Крім того, організація фотоколекцій створює зручність у створенні альбомів, історій та спільного використання зображень з друзями та родиною. Роль організації фотоколекцій важлива для покращення користувацького досвіду та вирішення викликів, пов'язаних із зростанням кількості фотографій.

Важливо відзначити, що фотоколекції відрізняються від інших форм зберігання фотографій, таких як альбоми чи фотоархіви. У відмінність від аналогових альбомів, де фотографії фіксовані на папері, фотоколекції є цифровими та можуть зберігатися на електронних пристроях, завдяки чому користувачі можуть легко керувати та ділитися ними в онлайн-середовищі. Також, фотоколекції можуть включати значно більший обсяг фотографій, ніж традиційні фотоальбоми чи архіви.

Фотоколекції вражають своєю мультиформатністю. Вони можуть включати фотографії у різних форматах, таких як JPEG, PNG, RAW та інші. Кожен з цих форматів може мати свої особливості щодо роздільної здатності, якості зображення та обсягу файлу. Важливо враховувати, що різниця у форматах впливає на якість та займає місце на пристрої.

З впровадженням онлайн-сховищ і соціальних мереж фотоколекції стали доступними не лише для особистого використання, а й для спільного використання з іншими користувачами. Фотографи та фотолюбители можуть легко ділитися своїми творами, створювати колективні альбоми та спільні фотоісторії. Ця можливість робить фотоколекції важливим інструментом для комунікації та співпраці у цифровому середовищі.

1.2 Виклики автоматичної організації фотографій

Автоматична організація фотографій може бути складною проблемою. Виконуючи це завдання вручну, люди покладаються на зміст фотографій і, можливо, на свої знання про подію, щоб визначити контекст потоку і зробити організаційний вибір. Електроний девайс не має знань про подію і не може визначити необмежений "контекст" для довільного потоку. Визначення контексту всередині зображень залишається відкритою проблемою в галузі комп'ютерного зору залишається відкритою проблемою. Незважаючи на значний прогрес, вона все ще далека від вирішення [2].

Замість того, щоб намагатися визначити цю нематеріальну або абстрактну інформацію високого рівня, я покладаюся на наступне ключове розуміння, на якому ґрунтується ця робота. Фотографії, зроблені протягом відносно короткого проміжок часу, зроблені одним і тим же фотографом, пов'язані між собою. Було б фізично неможливо мати дві абсолютно не пов'язані між собою фотографії, зроблені протягом декількох хвилин одна від одної.

1.3 Ієрархічна організація фотоколекцій

Одним із способів організації фотографій є впорядкування їх у лінійний порядок за певною метрикою. Наприклад, це може бути сортування фотографій за датою створення, алфавітним порядком за назвою файлу чи іншими критеріями. Лінійний підхід допомагає створити порядок у колекції, але він не враховує специфічний контекст фотографій.

Іншим підходом є розміщення фотографій у структурі, що відображає ієрархію або вкладеність об'єктів. Наприклад, це може бути організація фотоколекції в каталозі користувачів з папками та підпапками. Ієрархічна структура допомагає відображати зв'язки між фотографіями, але такий підхід не завжди підходить для великих колекцій і може стати заплутаним.

Пошук фотографій та надання рекомендацій щодо подібних зображень є важливою складовою організації фотоколекцій. Пошукові системи, бази даних, рекомендаційні системи та автоматичні підсумовувачі зображень можуть бути використані для пошуку та відбору фотографій за певними критеріями [4].

Описувати фотографії і розуміти їхнє суб'єктивне відчуття для користувача є складною задачею, як для людей, так і для машин. Однак деякі системи можуть спробувати врахувати індивідуальні вподобання користувачів та пропонувати фотографії на основі їхньої схожості або репрезентативності.

Фотоколекції можуть бути загальнодоступними для багатьох користувачів або обмеженими доступом. Приватні колекції можуть переглядати лише обмежена кількість користувачів, що може впливати на підходи до організації та відображення фотографій.

Одним із підходів до організації фотоколекцій є використання автоматичних підсумовувачів зображень, які вибирають найбільш репрезентативні або схожі зображення з групи. Цей метод може спрощувати перегляд і відбір фотографій для користувачів, але він має свої виклики,

особливо коли об'єкти на фотографіях схожі між собою та можуть бути різними лише у дрібних деталях.

Фотографії можуть мати суб'єктивний характер, і описати їх об'єктивно може бути важко. Користувачі можуть шукати не лише конкретні об'єкти чи сцени, але і вирази емоцій, настроїв, атмосферу та інші аспекти, які важко визначити автоматично.

Один із викликів полягає в тому, що користувачі можуть мати різні вподобання та сприймати фотографії по-різному. Системи рекомендацій повинні враховувати індивідуальні вподобання користувачів та пропонувати фотографії, які їм можуть сподобатися.

Одним із основних викликів є обробка великих обсягів фотографій у реальному часі. Обробка та аналіз великих колекцій може вимагати великих обчислювальних ресурсів та ефективних алгоритмів.

Описувати фотографії детально для організації вимагає виявлення ключових функцій та об'єктів на зображеннях. Це може бути викликом для автоматичних систем, оскільки об'єкти можуть бути різними та знаходитися в різних сценах.

Загалом, організація фотоколекцій є складною задачею, яка вимагає врахування багатьох факторів і може бути вирішена різними способами залежно від конкретних потреб користувачів та характеристик фотоколекції. У наступних розділах дослідження ми розглянемо методи та підходи до автоматичної організації фотографій та їхню реалізацію для вирішення зазначених викликів.

1.4 Аналіз проблематики

Сьогодні ми можемо робити більше фотографій, ніж будь-коли раніше. Без певної схеми організації користувачеві стає надзвичайно складно переглядати, шукати або ділитися особистими фотографіями.

Комп'ютерна операційна система може допомогти впорядкувати файли, але зі збільшенням кількості файлів організація стає менш ефективною. З цією метою було створено кілька продуктів і дослідницьких проектів, які допомагають впорядковувати фотографії. Браузери цифрових фотографій (наприклад, Adobe Lightroom, Apple iPhoto, Google Picassa тощо) автоматично впорядковують фотографії на основі часової шкали; коли вони були зроблені, в одному вимірі (можливо, в більшому, якщо користувач витратить час на організацію глибшої структури вручну). Коли колекція стає більшою, користувач може розгубитися від кількості фотографій, оскільки такий перегляд показує всі фотографії одразу.

Переважає більшість людей на основі дослідження [5], не користується жодним із перелічених вище інструментів для роботи з фотографіями, а просто використовують файлову систему. Існує єдина папка, позначена як "фотографії", і кожна подія, або дні знімків створюється окрема під папка. Якщо зйомку ведеться кількома камерами, то кожній камері створюється під папка під подією. На рисунку 1.1 наведено схему зберігання фотографій. Загалом, це не дозволяє гнучко переглядати, ділитися або знаходити окремі зображень.

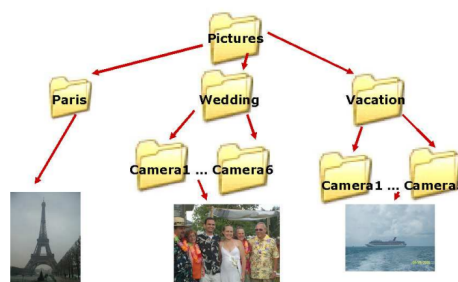


Рисунок 1.1 - Приклад типового рішення для зберігання фотографій з використанням файлової системи

Для того, щоб створити організацію, яка не базується на часі, необхідні семантичні знання про набір необхідне семантичне знання набору. Іншими словами, знання про подію: хто брав участь, що відбувалося, де відбувалося, чому це фотографували, як це пов'язано з іншими фотографіями в колекції тощо. Сучасні технології комп'ютерного зору та інші технології не надають універсального, надійного методу для автоматичного отримання цієї інформації. Скоріше, найкращий спосіб отримати найбільш семантичну семантичної інформації - це коли користувач вносить її вручну. Більшість існуючих інструментів дозволяють користувачеві надавати цю інформацію (зазвичай у вигляді тегів), однак вони вимагають, щоб кожне зображення було позначене окремо, хоча до інструментів керування фотографіями було додано кращі інтерфейси для тегування. Крім того, існуючі інструменти не можуть легко переносити метадані, надані користувачем, до інших програм, а це означає, що користувачеві доведеться надавати інформацію для кожної окремої фотографії в кожному окремому додатку.

Оскільки набір фотографій стає більш організованим, користувач повинен мати можливість використовувати цю організацію, що полегшить (або, принаймні, не ускладнить) взаємодію з колекцією. Однак надання більшої організованості наборам фотографій часто вимагає від користувача більшої попередньої роботи ніж очікувана вигода [6].

Зазначимо, що проблема, яку розглядає ця робота, схожа на ту, з якою стикаються професійні фотографи при організації великих колекцій цифрових активів. Професійні рішення існують, проте вони вимагають значних витрат часу та ресурсів. Більшість користувачів є аматорами та не мають можливості чи бажання інвестувати в професійні рішення. Методи, запропоновані в цій роботі, дозволяють досягти певного рівня організації фотографій без великих зусиль, що особливо актуально для аматорів. Професійні фотографи часто використовують ретельне маркування кожної фотографії на основі різних атрибутів, таких як час/дата, тема, тип події, параметри фотозйомки та

авторські права, проте для багатьох користувачів це є занадто часомістким та складним завданням.

1.5 Постановка задачі

Перехід з операційної системи Windows на Linux та втрата важливих фотографій через недоречну дію, пов'язану зі зміною флешки на зовнішній диск резервного копіювання – це стало поштовхом до глибшого вивчення можливостей та функціоналу терміналу Linux. Ця ситуація послужила важливим кроком у розробці власного рішення для відновлення та організації збережених фотографій.

Однак завдання не обмежується просто відновленням фотографій. Після використання інструмента Foremost для відновлення, з'являється велика кількість відновлених зображень, які включали в себе не лише особисті фотографії, але і різноманітні інші зображення, такі як мініатюри, текстури, зображення профілю та шпалери. Це створило потребу в розробці ефективного інструменту для організації цих великих обсягів відновлених фотографій.

Відновлені фотографії містилися в численних папках, тому є потреба в розподіленні зображень за папками, в кожній з яких містилося б приблизно 1000 зображень. Після ручного просіювання цих фотографій та відокремлення важливих особистих зображень від випадкових, було вирішено покращити процес подальшої організації.

Для досягнення цієї мети запропоновано розробити скрипт, який автоматизував процес групування фотографій за схожістю. Ця автоматизація спростила і полегшила роботу автору після втрати фотографій і відновлення їх.

Отже, головна задача полягає у розробці та вдосконаленні інструменту для організації та групування відновлених фотографій за схожістю з метою

полегшення роботи з великим обсягом зображень та відокремлення особистих фотографій від інших зображень.

У цій роботі запропоновано наступні завдання:

- розробка набору методів для автоматичної організації великих колекцій фотографій. Ці методи розглядаються у контексті покращення організації фотоколекцій та спрощення роботи з великим обсягом зображень;
- реалізація цих методів у вигляді нового інструменту для перегляду та інтерфейсу для фотографій. Ці інструменти можуть бути інтегровані в існуючі інструменти організації фотографій, або функціонувати як окремий браузер для перегляду фотографій;
- забезпечення початкової організаційної структури, яка є більш детальною, ніж та, що надається сучасними інструментами. Ця структура спрямована на розв'язання основного завдання - ефективного перегляду фотографій;
- забезпечення можливості користувачу взаємодіяти з фотографіями безпосередньо в організованій структурі або використовувати її як відправну точку для подальшої організації фотографій.

1.6 Висновки до розділу 1

У першому розділі кваліфікаційної роботи було проведено докладний аналіз предметної області, пов'язаної з організацією фотоколекцій. Аналіз був спрямований на встановлення ключових аспектів, які визначають складність та актуальність проблеми організації великих фотоколекцій у сучасному світі.

Описано велику кількість факторів, які впливають на організацію фотоколекцій, включаючи зростання обсягів цифрових фотографій, розвиток фототехнологій та супутні зручності для створення фотоколекцій. Визначено виклики, з якими стикаються користувачі при спробах організувати та

управляти великими обсягами фотографій, включаючи проблеми пошуку, сортування та поділу зображень.

У результаті аналізу було встановлено, що існує необхідність в розробці нових методів та інструментів, які допомагали б автоматизувати організацію фотоколекцій та спрощували процес роботи з великими обсягами фотографій. Дисертація має на меті вирішити цю проблему та надати користувачам інноваційні рішення для більш ефективної організації та управління фотографіями.

Результати аналізу предметної області вказують на актуальність та значущість дослідження, яке буде проведено в наступних розділах дисертації.

РОЗДІЛ 2. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ НА ОСНОВІ ІНТЕЛЕКТУАЛЬНИХ АЛГОРИТМІВ ДЛЯ ОРГАНІЗАЦІЇ ФОТОКОЛЕКЦІЙ

2.1 Програмні інструменти організації колекцій фотографій

Проблема організації великих колекцій фотографій давніша за цифрову фотографію. У 2002 році Фроліх та ін. [7] представили глибоке дослідження того, як 11 різних сімей організують свої колекції фотографій. Обрані сім'ї використовували як цифрову, так і традиційну фотографію. технологію, як цифрову, так і традиційну. Один із висновків дослідження полягав у тому, що організація фотографій відрізнялася між традиційними та цифровими фотографіями. Дослідження показало, що хоча обидва типи фотографій, як правило не мали великої кількості впорядкованих фотографій, цифрові фотографії були ще менш впорядковані, ніж друковані. друковані фотографії.

