

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

МОДЕЛЮВАННЯ МЕТОДУ КЛАСИФІКАЦІЇ З ВИКОРИСТАННЯМ
ПАРАМЕТРІВ КЛАСТЕРНОГО ПОДАВАННЯ ДЛЯ ОПИСУ ЗОБРАЖЕНЬ
(тема)

Виконав:
студент 4 курсу, групи ІТІНФ-20-1

Мазур Є.В.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник проф. Гороховатський В.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Кобилін О.А.
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Мазуру Єгору Володимировичу
(прізвище, ім'я, по батькові)1. Тема роботи Моделювання методу класифікації з використанням параметрів кластерного подання для опису зображень

затверджена наказом університету від 20 травня 2024 року № 464 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 27 травня 2024 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали наукових конференцій, дані інтернет-мережі, середовище розробки Visual Studio 2022, бібліотека-обгортка комп'ютерного зору OpenCVSharp.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Структурні методи аналізу і розпізнавання зображень у комп'ютерному зорі.

2. Моделі класифікації зображень із використанням кластерного подання.

3. Здійснення комп'ютерного моделювання методів класифікації із кластеруванням.

4. Аналіз результатів програмного моделювання методів.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми класифікації зображень, постановка задачі, впровадження кластерування даних, схеми роботи алгоритмів класифікації зображень з кластеруванням, таблиці значень результатів класифікації (точність та швидкодія), гістограмні характеристики класифікації.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.04.2024	
2	Аналіз завдання, підбір літератури	08.04.24-16.04.24	
3	Аналіз літератури з досліджуваної проблеми	17.04.24-19.04.24	
4	Аналіз технічних засобів	20.04.24-24.04.24	
5	Розробка методів	25.04.24-15.05.24	
6	Програмна реалізація	16.05.24-23.05.24	
7	Оформлення пояснювальної записки	24.05.24-27.05.24	
8	Перевірка на плагіат	28.05.24	
9	Рецензування	29.05.24	
10	Підготовка презентації та доповіді	30.05.24-05.06.24	
11	Занесення роботи в електронний архів	06.06.24	
12	Попередній захист кваліфікаційної роботи	06.06.24	

Дата видачі завдання 8 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

проф. Гороховатський В.О.
(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 58 с., 11 табл., 20 рис., 45 джерел.

КОМП'ЮТЕРНИЙ ЗІР, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, ДЕСКРИПТОРИ КЛЮЧОВИХ ТОЧОК, КЛАСТЕРИЗАЦІЯ, АПРОКСИМАЦІЯ ОПИСУ.

Об'єктом роботи є кластерне подання для опису зображень в задачі класифікації.

Метою роботи є розробка методів класифікації, що використовують кластерне подання для опису зображень, отримані завдяки спільній кластеризації всієї бази описів еталонних зображень та кластеризації кожного еталону окремо.

Використано метод k -середніх для здійснення кластеризації структурного опису у формі множини дескрипторів ключових точок. Множина дескрипторів сформована детектором BRISK. Досліджено результативність та швидкодію методів класифікації зображень з роздільною та спільною кластеризацією векторів ознак.

У результаті роботи програмно реалізовані традиційний метод класифікації зображень, метод класифікації зі спільною кластеризацією та з роздільною кластеризацією множини еталонів.

COMPUTER VISION, IMAGE CLASSIFICATION, DESCRIPTORS, CLUSTERING, DESCRIPTION APPROXIMATION.

The object of the work is the cluster representation for image description in the image classification problem.

The aim of the work is to develop classification methods that use a cluster representation for image description, obtained by joint clustering of the entire base of reference image descriptions and clustering of each reference separately.

The k -means method was used to implement the clustering of the structural description in the form of a set of keypoint descriptors. The set of descriptors is formed by the BRISK detector. The effectiveness and speed of image classification methods with separate and joint clustering of feature vectors were studied.

As a result of work, the traditional image classification method, the methods of classification with joint clustering and with separate clustering of a set of reference images were implemented.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ.....	7
1 Методи аналізу зображень у комп'ютерному зорі	9
1.1 Термінологія і основні задачі комп'ютерного зору	9
1.2 Аналіз детектора BRISK для виявлення ключових точок.....	11
1.3 Аналіз алгоритму кластеризації <i>k</i> -means.....	16
1.4 Постановка задачі	18
2 Моделі класифікації зображень із використанням кластерного подання...	19
2.1 Формалізація структурного методу класифікації зображень.....	19
2.2 Аналіз метрик для обчислення релевантності множин векторів...	21
2.3 Моделі застосування мультимножин	24
2.4 Використання кластеризації у методі класифікації	26
2.5 Аналіз методів класифікації зображень з використанням кластерного подання опису.....	29
3 Результати комп'ютерного моделювання	33
3.1 Вибір середовища програмної реалізації	33
3.2 Особливості програмної реалізації	34
3.3 Інструкція користувача	37
3.4 Аналіз результатів тестування.....	40
Висновки	53
Перелік джерел посилання	54