Ще одним висновком дослідження є припущення, що організація цифрових фотографій повинна більше наближатися до соціального досвіду. Існує кілька дослідницьких проєктів і комерційних підприємств, які допомагають споживачам працювати з великими наборами зображень, що включають соціальну взаємодію. Три такі комерційних систем як Flickr [8], Kodak Easy Share Gallery [9] та Tag World [10] – це веб-сторінки, які дозволяють користувачам завантажувати фотографії, позначати їх і ділитися ними з іншими користувачами через веб-інтерфейс. Flickr і Tag World дозволяють спільноті позначати фотографії фотографій. Це означає, що будь-хто (з дозволу) може наносити мітки на фотографію, незалежно від того, хто є її власником. Ці веб-сторінки засновані на організації фотоальбому. Тобто, користувач завантажує фотографії в певну папку або альбом. Автоматичної організації фотографій немає.

Так само існують комерційні програми для організації фотоколекцій.

Найвідомішою з них є Picasa [11], безкоштовна програма для управління фотографіями від Google. Вона призначена для зберігання фотографій та

швидкого пошуку. Adobe Photoshop Elements [12] – ще одна комерційне програмне забезпечення, яке включає в себе інструменти для зберігання та організації фотографій.

Elements покликаний допомогти користувачеві в організації фотографій; наприклад, він використовує автоматичне розпізнавання облич і дозволяє користувачеві вручну позначати кожне знайдене обличчя в наборі фотографій.

Знову ж таки, ці системи могли б виграти від додаткових методів автоматичної організації, які я представляю в цій дисертації. Ці системи покладаються на користувача для організації зображень. Не робиться спроб зменшити представлену візуальну інформацію, що робить систему менш масштабованою.

Окрім комерційних проектів, проблема роботи з великими наборами зображень залишається відкритою, яку досліджували кілька дослідників користувацьких інтерфейсів. Друкер та ін. [10] розробили MediaBrowser. У цій системі користувачі позначають окремі фотографії та відео. Система може складати тематично пов'язані набори, а також здійснювати пошук у наборі зображень. Схожою на MediaBrowser є система MiAlbum [63]. Вона використовує користувацькі мітки для того, щоб допомогти керувати цифровими фотографіями "типової сім'ї". Знову ж таки, ці системи покладаються на користувача для управління організацію.

Шафт і Рамакрішнан [13] розробили систему, яка використовує класифікатори зображень і базу даних для організації зображень. Зображення, розміщені в базі даних, мають інформацію, таку як карта країв та кольорова гістограма, яка автоматично витягується, щоб допомогти надати інформацію про фотографії.

Крім того, користувач може застосовувати мітки до об'єктів на зображенні, що дозволяє йому здійснювати запити для пошуку зображень. Це один із прикладів з поля Image Based Content Retrieval (IBCR). Основна

відмінність між роботою в цій дисертації та IBCR полягає в тому, що фотографії в IBCR пов'язані між собою через зміст фотографій. Фотографії в цій дисертації пов'язані між собою подіями, знятими одним і тим же фотографом. Наприклад, в базі даних IBCR може бути багато зображень котів, які пов'язані між собою за змістом зображення. І навпаки, якщо фотографії кількох різних тварин у зоопарку були зроблені в зоопарку, вони будуть згруповані разом в описаній мною системі, незалежно від змісту.

Кожна з наведених вище систем намагається обробити весь набір фотографій, але не робить нічого, щоб для зменшення розміру набору зображень. У проекті Photo Triage Project інтерфейс дозволяє користувачеві швидко "сортувати" свої фотографії. Фотографії представляються користувачеві у вигляді розкладеної стопки, і за допомогою швидкої взаємодії з мишею користувач може позначити фотографію як "подобається" або "не подобається". Фотографії, які не подобаються можна відкинути, а фотографії, що сподобалися, можна перемістити до якогось альбому для відображення. Після цього користувач може зосередитися на виправленні тих фотографій, які не отримали жодної позначки.

2.2 Методи кластеризації зображень

Споріднені зображення часто об'єднують у кластери. Багато систем, як дослідницьких, так і комерційних, намагаються використовувати кластеризацію для впорядкування фотографій. Ключова ідея, полягає в тому, що фотографії можна автоматично кластеризувати на декількох рівнях, щоб створити таку організацію.

Описані нижче системи намагаються розбити колекцію фотографій на окремі альбоми, створюючи дворівневу схему організації. Кластеризація, яку я пропоную, організовує фотографії у деревоподібну структуру, таким чином, створюючи набагато глибший набір кластерів.

AutoAlbum, розроблений в Microsoft Research Платтом, є системою для кластеризації фотографій. Як і в запропонованій мною роботі, вона бере часову мітку кожної фотографії для того, щоб згенерувати кластеризації. У цій схемі фотографії організовані лише на одному рівні. Однорівнева схема працює для AutoAlbum, оскільки створюються лише альбоми. створюються лише альбоми; немає концепції пошуку чи більш глибокої організації. Хоча використання середньої гистограми не є надійним методом вибору репрезентативного зображення для зменшення візуальної складності. Методи, що використовуються для автоматичної організації добре справляються зі створенням окремих альбомів, але не впорядковують зображення. Це може призвести до дуже великих альбомів, які погано масштабуються.

Луї представляє альтернативний підхід до кластеризації на основі часу та контенту [14] для автоматичного створення фотоальбомів на основі часу зйомки зображень. У його підході для створення альбомів використовується кластеризація K-Means на основі часової мітки фотографії. Комп'ютерний зір також використовується для подальшого пошуку схожих фотографій, а також для видалення погано знятих фотографій з альбомів. Загальна проблема кластеризації за методом K-середніх полягає в тому, що значення "K" має бути відоме заздалегідь, щоб запобігти утворенню неприродних зв'язків.

Подібно до Луї, кілька інших дослідників припустили, що фотографії можна кластеризувати шляхом знаходячи сплески в часовому потоці [14]. Центральна ідея кожної з цих робіт, а також є те, що цифрові фотографії робляться серіями. Це пояснюється тим, що без традиційних обмежень плівки, фотограф може зробити кілька знімків тієї самої події (або об'єкта), щоб зафіксувати подію в процесі її розгортання або переконатися, що принаймні одне зображення, яке нас цікавить, було зроблено. Graham, та ін. [16] описує це явище наступним чином: "Люди схильні робити особисті фотографії серіями. Наприклад, на дні народження може бути зроблено багато фотографій, але мало, якщо взагалі буде до наступної важливої події... Сам

того не усвідомлюючи, користувач надає структурованості своїй особистій колекції фотографій тим, як він їх робить". Роботи, представлені в [15, 16], описують як фотографії можуть бути згруповані на різних рівнях часової лінії, і цей факт я теж використовую. Автори в роботах [15], і [16] використовують ієрархію для кластеризації, і їхні методи найбільш схожі на мої. Однак є деякі відмінності в підході, який застосовують представлені мною методи, порівняно з їхньою реалізацією. У [15] для завантаження процесу кластеризації потрібна константа, тобто ця константа використовується на верхньому рівні для визначення того, де мають бути розміщені межі кластерів. Метод, що використовується в [16], вимагає налаштування трьох різних констант для визначення межі на всіх рівнях дерева. Методи, вимагають, щоб одна константа (яку я надаю) для автоматичного визначення правильних меж між кожним кластером. Інші системи, які я описав вище, не використовують ієрархію кластеризації. Швидше за все, інші системи кластеризують лише на одному рівні. Були запропоновані інші метадані, такі як глобальне місцезнаходження. У цьому випадку зображення, які знаходяться близько один до одного у фізичному просторі, ймовірно, пов'язані між собою. Цю інформацію можна бути використана в подальшому для порівняння з базою даних відомих локацій, щоб допомогти в подальшій ідентифікації та тегуванні фотографів. Недоліком цього підходу є те, що хоча GPS-локація є частиною специфікацій даних EXIF специфікацій даних EXIF, він вимагає, щоб фотограф мав певний тип GPS-системи для збору цієї інформації. для збору цієї інформації. Можливо в майбутньому камери будуть оснащені такою можливістю як стандартну функцію в майбутньому, сучасні моделі фотоапаратів, які містять функції GPS, мають дуже високу ціну. Інший підхід полягає в тому, щоб дозволити користувачеві вказати цю інформацію вручну, або у вигляді мітки, або безпосередньо на карті. Недоліком цього підходу, є те, що додавання тегів до фотографій вручну може бути складним і трудомістким процесом.

Інші дослідники представили ідеї щодо кластеризації зображень на основі візуального контенту, а не на основі метаданих. а не на основі метаданих зображення. Шаффаліцкі та Зіссерман [17] представляють систему для кластеризації зображень на основі комп'ютерного зору. На відміну від попередніх робіт у галузі комп'ютерного зору, їхня система кластеризує зображення однієї і тієї ж сцени, навіть якщо між двома зображеннями існує велика різниця, тобто вона не потребує малої базової лінії. Цей підхід добре працює для кластеризації, якщо фотограф повертається в одне і те ж місце в різні моменти часу. На рисунку 2.1 показано приклад кількох різних фотографій церкви, зроблених (під різними кутами і в різних напрямках). Їхній метод здатен об'єднати ці зображення разом, незважаючи на широку базову лінію та інші відмінності. Puzicha1 представляє поглиблене дослідження декількох різних методів комп'ютерного зору для кластеризації зображень.



Рисунок 2.1 – Зображення церкви у Вальбоні, Франція [17]

2.3 Тегування фотографій

Альтернативним підходом до кластеризації фотографій є їхнє позначення (тегування). Якщо до кожної фотографії у наборі прикріплено хоча б одна мітка (навіть якщо ця мітка "без мітки"), то існує певна організація, яку можна застосувати до організації, яку можна застосувати до набору фотографій. Багато в чому кластеризація (описана вище) є специфічною формою

маркування. Однією з проблем з етикетуванням фотографій є те, що це займає багато часу трудомісткий і виснажливий процес. Багато користувачів не захочуть витратити час на те, щоб позначати кожне окреме зображення, тому що передбачувана вигода не переважає над витратами. Таким чином, більшість досліджень у сфері маркування спрямовані на те, щоб зробити маркування автоматичним або, принаймні, більш цікавим.

Методи, представлені в цій роботі, не покладаються на маркування фотографій. Однак деревоподібна структура дозволяє створити новий інтерфейс для полегшення цього завдання. Замість того, щоб вимагати від користувача позначати фотографії індивідуально, можна позначати гілки дерева. Таке позначення можна комбінувати з будь-яким з методів, описаних нижче.

Було проведено багато досліджень щодо використання класифікаторів (наприклад, класифікаторів облич) для того, щоб маркування зображень. Вей і Сеті [18] представляють алгоритм для виявлення облич на зображеннях, який, у свою чергу, може бути використаний для маркування. в свою чергу, може бути використаний для маркування. У найновішій версії Adobe Photoshop Elements [5] інструмент фотоальбому фотоальбом містить інструмент, який знаходить всі обличчя в наборі фотографій. Після цього користувач може позначити всі обличчя окремо.

Автори Шріхарі та Чжан описують систему для напівавтоматичного тегування зображень. Вони використовують комбінацію класифікаторів зображень разом з обробкою природної мови для створення підписів. У своїй системі вони зосереджуються на медичних зображеннях, оскільки лікарі звикли диктувати інформацію про зображення. Окрім використання метаданих для кластеризації, інші дослідники застосовують для цього комп'ютерний зір. Jeon та ін. розробили систему для автоматичного маркування зображень. Система бере навчальний набір зображень,

позначених вручну. Коли вона стикається з новим зображенням, вона намагається зіставити його з навчальними даними. Недоліком цієї системи

Коли система натрапляє на нове зображення, вона намагається підібрати йому відповідність на основі тренувальних даних. Недоліком цієї системи є те, полягає в тому, що вона настільки хороша, наскільки хороша її навчальна база. Наприклад, якщо система зустрічає зображення лева, а в її навчальній базі є лише зображення лева, а в навчальній базі є лише зображення котів, вона позначить зображення лева як kota. Якщо система зустрічає зображення лева, але має лише обмежений тип навчальних даних, наприклад, архітектурні архітектурні зображення, то маркування буде помилковим. Інші автори також представили роботи з автоматичного маркування зображень [14]. маркування зображень.

У 2000 році Шнейдерман і Канг розробили систему маркування для маркування зображень методом перетягування зображень. У їхньому підході користувач обирає фотографію і мітки, а потім просто перетягує мітки на місце. Це забезпечує простий і швидкий метод маркування фотографій.

Нещодавно дослідники виявили, що якщо перетворити маркування фотографій на гру, то люди люди будуть більш охоче виконувати цей процес. Зокрема, в цій галузі можна відзначити ESP Game та Peekaboом. В обох іграх анонімні гравці об'єднуються в пари один з одним і намагаються відгадати одне й те саме слово. Роблячи це, користувачі маркують фотографії в Інтернеті. Популярність цих ігор дозволила творцям цих ігор дозволила творцям позначити мільйони зображень зі всесвітньої павутини. Така система, однак, не працює для маркування власної колекції фотографій. Мейерс та ін. описують гру для маркування особистої колекції фотографій. Вони використовують танцювальний для того, щоб мати етикетки, описані танцювальними діями. Іншими словами, фотографії дають фотографії даються етикетки за танцювальними рухами. Цей підхід обмежується відображенням танцювальних рухів на етикетки.

2.4 Автоматична кластеризація фотографій

Автоматична кластеризація фотографій є важливою галуззю обробки зображень та інтелектуального аналізу даних, яка має безліч застосувань в сучасному світі. В даному тексті розглядається суть та методи автоматичної класифікації фотографій, включаючи визначення категорій, міток або тегів, які можуть бути присвоєні фотографіям на основі їхнього вмісту. Основна мета цього процесу - спростити організацію та навігацію великих фотоколекцій, роблячи фотографії більш доступними та легкими для пошуку.

Автоматична класифікація фотографій полягає в присвоєнні одній або декільком категоріям, міткам або тегам, які відображають зміст або характеристики фотографії. Наприклад, фотографії людей можуть бути класифіковані за статтю, віком, виразом обличчя тощо. Фотографії природи можуть бути класифіковані за видами рослин або тварин, типами ландшафтів та іншими параметрами. Класифікація дозволяє ефективно організовувати фотоколекції та швидко знаходити фотографії за певними критеріями.

Фотографії часто знімаються серіями, і їх можна автоматично згрупувати в підгрупи, шукаючи ці серії. Як зазначають дослідники, така поведінка не повинна бути несподіваною; фотограф часто намагається зробити кілька знімків чогось, що його цікавить, або щоб зафіксувати всю подію, як вона розгортається, або щоб зробити хоча б один гарний знімок об'єкту зйомки. об'єкта. Дослідники, вказують на те, що цей патерн серій існує на різних рівнях часової лінії [16].

Одним із ключових аспектів автоматичної класифікації фотографій є визначення схожості та відмінностей між ними. Для цього застосовуються різні алгоритми та методи, які дозволяють визначити, наскільки фотографії схожі одна на одну. Наприклад, алгоритми порівняння зображень можуть визначати ступінь схожості за певними характеристиками, такими як колір,

текстура, форма та інші. Це допомагає встановити відношення між фотографіями та створити кластери схожих зображень.

Дослідження, представлене в [13], показує, що хоча більшість людей хотіли б упорядкувати свою особисту колекцію фотографій (як друкованих, так і цифрових), вони часто не знаходять часу, щоб виконати це завдання, якщо тільки немає якоїсь конкретної функції, що примушує до цього (наприклад, доручення). Що стосується великих колекцій цифрових фотографій загалом, Родден і Вуд представляють дослідження, в якому зроблено висновок, що більшість фотографів в якому робиться висновок, що більшість фотографів не відчують, що зусилля з організації фотографій окупаються порівняно з необхідною роботою. Незалежно від того, чи це брак часу, мотивації, чи поєднання чи то брак часу, чи мотивації, чи поєднання цих двох чинників, але більшість людей дуже мало впорядковують свої фотографії.

Деревоподібна організація (рис.2.2) - це один із способів, у який люди зрештою хотіли б упорядкувати свої фотографії, якби мали на це час. Розглянемо друкований альбом. Мотивована людина може створити альбом не лише зі сторінками, але й розділами, підрозділами тощо. З правильно оформленими роздільниками, перегляд такого фотоальбому буде дуже схожий на навігацію по колекції в деревовидною структурою. Недоліком організації фотографій у вигляді дерева є те, що вона може бути нудною і забирати багато часу. так само, як і створення складного фізичного фотоальбому. Це дає переваги дерева (описані вище) без жодних витрат, пов'язаних зі створенням такої структури.

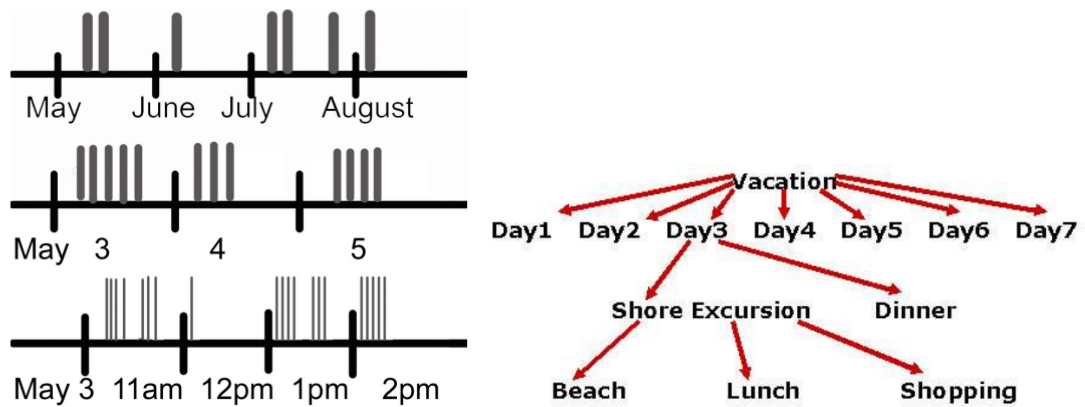


Рисунок 2.2 – Деревоподібна організація даних

Нижче наведено опис алгоритму автоматичної кластеризації фотографій у потоку S :

1. Відсортуйте S за часом, коли було зроблено кожну фотографію (починаючи з першої зробленої). За замовчуванням фотографії зберігаються на карті пам'яті у такий спосіб, а також організовані у більшості операційних систем, тому цей крок можна пропустити.

2. Визначити середню відстань у часі (t_{avg}) між кожною наступною фотографією у S . Тобто визначити середній проміжок часу між знімками. Математично знаходиться за формулою:

$$t_{avg} = \frac{(T_2 - T_1) + (T_3 - T_2) + \dots + (T_z - T_{z-1})}{z}, \quad (2.1)$$

де T_i час, коли було зроблено i -те зображення, а z - кількість знімків у S .

3. Створіть границю кластера між будь-якими двома послідовними фотографіями, середній час між якими більший за середній у t разів. Іншими словами, якщо $T_i - T_{i-1} < t_{avg} \times t$, то між T_{i-1} і T_i існує межа, і вони поміщаються в окремі кластери.

4. Щоб побудувати дерево, виконайте кроки 2 і 3 для кожного кластера. Зупиніться, коли кожна фотографія опиниться у власному кластері, тобто

листку, або коли кількість фотографій у кластері стане "достатньо малою", залежно від програми (тобто кожна фотографія у кластері може поміститися на екрані дисплея).

Одним із ключових аспектів автоматичної класифікації фотографій є визначення схожості та відмінностей між ними. Для цього застосовуються різні алгоритми та методи, які дозволяють визначити, наскільки фотографії схожі одна на одну. Наприклад, алгоритми порівняння зображень можуть визначати ступінь схожості за певними характеристиками, такими як колір, текстура, форма та інші. Це допомагає встановити відношення між фотографіями та створити кластери схожих зображень.

2.5 Інтелектуальні алгоритми групування фотографій

Інтелектуальні алгоритми групування фотографій на кластери грають важливу роль у вирішенні завдань автоматичної організації фотоколекцій. Групування фотографій полягає в розділенні фотографій на підмножини, які мають спільні ознаки або схожий контент. Це допомагає користувачам легше орієнтуватися в колекції, знаходити схожі зображення та шукати необхідні фотографії.

Один із інтелектуальних алгоритмів для групування фотографій - це алгоритм кластеризації, такий як k-середніх або ієрархічна кластеризація. K-середніх шукає k центрів кластерів та розподіляє фотографії до найближчого кластера. Цей метод допомагає створити кластери з фотографій, які мають схожі характеристики, такі як кольори, текстури чи форма [19].

Інший підхід - це агломеративна ієрархічна кластеризація, де фотографії поступово об'єднуються в більші кластери на основі схожості. Цей метод дозволяє створити ієрархію кластерів, що відображає різні рівні схожості між фотографіями.

Значним по кроком в автоматичній групуванні фотографій є використання машинного навчання та нейронних мереж. Моделі глибокого навчання можуть навчитися виявляти схожість між фотографіями на основі вмісту, а не тільки на основі структурних ознак. Вони враховують більше аспектів зображення, що дозволяє створювати більш точні та інформативні кластери.

Інші інтелектуальні методи включають в себе аналіз зображень з використанням методів комп'ютерного зору, визначення схожості на основі контенту та метаданих фотографій, а також врахування контексту та відносин між фотографіями.

Ще одним способом використання є пакетна анотація. Враховуючи набір фотографій, ручна анотація кожної фотографії буде трудомістким процесом і марною тратою зусиль, оскільки багато фотографій схожі між собою. Наприклад, часто роблять багато знімків, що зазвичай описують ту саму сцену чи об'єкт. Пакетне анотування – це ефективний підхід для зниження витрат, оскільки передбачає безпосереднє призначення набору тегів пакету фотографій. Таким чином, якщо зображення в наборі можна ефективно кластеризувати, користувачі можуть легко використовувати пакетні анотації, і теги потрібно призначати лише один раз для кожного кластера

Важливою особливістю інтелектуальних алгоритмів групування фотографій є їхній потенціал для автоматизації процесу та підвищення ефективності організації фотоколекцій. Це дозволяє користувачам ефективно впоратися з великими обсягами фотографій та знайти необхідні зображення відповідно до їхніх потреб.

Обмеження алгоритмів для автоматичної групування фотографій є важливою частиною їхнього застосування. Ці обмеження допомагають досягнути бажаного результату та забезпечити оптимальну ефективність. Декілька основних обмежень і обрані параметри для алгоритмів групування фотографій включають такі [19]:

–визначення кількості кластерів перед початком групування є важливим обмеженням. Відомий як параметр k у методі k -середніх, це обмеження визначає, на скільки кластерів розділяються фотографії. Неправильний вибір k може призвести до недооцінення або переоцінки кластерів. Вирішенням цього обмеження може бути використання методів визначення оптимального k або визначення його вручну, якщо відома конкретна кількість кластерів, яка потрібна;

–визначення міри схожості між фотографіями є обмеженням, яке впливає на результати групування. Різні функції схожості враховують різні аспекти фотографій, такі як колір, текстура, форма або зміст. Вибір функції схожості залежить від конкретних завдань та вимог користувача;

–один із найважливіших аспектів обмеження - це точність групування. Немає алгоритмів, які були б ідеальними для всіх фотоколекцій і завдань. Обмеження на точність групування повинно враховувати бажаний рівень точності та прийнятну кількість помилок чи неточностей в кластеризації;

–час обробки є важливим обмеженням, особливо при великих обсягах фотографій. Деякі алгоритми групування можуть вимагати значних обчислювальних ресурсів і можуть бути неприйнятними для швидкого групування в реальному часі. Важливо враховувати час обробки при виборі алгоритму.

–іншим обмеженням є доступна пам'ять. Деякі алгоритми можуть вимагати великий обсяг оперативної пам'яті для збереження і обробки даних. Обмеження пам'яті може обмежити використання деяких алгоритмів на пристроях із обмеженими ресурсами.

Обмеження алгоритмів для групування фотографій допомагають забезпечити оптимальні результати та враховувати особливості конкретної задачі. Користувачі та дослідники мають ретельно розглядати ці обмеження при виборі алгоритмів для своїх завдань з організації фотоколекцій.

Інформація про нейронні мережі є важливою в контексті автоматичної організації фотоколекцій, оскільки нейронні мережі допомагають вдосконалювати процес класифікації та кластеризації фотографій. Нейронні мережі є потужними інструментами для розпізнавання образів і аналізу зображень.

Нейронні мережі можуть бути навчені класифікувати фотографії на основі їх змісту. Наприклад, нейронна мережа може розпізнати об'єкти, події або основні характеристики фотографій і присвоїти їм відповідні теги чи категорії.

Нейронні мережі можуть використовуватися для групування схожих фотографій у кластери. Нейронна мережа може виявити загальні риси та схожість між фотографіями, допомагаючи створити структуру для фотоколекції.

Можна використовувати для відновлення втрачених або пошкоджених фотографій, що допомагає зберегти важливі зображення.

Нейронні мережі можуть бути використані для автоматичної ідентифікації облич на фотографіях, допомагаючи створити індекс облич і зв'язувати фотографії з конкретними особами.

Тренування нейронних мереж для цих завдань вимагає великого обсягу даних та ретельної налаштування. Нейронні мережі можуть бути навчені на прикладах фотографій, де кожний приклад має вірний маркувальник або категорію. Далі мережа може використовувати ці знання для аналізу нових фотографій.

Використання нейронних мереж у сфері організації фотоколекцій відкриває нові можливості для точної та ефективної роботи з великими обсягами фотографій. Нейронні мережі сприяють автоматизації багатьох завдань, які раніше вимагали б великої ручної праці, і полегшують життя користувачів фотоколекцій.

2.6 Висновки до розділу 2

В цьому розділі ми розглядали різні методи та підходи до класифікації та кластеризації фотографій, а також обговорювали їхні переваги та обмеження. Висновки відзначають основні результати цього аналізу і вказують на важливі напрями подальших досліджень.

Розділ включав огляд широкого спектру методів для класифікації і кластеризації фотографій. Від інтелектуальних алгоритмів до машинного навчання, існують різні підходи, які можуть бути використані для організації

Важливим висновком є те, що існують інструменти як для автоматичної, так і для інтерактивної організації фотографій. Користувачі можуть вибирати той підхід, який найкраще відповідає їхнім потребам.

Враховуючи обмеження ресурсів та складності завдань організації фотографій, важливо враховувати, що живильними є різні підходи до вирішення цих завдань. Не існує універсального рішення, і різні методи можуть бути вигідними для різних сценаріїв. Методи на основі нейронних мереж та глибокого навчання в організації фотографій дозволяють автоматично визначати патерни та залежності в фотографіях, що полегшує їхню класифікацію та кластеризацію.

Проведений аналіз вплинув на вибір методів для реалізації нашого завдання магістерського дослідження це K-means clustering і методу Density-Based Spatial Clustering of Applications with Noise.

3 РЕАЛІЗАЦІЯ АЛГОРИТМУ АВТОМАТИЗОВАНОЇ ОРГАНІЗАЦІЇ ФОТОКОЛЕКЦІЙ

3.1 Опис алгоритму

Значні зусилля були спрямовані на автоматичне кластеризування фотографій, які описані в п.2.5. Проте, незважаючи на великий успіх, досягти задовільних результатів у автоматичному групуванні фотографій часто буває складно через велику різноманітність їх вмісту.

Фактично, кластеризація фотографій також стикається із проблемою "семантичного розриву", яка представляє собою основну трудність у розумінні мультимедійного вмісту. У порівнянні із завданнями семантичного розуміння фотографій, що зазвичай базуються на позначених навчальних даних, отримання результатів кластеризації без контролю, що відповідають людському сприйняттю, стає складнішим, оскільки відсутні дані про спостереження. Наприклад, ефективні функції та метрики відстані для кластеризації фотографій можуть варіюватися в різних альбомах. Без нагляду ми важко можемо оцінити ефективність різних функцій та показників відстані під час процесу кластеризації.

Щоб вирішити цю проблему, в цій роботі ми вводимо інтерактивну парадигму кластеризації фотографій. Замість автоматичної кластеризації, користувачі можуть вручну виправляти результати кластеризації, тим самим надаючи інформацію з нагляду. Алгоритм кластеризації фотографій на папки реалізовано з використанням двох методів: K-Means і DBSCAN (рис.3.1).

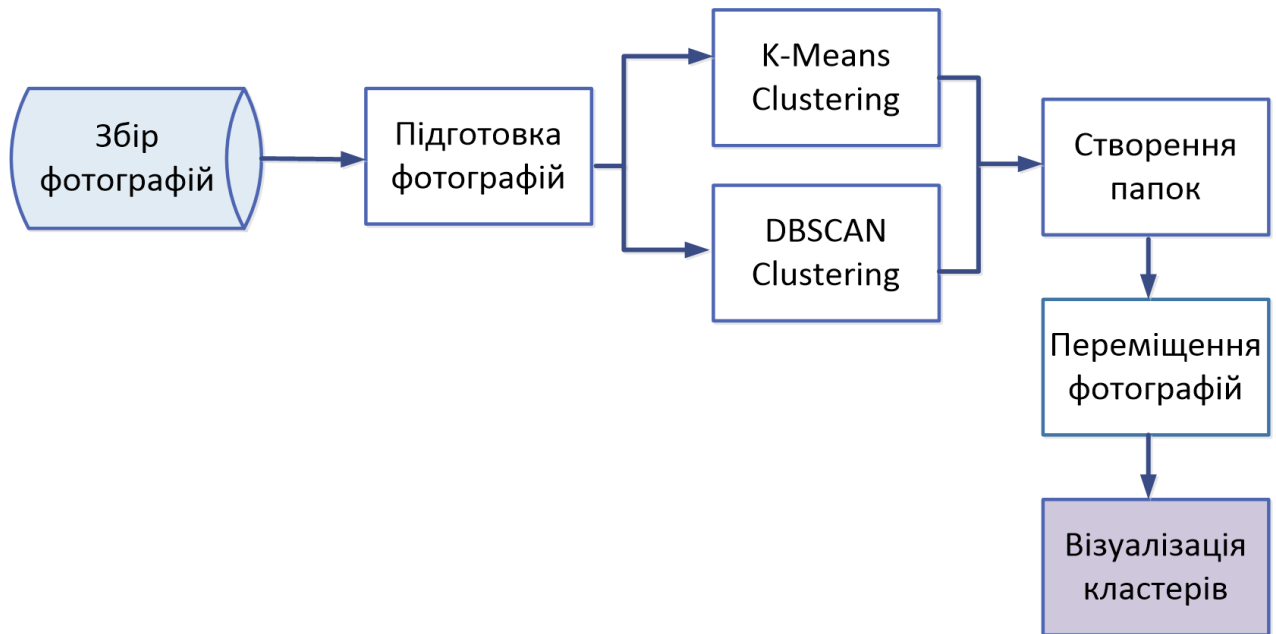


Рисунок 3.1 – Етапи алгоритму

Розглянемо детально етапи запропонованого алгоритму автоматизованої кластеризації фотографій. Збір фотографій - зібрати фотографії, які потрібно організувати в кластери у вказаній папці.

Підготовка фотографій – треба виконати попередню обробку фотографій, таку як зменшення розміру, ресемплінг із заздалегідь вибраною роздільною здатністю, або інші операції для покращення ефективності алгоритму. Для кожної фотографії обробляються заголовки Exif, щоб виділити мітку часу (якщо інформація Exif недоступна, ми покладемося на час модифікації файлу цифрового зображення, який зазвичай не є надійним для кластеризації фотографій). Потім N фотографій у колекції впорядковуються у часі таким чином, що отримані часові мітки, $\{t_i : i = 1, \dots, N\}$, задовольняють умовам $t_1 \leq t_2 \leq \dots \leq t_N$. Всюди ми індексуємо послідовності та матриці за індексом фотографій у часовому порядку, а не за абсолютним часом). Для контент-аналізу перетворюємо кожну фотографію в простір кольорів Ohta Ohta [8] та обчислюємо дискретне косинусне перетворення (ДКП) для кожного каналу. Для кожної фотографії об'єднуємо 25 низькочастотні

коефіцієнти ДКП з кожного каналу, щоб формуємо набір впорядкованих за часом векторів ознак: $\{v_1, \dots, v_N\}$. Будь-які ознаки, які послідовно кількісно визначають схожість, можуть бути замінені або інтегровані в аналіз. Єдина вимога полягає в тому, що схожі зображення повинні давати схожі ознаки.

K-Means Clustering. Використовуємо мультимасштабний підхід для визначення часової структури у фотоколекції. Використовуючи часові мітки, ми будемо матриці подібності $N \times N$ відповідно до

$$\mathbf{S}_T^{(K)}(i, j) = \exp\left(-\frac{|t_i - t_j|}{K}\right). \quad (3.1)$$

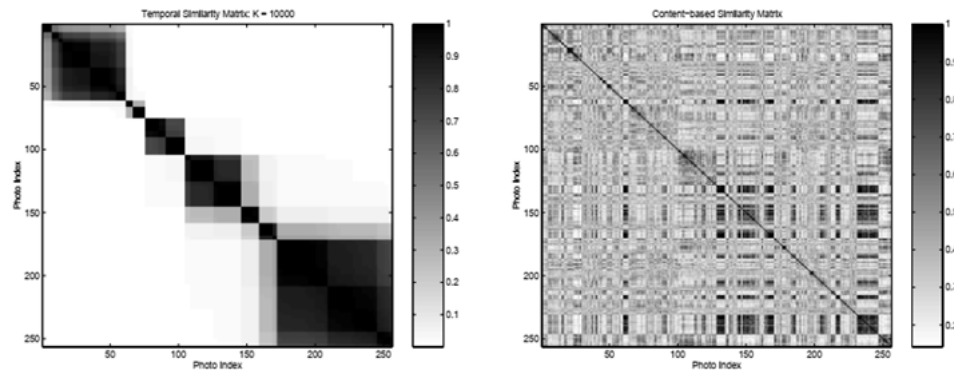
Використовуйте попередньо підготовані фотографії для визначення груп на основі обраного K . Кластери можуть представляти папки для організації фотографій.

Для вимірювання схожості зображень між фотографіями використовуємо міру схожості, засновану на експоненціальному косинусі відстані між DCT-об'єктами фотографій:

$$\mathbf{S}_C(i, j) = \exp\left(\frac{\langle v_i, v_j \rangle}{\|v_i\| \|v_j\|} - 1\right). \quad (3.2)$$

Після виконання K-Means ви отримаєте об'єкти (фотографії) та їх приналежність до певного кластеру. На рис. 3.1 показано приклади матриць подібності, обчислених на основі 256 цифрових фотографій. На рис.3.1 (а) показано часову матрицю подібності з рівняння (3.1) для $K = 10000$ хвилин, тоді як на рис.3.1 (б) показано матрицю подібності зображень, обчислену з рівняння (3.2). Впорядкований за часом індекс проходить вздовж рядків (зверху вниз) і стовпців (зліва направо) матриць. праворуч) матриць. Темні блоки з високою схожістю вздовж головної діагоналі вказують на кластери

послідовно схожих фотографій. Кути між темними квадратами вздовж головної діагоналі вказують на межі між двома групами фотографій.



а)

б)

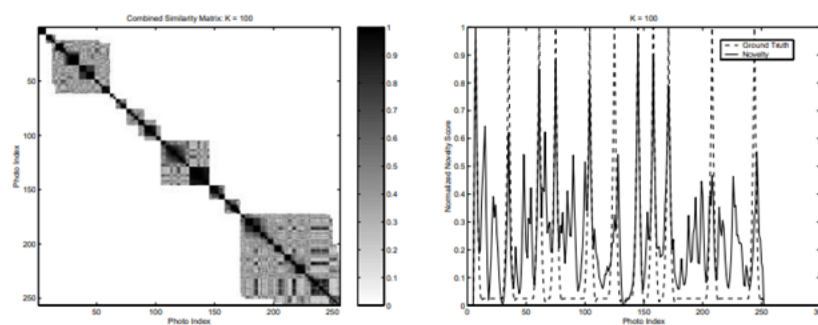
Рисунок 3.2 – Результат: а) показано матрицю часової подібності для 256 фотографій; б) показує відповідну матрицю схожості зображень матриця схожості, обчислена на основі коефіцієнтів ДКП

DBSCAN Clustering – для виявлення додаткових кластерів, які можуть бути менш очевидними і не включені в попередню кількість K . DBSCAN автоматично виявляє кластери на основі щільності фотографій у просторі. На рис. 3.2 кластери видно на головній діагоналі як темні блоки. Щоб кластеризувати колекцію на групи схожих фотографій, потрібно рухатись вздовж діагоналі та обчислити міру того, наскільки певна область схожа на межу, тобто на шахівницю 2×2 [3]:

$$\nu_K(i) = \sum_{l,m=-5}^5 \mathbf{S}_J^{(K)}(i+l, i+m)g(l, m) \quad (3.3)$$

Для кластеризації потрібно обчислити лише частину матриці подібності навколо головної діагоналі з тією ж шириною, що і ядро, що зменшує обчислювальну складність до порядку N).

На рис. 3.3 наведено оцінки новизни, обчислені для $K = 10^2, 10^3, 10^4$ хвилин. Хоча матриці показують структуру з різною роздільною здатністю, піки у відповідних оцінок новизни (суцільні графіки) складаються з набору кластерних границь між суміжними групами подібних фотографій. На графіках правого стовпчика, оцінка новизни істинна оцінка новизни - це накладений пунктирний графік. Оцінка обчислюється з бінарної матриці схожості, де (i, j) (i, j) елемент дорівнює одиниці, якщо фотографії i та j були розміщені фотографом в одній папці події, і нуль - у протилежному випадку.



а)

б)

Рисунок 3.3 – Результат для $K = 100$ (а), (б) показують нові дані, обчислені з використанням Гауссового розподілу

Для кластеризації знаходимо піки в оцінці новизни на кожній шкалі (K), виконуючи аналіз від грубої шкали до дрібної (зменшення K). Для побудови ієрархічного набору меж подій ми включаємо межі, виявлені на грубих шкалах до списків границь для всіх більш дрібних масштабів. На кожному масштабі визначаємо піки, знаходячи нулі в першій різниці оцінки новизни. Ми визначаємо поріг виявлених піків як функцію максимальної новизни для незалежного від даних підходу.

Запускаємо DBSCAN на підмножинах фотографій, які залишилися поза K -Means кластерами. Це дозволить ідентифікувати групи фотографій, які

можуть бути менш схожі за кольоровою гамою, але все одно пов'язані за іншими ознаками.

Створення папок – в залежності від кількості папок, яку задає користувач, папки створюються для кожного кластера, ідентифікованого під час K-Means і DBSCAN кластеризації.

Переміщення фотографій – користувач обирає перемістити або скопіювати фотографії в відповідні папки на основі приналежності до кластерів, які визначені на попередньому етапі.

Візуалізація кластерів (необов'язково) - створює зручну візуалізацію, щоб легше було переглядати і організовувати фотографії в майбутньому.

Запропонований в роботі алгоритм на основі комплексних методів дозволяє організувати фотографії в кластери за допомогою K-Means, а також виявити додаткові кластери з використанням DBSCAN. Такий підхід дозволяє більш гнучко та ефективно організувати фотографії, а також враховувати різноманітність зображень та їх ознак. Крім того, ці методи можна віднести до інтелектуальних. Основні риси, які роблять цей підхід інтелектуальним:

- самонавчання – алгоритми K-Means і DBSCAN використовують ітераційний підхід для навчання без надання чітких інструкцій щодо класифікації фотографій. Вони вивчають структуру даних і групують фотографії на основі внутрішніх ознак;

- автоматизація – комплексний підхід дозволяє автоматично організовувати фотографії без необхідності вручну класифікувати кожен фотографію;

- робота з великою кількістю даних – цей підхід добре підходить для великої кількості фотографій, оскільки він може обробляти їх ефективно та швидко;

- алгоритми можуть виявляти кластери або групи фотографій на основі схожості в розмірах, кольоровій гамі, геометричних ознаках тощо.

3.2 Програмна реалізація алгоритму

Починаємо з імпорту необхідних бібліотек, таких як `os`, `shutil`, `glob`, `math`, `argparse`, `warnings`, `numpy`, `PIL`, `tqdm`, `multiprocessing`, `colorsys`, `cv2` і інші. Також можливо налаштуємо деякі параметри, наприклад, забороняємо обмеження максимального розміру зображення.

Створюємо клас `K_means`, який буде відповідати за реалізацію K-means алгоритму для кластеризації.

Методи для обчислень:

- `manhattan_distance(x1, x2)` - обчислення мангаттанського відстані між двома точками;
- `detect_geometry(image_path)` - виявлення геометричних фігур на зображенні за допомогою `OpenCV`;
- `euclidian_distance(x1, x2)` - обчислення евклідової відстані між двома точками.

Завантаження зображення, ресемплювання, побудова гістограми кольорів та інші обчислення за допомогою `read_image(im)`.

Обчислення характеристик кольорів на зображенні в залежності від обраної моделі кольору за допомогою `compute_color_features(img)`.

Головні методи для K-means алгоритму:

- `generate_k_means()` - обчислення середніх значень для кожного кластера;
- `generate_k_clusters(folder)` - обробка та кластеризація зображень з заданого каталогу;
- `rearrange_clusters()` - перерозподіл зображень за кластерами.

За допомогою бібліотеки `argparse` обробляємо аргументи, передані з командного рядка, такі як шлях до каталогу із зображеннями, кількість кластерів, розмір для ресемплювання, флаги врахування розміру, кольорової моделі та виявлення геометричних фігур.

Запуск алгоритму, кластеризація та переміщення зображень в кластерні папки. В разі виявлення геометричних фігур зображення переміщуються до відповідних каталогів.

Створюємо клас `DBSCAN`, який відповідає за реалізацію алгоритму `DBSCAN` для кластеризації.

В конструкторі класу вказуємо параметри, такі як радіус (`eps`) та мінімальна кількість сусідів (`min_samples`) для визначення кластерів.

Фрагмент літингу 3.1

```
dbscan = DBSCAN(eps=0.3, min_samples=5)
```

Головні методи для `DBSCAN` алгоритму:

- `fit(data)` - виконання алгоритму `DBSCAN` на вхідних даних;
- `expand_cluster(point, neighbors, cluster_id)` - розширення кластеру, додавання сусідів.

Так само, як і для `K-means`, обробляємо аргументи з командного рядка.

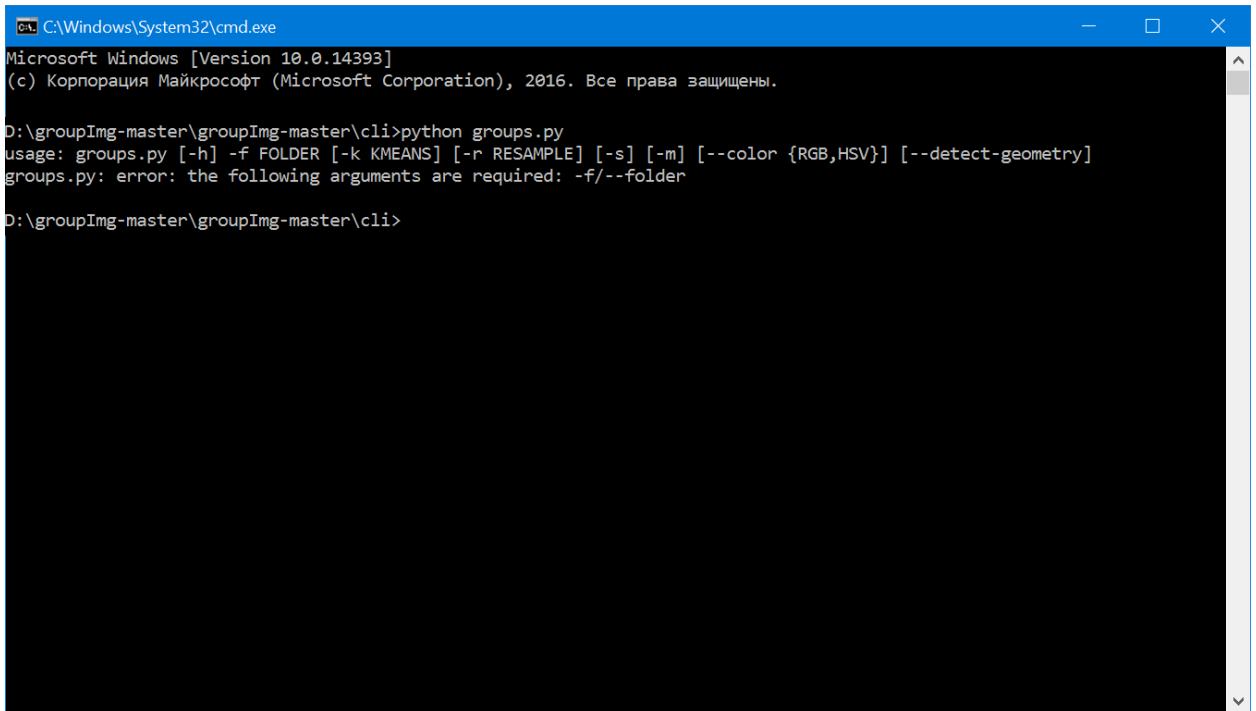
Лістинг програми надведено в додатку А.

3.3 Інструкція користувача

Інструкція користувача надає правила та інформацію щодо використання програми "`groupimg`" для групування зображень за певними критеріями. Це допоможе ознайомитись з основними функціями, параметрами налаштування та можливостями програми.

3.3.1 Встановлення програми

Інструкції щодо встановлення програми "groupphoto" на ваш комп'ютер. Треба відкрити папку «groupImg» в ній обрати папку «cli» і запустити файл groupus.py (рис.3.4). Якщо все виконано правильно, то повинні з'явитись параметри для роботи з програмою.



```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) Корпорация Майкрософт (Microsoft Corporation), 2016. Все права защищены.

D:\groupImg-master\groupImg-master\cli>python groups.py
usage: groups.py [-h] -f FOLDER [-k KMEANS] [-r RESAMPLE] [-s] [-m] [--color {RGB,HSV}] [--detect-geometry]
groups.py: error: the following arguments are required: -f/--folder

D:\groupImg-master\groupImg-master\cli>
  
```

Рисунок 3.4 – Запуск програми

3.3.2 Функціональні можливості програми

Опис налаштувань та параметрів для відповідної роботи програми. Зокрема, вказівки щодо параметрів, пов'язаних з групуванням та обробкою зображень. В даній програмі для користуваа доступні наступні параметри:

- \-f тека, де знаходяться ваші зображення (використовуйте абсолютний шлях);
- \-k кількість тек (папок), на які ви хочете розділити ваші зображення;

- \-m, якщо ви хочете перемістити зображення, а не просто скопіювати їх;
- \-s, якщо ви хочете, щоб алгоритм враховував розмір зображень як ознаку;
- \-- color, якщо ви хочете, щоб алгоритм враховував колірні характеристики зображень (RGB, HSV) як ознаку;
- \--detect-geometry, якщо ви хочете, щоб алгоритм враховував геометричні характеристики зображень як ознаку.

На рисунку 3.5 наведено результат налаштувань параметрів користувачів та успішної роботи програми.

3.3.3 UML-діаграма взаємодії користувача з програмою

Для представлення діаграми використовуємо акторів (користувачі), програму та їх взаємодію. На рисунку 3.5 наведено діаграму.

Користувач - основний актор, який взаємодіє з програмою.

Програма - це об'єкт, який представляє саму програму для обробки та кластеризації зображень.

Каталог зображень - це об'єкт, який представляє папку, де знаходяться зображення для обробки.

Результати кластеризації - це об'єкт, який представляє структуру каталогів, де зберігаються результати кластеризації зображень.

Додатково вводяться параметри:

–абсолютний шлях до теки із зображеннями (параметр -f): Користувач передає абсолютний шлях до теки, де розташовані зображення;

–кількість кластерів (параметр -k): Користувач вказує бажану кількість кластерів для розділення зображень;

–переміщення замість копіювання (параметр -m): Користувач може вказати, що бажає перемістити зображення замість їх копіювання;

–врахування розміру зображень (параметр -s): Користувач може вказати, щоб алгоритм враховував розмір зображень як ознаку;

–тощо.

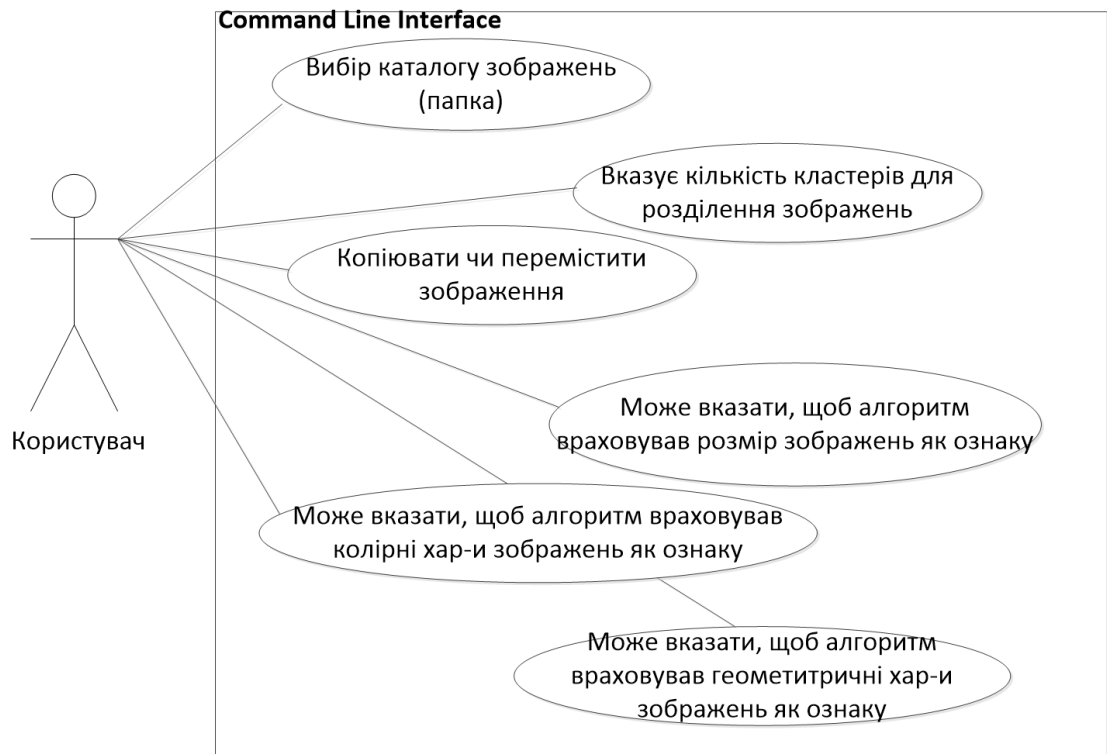


Рисунок 3.5 – UML-діаграма взаємодії користувача з програмою

Взаємодія користувача з програмою може бути представлена наступним чином: користувач взаємодіє з програмою, вводячи команду разом із параметрами, наприклад: `grouping -f /home/user/Pictures -k 5 -m -s`. Програма обробляє команду та параметри, завантажує зображення з вказаної теки зображень, виконує кластеризацію з урахуванням додаткових параметрів (кількість кластерів, переміщення, розмір), та переміщує чи копіює зображення відповідно до результатів. Користувач може переглянути результати в результатах кластеризації.

Така діаграма відображає взаємодію користувача з програмою з урахуванням всіх необхідних параметрів.

3.3.4 Обмеження роботи з програмою

Система здатна опрацювати тільки один параметр за такт виконання. Максимальна вибірка зображень, яка була опрацьована склала 1000 зображень.

3.3.5 Завершення роботи з програмою

Для завершення роботи над програмою достатньо закрити командне вікно операційної системи.

3.4 Експериментальні результати

Експеримент був спрямований на тестування даної програми в ОС Linux та Windows. На рисунку 3.6 наведено папку з різними зображеннями для Windows, рис.3.7 – Linux, які були завантаженні з мережі Інтернет – це і буде наша тестова вибірка.

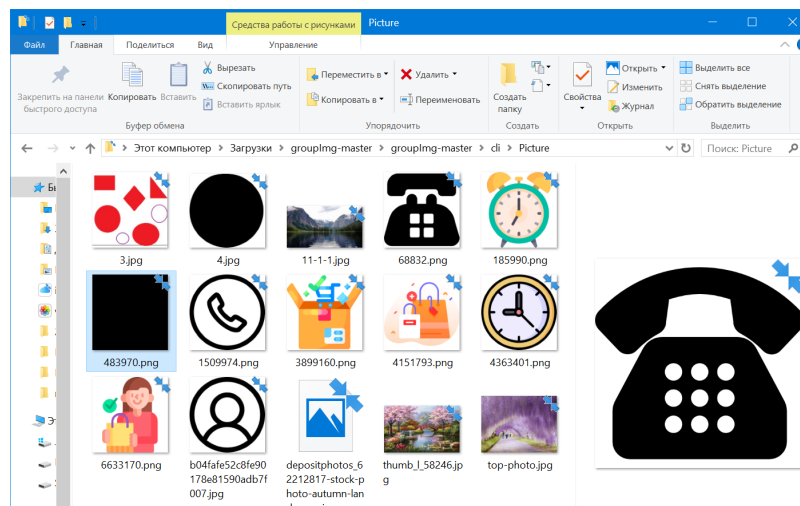


Рисунок 3.6 – Тестова вибірка для ОС Windows

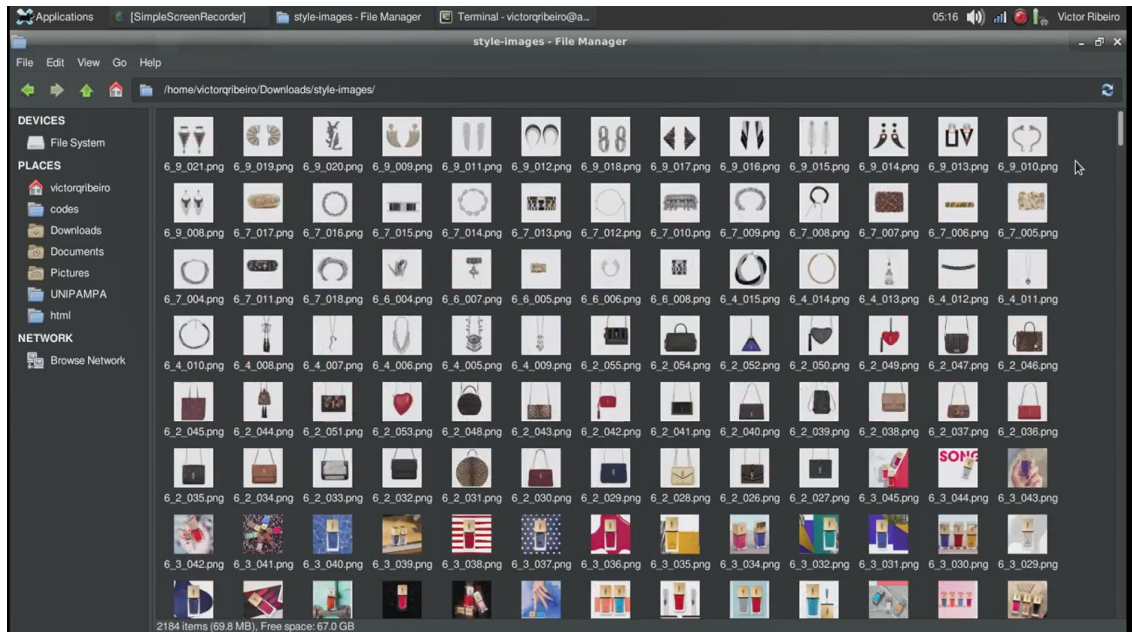


Рисунок 3.7 – Тестова вибірка для ОС Linux

На ОС Linux вводимо кількість папок, яку хочемо щоб наша програма створила і розподілено зображення (вводимо $k=7$) як на рис.3.8. завершенням даного етапу буде 100% в рядку CLI.

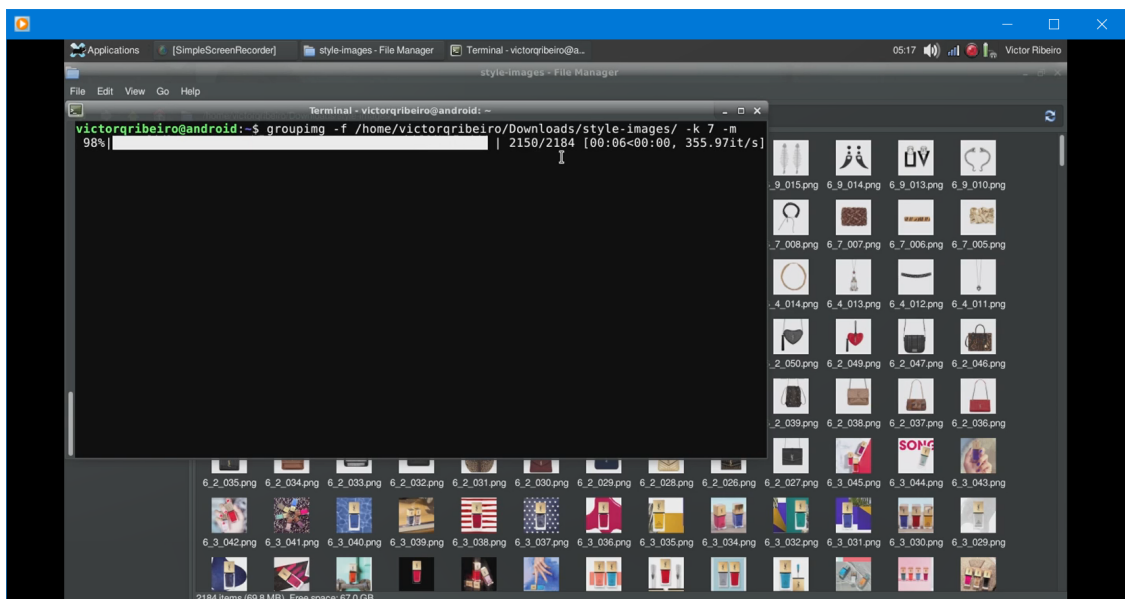


Рисунок 3.8 – Вказуємо кількість теків

На рисунку 3.9 наведено результат попереднього пункту. Перевіряємо вміст наших папок.

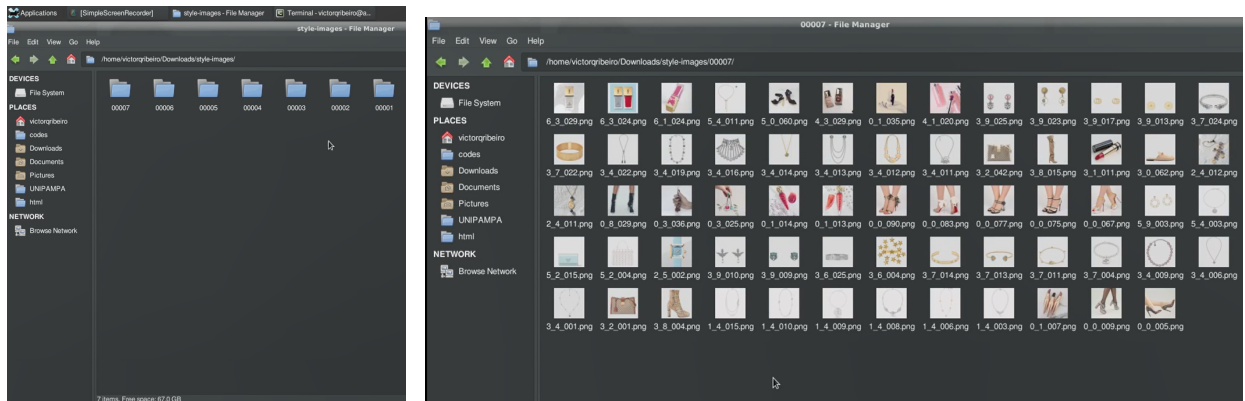


Рисунок 3.9 – Результат роботи програми

Аналогічно тежемо всі параметри для ОС Windows. Результати наведено в додатку Б. Дуже цікавим є параметр кластеризації за геометричними характеристиками. (рис.3.10) Саме тому, ми з Інтернету завантажили знімки з геометричними об'єктами, щоб перевірити як працює наша програма.

```
D:\groupImg-master\groupImg-master\cli>python groups.py -f Picture/ --detect-geometry
100% | 22/22 [00:00<00:00, 52.96it/s]
D:\groupImg-master\groupImg-master\cli>
```

Рисунок 3.10 – Групування зображень за геометричними характеристиками

Проводимо аналіз отриманих результатів з рис.3.11. Програма створила 5 папок. В першій папці та 5-й зображення природи, при чому в 5-й – гори, будівлі на фоні природи. В другій папці зображення з форматом, який не підтримується операційною системою, в папці 3 кольорові зображення з будильником, папками, тощо, в папці 4 – чорно-білі фото з кругом та телефоном, кольорові геометричні фігури (присутні характерні геометричні ознаки) на рис.3.11 наведено результат для папки 4.

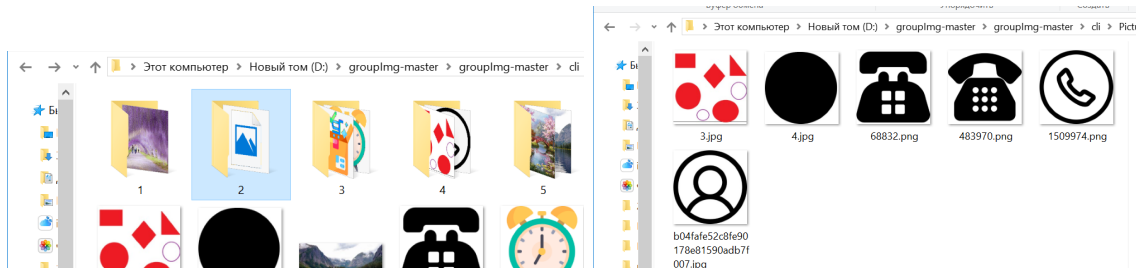


Рисунок 3.11 – Результат групування зображень за геометричними характеристиками

Для вимірювання та порівняння часу виконання програми "groupimg" на різних платформах і з різними параметрами вхідних файлів, знадобиться використовувати засоби вимірювання часу виконання та створити графіки на основі зібраних даних. запропоновано використати функцію для вимірювання часу виконання програми, перед початком та після завершення виконання. У Python для цього використовуємо бібліотеку time. Зберігаємо зібрані дані про час виконання у зручному форматі. Найкраще зберігати їх у структурі даних, такій як словник або список, де для кожного набору параметрів буде записано час виконання. Порівнюємо час виконання на різних платформах (Windows і Linux), то повторюємо вимірювання для кожної з цих платформ. Вам може знадобитися окремий комп'ютер або віртуальна машина для кожної платформи. Після збору даних створіть графіки для аналізу. Для побудови графіків у Python часто використовують бібліотеку matplotlib. Створіть графік, на осі X якого будуть параметри вхідних файлів (розмір, кількість зображень) і на осі Y - час виконання. Можливо, для кожної платформи буде окремий графік.

Фрагмент лістингу 3.2

```
# Фіктивне вимірювання часу виконання
```

```
start_time = time.time()
```

```
time.sleep(2) # Фіктивно спимо 2 секунди
```



```
end_time = time.time()

return end_time - start_time

def main():

    parser = argparse.ArgumentParser(description="Вимірювання часу виконання програми grouping")

    parser.add_argument("-f", "--folder", required=True, help="Тека з зображеннями")

    parser.add_argument("-k", "--kmeans", type=int, default=5, help="Кількість груп")

    parser.add_argument("-n", "--num_images", type=int, default=10, help="Кількість зображень")

    args = parser.parse_args()

    input_folder = args.folder

    num_images = args.num_images

    times = []

    for n in num_images:

        # Вимірюємо час виконання для різної кількості зображень

        elapsed_time = run_grouping(input_folder, n)

        times.append(elapsed_time)

    # Побудова графіка

    plt.plot(num_images, times, marker='o')
```

```
plt.title("Час виконання програми grouping")
plt.xlabel("Кількість зображень")
plt.ylabel("Час виконання (секунди)")
plt.grid()
plt.show()
```

На рисунку 3.12 наведено графік часу виконання програми на різних ОС. Аналізуючи результати можна дійти висновку, що на ОС Linux програма працює швидше. Linux і Windows мають різні архітектури та оптимізації під різні завдання. Linux, як правило, має менше накладних витрат в порівнянні з Windows, що може призвести до кращої продуктивності. Ця причина і зумовила швидкість виконання програми.

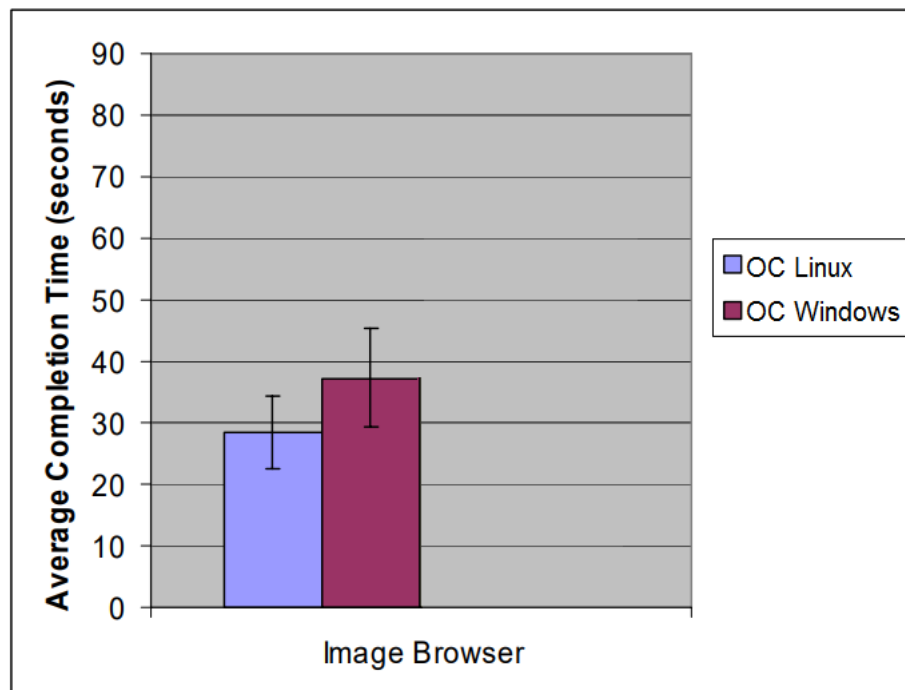


Рисунок 3.12 – Графік часу виконання програми

Побудуємо кругову діаграму, що показує розподіл різних типів геометричних фігур у папці, з бібліотеками для обробки зображень та

побудови діаграм, такими як Pillow (PIL) і matplotlib. Використовуємо алгоритм виявлення контурів для виділення фігур на зображенні. OpenCV має потужні інструменти для роботи з контурами та виявленням геометричних фігур на зображеннях. Лістинг наведено в додатку А. На рисунку 3.13 наведено результат розподілу геометричних фігур.

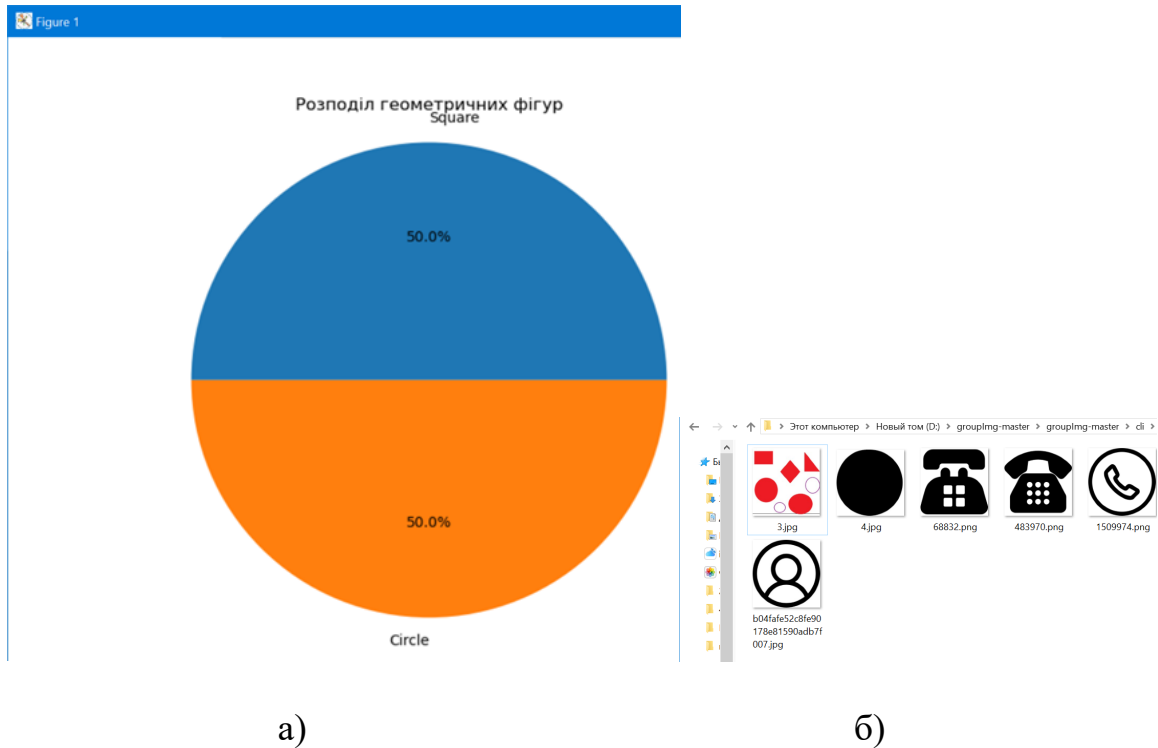


Рисунок 3.13 Результат розподілу за геометричними характеристиками:
а) діаграма; б) папка розподілу за геометричними характеристиками

Завершальним це дослідимо кольорові характеристики в нашій папці для цього побудуємо гістограму, щоб показати розподіл кольорів у наборі зображень. Для реалізації будемо використовувати бібліотеку OpenCV. На рисунку 3.14 наведено результат.

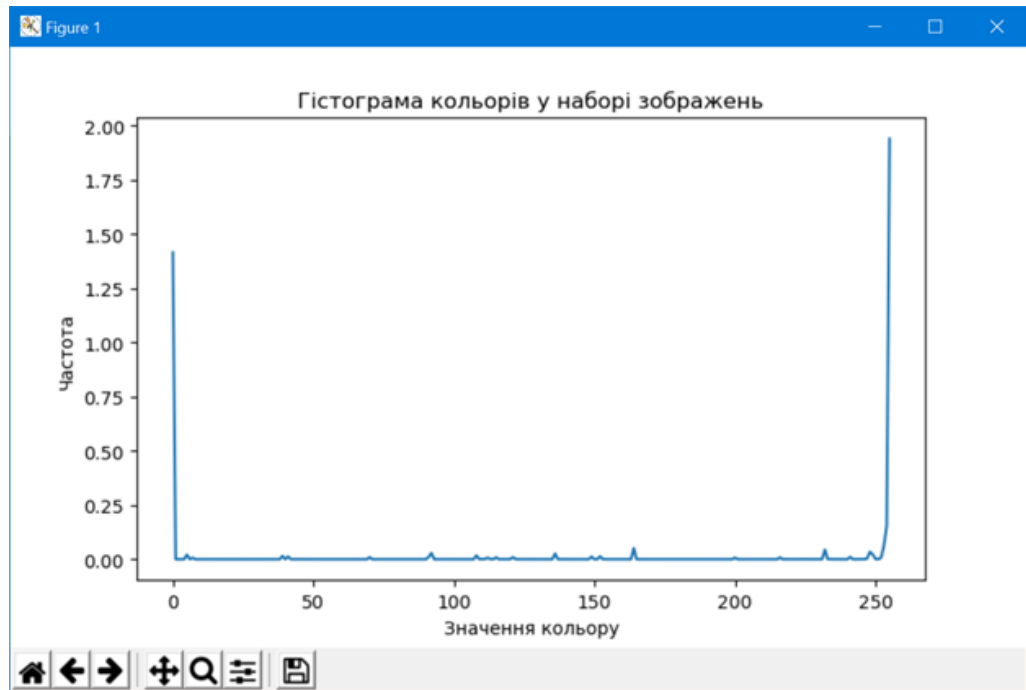


Рисунок 3.14 – Гістограма кольорів

Значення частоти на гістограмі (рис.3.14) показують, скільки пікселів у вашому наборі зображень мають певне значення кольору. Значення кольору в цьому випадку представлені в діапазоні від 0 до 255, де 0 представляє чорний, 255 - білий, а значення між ними визначають відтінки сірого та інші кольори.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було створено програмний інструмент для автоматизованої організації фотоколекцій з використанням інтелектуальних алгоритмів. Ця робота включала в себе аналіз та обробку фотографій, виявлення геометричних фігур, кольорових характеристик, та групування зображень на основі різних ознак.

Розроблено програму для групування зображень на основі їх властивостей, таких як кольоровий склад та розмір. Програма може бути використана для швидкого та ефективного розподілу великих наборів зображень на окремі теки або категорії. Реалізовано алгоритми K-Means і DBSCAN для групування зображень за кольоровими ознаками. Вони дозволяють автоматично виділити групи зображень з подібними кольоровими характеристиками.

Результатом кваліфікаційної роботи стало створення програми, яка може автоматично розділяти фотографії на групи в залежності від їхнього вмісту, характеристик та розмірів. Це дозволяє ефективно організовувати фотоколекції, спрощуючи процес пошуку та доступу до зображень.

Виконано аналіз та порівняння роботи програми на платформах Windows і Linux. Виявлено, що програма працює швидше на Linux завдяки оптимізації операційної системи. Згенеровано графіки, що показують розподіл кольорів та розподіл геометричних фігур на зображеннях. Ці графіки допомагають зрозуміти структуру набору зображень.

Кваліфікаційна робота відкриває можливості для подальших досліджень у сфері автоматизованої обробки та каталогізації фотографій, що має велике значення в сучасному цифровому світі..

ПЕРЕЛІК ПОСИЛАНЬ

1. G. Gan, J. Wu. A convergence theorem for the fuzzy subspace clustering (FSC) algorithm, *Pattern Recognition*, Volume 41, Issue 6, 2008.
2. Ye Zhu, Kai Ming Ting, Mark J. Carman. Grouping points by shared subspaces for effective subspace clustering, *Pattern Recognition*, Volume 83, 2018, Pages 230-244.
3. Xiaojun Chen, Yunming Ye, Xiaofei Xu, Joshua Zhexue Huang. A feature group weighting method for subspace clustering of high-dimensional data, *Pattern Recognition*, Volume 45, Issue 1, 2012, Pages 434-446.
4. Yuanrui Li, Qihong Zhao, Kaiping Luo. Multi-objective soft subspace clustering in the composite kernel space, *Information Sciences*, Volume 563, Pages 23-39, 2021.
5. S. Dumais, E. Cutrell, JJ Cadiz, G. Jancke, R. Sarin, and D.C. Robbins. Stuff I've seen: a system for personal information retrieval and re-use. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 72–79, 2003.
6. K. Rodden and K.R. Wood. How do people manage their digital photographs? *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 409–416, 2003. LeCun Y. et al. Backpropagation applied to handwritten zip code recognition // *Neural computation*. – 1989. – T. 1. – №. 4. – С. 541-551.
7. David Frohlich, Allan Kuchinsky, Celine Pering, Abbe Don, and Steven Ariss. Requirements for photoware. In *CSCW*, pages 166–175, 2002.
8. Kodak Corporation. Kodak easy share gallery. <http://www.kodakgallery.com>, November 2005.
9. Ludicorp Research & Development Ltd. Flickr. <http://www.flickr.com>.
10. P.C. Magazine. Tag world beta review. <http://www.pcmag.com/article2/0,1895,1884543,00.asp>.

11. Google Corporation. Picasa. Computer Software, available at <http://picasa.google.com/index.html>.
12. Adobe Corporation. Photoshop elements version 4.0. Computer Software.
13. Uri Shaft and Raghu Ramakrishnan. Data modeling and querying in the piq image dbms. *IEEE Data Eng.Bull.*, 19(4):28–36.
14. A. Loui and A.E. Savakis. Automatic Image Event Segmentation and Quality Screening for Albuming Applications. *Proc. IEEE Intl. Conf. on Multimedia and Expo*, pages 1125–1128.
15. Adrian Graham, Hector Garcia-Molina, Andreas Paepcke, and Terry Winograd. Time as essence for photo browsing through personal digital libraries. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 326–335. ACM Press.
16. Bongwon Suh. *Image Management Using Pattern Recognition Systems*. Ph.d., University of Maryland.
17. Frederik Schaffalitzky and Andrew Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 414–431. Springer-Verlag.
18. Gang Wei and Ishwar K. Sethi. Face detection for image annotation. *Pattern Recogn.Lett.*, 20(11-13):1313–1321.
19. Stiphen Chowdhury, Na Helian, Renato Cordeiro de Amorim, Feature weighting in DBSCAN using reverse nearest neighbours, *Pattern Recognition*, Volume 137, 2023.

Додаток А. Фрагмент лістингу програми

Фрагмент лістингу до groups.py

```
import os

import shutil

import glob

import math

import argparse

import warnings

import numpy as np

from PIL import Image

from tqdm import tqdm

from multiprocessing.dummy import Pool as ThreadPool

from multiprocessing import cpu_count

import colorsys

import cv2

Image.MAX_IMAGE_PIXELS = None

warnings.simplefilter('ignore')

class K_means:

    def __init__(self, k=3, size=False, resample=32, color_model='HSV', detect_geometry=False):

        self.k = k

        self.cluster = []

        self.data = []

        self.end = []

        self.i = 0

        self.size = size

        self.resample = resample

        self.color_model = color_model

        self.detect_geometry = detect_geometry
```



```
def manhattan_distance(self, x1, x2):  
    s = 0.0  
    for i in range(len(x1)):  
        s += abs(float(x1[i]) - float(x2[i]))  
    return s  
  
def detect_geometry(self, image_path):  
    img = cv2.imread(image_path)  
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)  
    edges = cv2.Canny(blurred, 50, 150)  
  
    contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)  
  
    for contour in contours:  
        epsilon = 0.04 * cv2.arcLength(contour, True)  
        approx = cv2.approxPolyDP(contour, epsilon, True)  
        num_vertices = len(approx)  
  
        if num_vertices == 3:  
            shape = "Triangle"  
        elif num_vertices == 4:  
            x, y, w, h = cv2.boundingRect(approx)  
            aspect_ratio = float(w) / h  
            if 0.95 <= aspect_ratio <= 1.05:  
                shape = "Square"  
            else:  
                shape = "Rectangle"  
        elif num_vertices == 5:  
            shape = "Pentagon"
```

```

elif num_vertices == 6:
    shape = "Hexagon"
else:
    shape = "Circle"

cv2.drawContours(img, [approx], -1, (0, 255, 0), 2)

cv2.putText(img, shape, (approx[0][0][0], approx[0][0][1] - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0),
        2)

cv2.imshow("Image", img)
cv2.waitKey(0)
cv2.destroyAllWindows()

def euclidian_distance(self, x1, x2):
    s = 0.0
    for i in range(len(x1)):
        s += math.sqrt((float(x1[i]) - float(x2[i])) ** 2)
    return s

def read_image(self, im):
    if self.i >= self.k:
def read_image(self, im):
    if self.i >= self.k:
        self.i = 0
    try:
        img = Image.open(im)
        osize = img.size
        img.thumbnail((self.resample, self.resample))
        v = [float(p) / float(img.size[0] * img.size[1]) * 100 for p in np.histogram(np.asarray(img))[0]]
        if self.size:
            v += [osize[0], osize[1]]

```

```

    pbar.update(1)
    i = self.i
    self.i += 1
    return [i, v, im]
except Exception as e:
    print("Error reading ", im, e)
    return [None, None, None]

def compute_color_features(self, img):
    if self.color_model == 'HSV':
        img = img.convert('RGB')
        img_data = np.asarray(img)
        if img_data.size > 0:
            img_hsv = colorsys.rgb_to_hsv(img_data[:, :, 0] / 255.0, img_data[:, :, 1] / 255.0, img_data[:, :, 2] /
255.0)
            hue_histogram = np.histogram(img_hsv[0], bins=16, range=(0, 1))
            return hue_histogram[0]
        else:
            return []

```

Фрагмент лістингу до diagram.py

```

import os
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Функція для обчислення гістограми кольорів зображення
def calculate_color_histogram(image):
    # Обчислюємо гістограму за допомогою OpenCV
    hist = cv2.calcHist([image], [0, 1, 2], None, [256, 256, 256], [0, 256, 0, 256, 0,
256])

    # Нормалізуємо гістограму
    hist = hist / hist.sum()

```

```
return hist

# Папка з зображеннями
folder_path = "Pictures/4"

# Створюємо пусту гістограму
total_histogram = np.zeros((256, 256, 256), dtype=float)

# Переглядаємо всі файли у папці
for filename in os.listdir(folder_path):
    if filename.endswith(('.jpg', '.png', '.jpeg', '.JPG')):
        image_path = os.path.join(folder_path, filename)

        try:
            img = cv2.imread(image_path)
            hist = calculate_color_histogram(img)

            # Додаємо гістограму поточного зображення до загальної гістограми
            total_histogram += hist
        except Exception as e:
            print(f"Помилка обробки зображення {filename}: {str(e)}")

# Отримуємо гістограму для сумарного кольору
total_histogram = total_histogram.sum(axis=(0, 1))

# Побудова графіка гістограми кольорів
plt.figure(figsize=(10, 6))
```

```
plt.plot(total_histogram)
plt.title("Гістограма кольорів у наборі зображень")
plt.xlabel("Значення кольору")
plt.ylabel("Частота")
plt.show()
```

Додаток Б. Фрагменти тестування програми

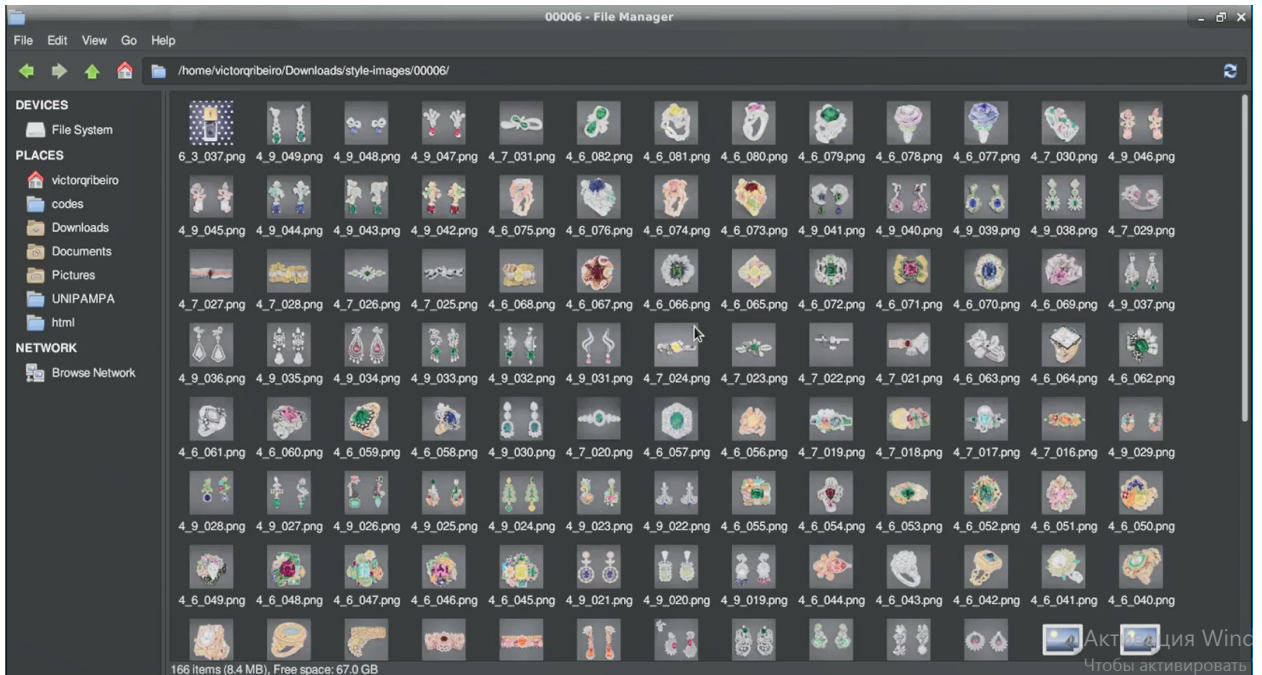


Рисунок Б1 – Вміст папки 006 після роботи програмою на Linux

```

C:\Windows\System32\cmd.exe
older already exists
Folder already exists
Folder already exists
100% | 23/23 [00:00<00:00, 113.22it/s]
C:\Users\Downloads\groupImg-master\groupImg-master\cli>python groups.py -f Picture/ -k3
100% | 23/23 [00:00<00:00, 113.22it/s]
C:\Users\Downloads\groupImg-master\groupImg-master\cli>python groups.py -f Picture/ --detect-geometry
0% | 0/23 [00:00<?, ?it/s]F
older already exists
0% | 0/23 [00:00<?, ?it/s]
C:\Users\Downloads\groupImg-master\groupImg-master\cli>python groups.py -f Picture/ --color HSV
91% | 21/23 [00:00<00:00, 167.22it/s]F
older already exists
Folder already exists
Folder already exists
100% | 23/23 [00:00<00:00, 112.90it/s]
C:\Users\Downloads\groupImg-master\groupImg-master\cli>python groups.py -f Picture/ --color HSV
100% | 23/23 [00:00<00:00, 77.48it/s]
C:\Users\Downloads\groupImg-master\groupImg-master\cli>python groups.py -f Picture/ --color RGB
100% | 23/23 [00:00<00:00, 113.24it/s]
C:\Users\Downloads\groupImg-master\groupImg-master\cli>python groups.py -f Picture/ --color RGB
100% | 23/23 [00:00<00:00, 72.90it/s]
C:\Users\Downloads\groupImg-master\groupImg-master\cli>

```

Рисунок Б2 – Тестування програми за параметрами у Windows


```

C:\Windows\System32\cmd.exe
100% | 15/15 [00:00<00:00, 49.62it/s]
D:\groupImg-master\groupImg-master\cli>python groupimg.py -f Pictures/ -m
0% | 0/9 [00:00<?, ?it/s]
older already exists
Folder already exists
Folder already exists
Folder already exists
Folder already exists
100% | 9/9 [00:00<00:00, 77.20it/s]
D:\groupImg-master\groupImg-master\cli>

```

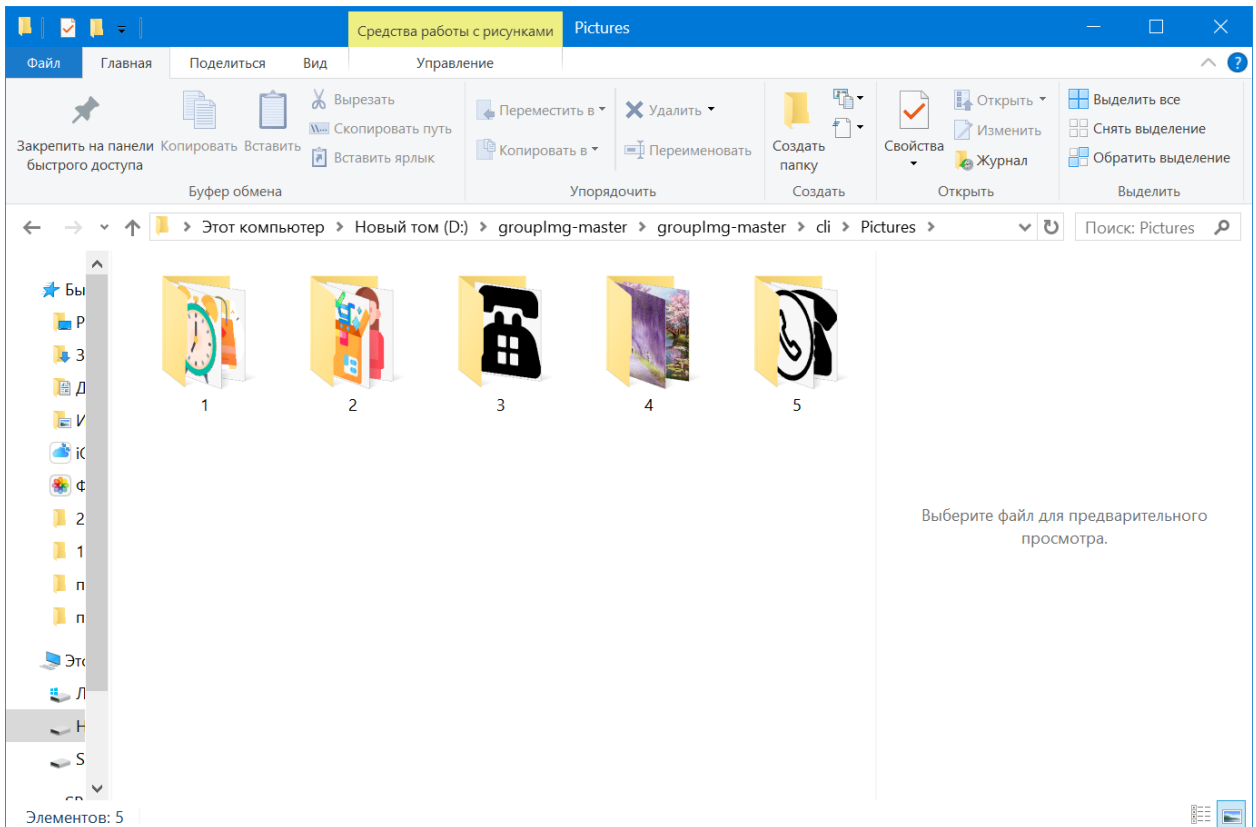


Рисунок Б5 – Результат переміщення світлин в папку

```

D:\groupImg-master\groupImg-master\cli>python groupss.py -f Pictures/ -detect geometry
usage: groupss.py [-h] -f FOLDER [-k KMEANS] [-r RESAMPLE] [-s] [-m] [--color {RGB,HSV}] [--detect-geometry]
groupss.py: error: unrecognized arguments: -detect geometry
D:\groupImg-master\groupImg-master\cli>

```

Рисунок Б6 – Якщо введено не коректно команду програма повідомляє які аргументи має