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

КТ – ключові точки

BRISK – Binary Robust Invariant Scalable Keypoints

FAST – Features from Accelerated Segment Test

BRIEF – Binary Robust Independent Elementary Features

SIFT – Scale-invariant Feature Transform

SURF – Speeded-Up Robust Features

ORB – Oriented FAST and Rotated BRIEF

AKAZE – Accelerated-KAZE

OpenCV – Open-Source Computer Vision Library

ВСТУП

У сфері комп'ютерного зору та системах, що у ній розробляються, часто постає задача створення методів класифікації зображень, де кожен окремий клас представляється у вигляді множин дескрипторів ключових точок [1–7]. У деяких з таких систем пріоритет ставиться на параметр швидкодії виконання обчислень. Такими системами зазвичай є системи, що виконують завдання комп'ютерного зору у реальному часі.

Важливими критеріями при виборі алгоритму класифікації зображень є швидкодія та точність класифікатора. Як спроби пришвидшення роботи систем комп'ютерного зору створені різні алгоритми виявлення та опису ключових точок зображень, що орієнтуються на пришвидшення опису. Прикладами таких алгоритмів є BRISK та SURF [5, 6]. Ці алгоритми знаходять ключові точки та описують їх у вигляді бінарних векторів уніфікованого розміру з урахуванням впливу масштабу та повороту. Бінарність цих векторів дозволяє використовувати швидкісні бінарні оператори для розрахунку відстані, а також використання простих метрик відстані для спрощення та прискорення їх розрахунку.

Як метод редукції множини векторів опису зображень та апроксимації цих даних задля підвищення швидкодії можна використовувати кластеризацію. Одним з сучасних методів кластеризації є метод k -середніх. Цей алгоритм є гнучким до використання будь-яких метрик відстані, а також є одним з найпростіших у реалізації [2, 7]. До того ж результативність алгоритму на великих обсягах даних дає змогу пришвидшити етап створення кластерного подання дескрипторів для використання у методах класифікації.

Актуальність роботи полягає у важливості створення швидкісних алгоритмів класифікації зображень для використання комп'ютерними системами, що потребують швидкодії обчислень для ефективного функціонування.

Практична значущість бакалаврської роботи полягає у визначенні швидкодії та точності завдяки експериментальному оцінюванню програмно реалізованих методів, що орієнтуються на модифікацію простору даних дескрипторів ключових точок, отриманих з прикладної бази зображень.

Завданням роботи є проектування та розробка методів класифікації зображень, що спираються на два різних варіанти структурного опису зображень у вигляді кластерного подання дескрипторів для пришвидшення розрахунків за рахунок потенційного незначного зниження точності.

1 МЕТОДИ АНАЛІЗУ ЗОБРАЖЕНЬ У КОМП'ЮТЕРНОМУ ЗОРІ

1.1 Термінологія і основні задачі комп'ютерного зору

Комп'ютерний зір – це галузь інформатики, що полягає у створенні програм або пристроїв, що здатні аналізувати візуальні дані та вирішувати певні задачі на основі цих даних [1, 8–10].

Задачі, які пов'язані з комп'ютерним зором по своїй суті є дуже складними. Складність задач полягає в тому, що вони є зворотними, тобто у них потрібно віднайти певне невідоме, маючи недостатню інформацію, щоб дати точну відповідь, через що ці задачі зводяться до побудови ймовірнісних моделей [2]. Прикладами задач комп'ютерного зору є такі завдання:

- класифікація зображень (Image classification). У цій задачі система повинна проаналізувати зображення та класифікувати його, тобто дати цьому зображенню мітку, що описує його належність до певного заздалегідь описаного класу;

- виявлення об'єктів (Object detection). У цій задачі, використовуючи класифікацію зображень, система повинна ідентифікувати певний об'єкт на фото або відео та виявити їх зовнішній вигляд;

- відстеження об'єктів (Object tracking). У цій задачі, система повинна знайти об'єкт та відстежити його рух;

- пошук зображень за основі вмісту (Content-based image retrieval). Суть задачі полягає у знаходженні зображень на основі вмісту вхідного зображення, не використовуючи метадані.

Як область досліджень, комп'ютерний зір виник більш ніж 60 років тому. Разом з винаходом технології сканування зображень та перетворення двовимірних зображень у тривимірні об'єкти, а також початку розвитку поля штучного інтелекту, комп'ютерний зір став задачею, яку фахівці все ще намагаються вирішити. Важливість технології та зацікавленість спеціалістів, звичайно, полягає в тому, що здатність комп'ютерів «бачити» відкриває

безмежну кількість можливостей для створення програмного забезпечення, що спрощує повсякденні завдання людей. Такі функції як автоматичний переклад тексту з використанням камери у Google Translate та автопілот на автомобілях є наглядними прикладами інтеграції технологій комп'ютерного зору у життя людини.

Серед задач комп'ютерного зору, класифікація зображень є однією з найважливіших та найскладніших. Від неї залежать такі завдання як, наприклад, сегментація, виявлення об'єктів та оцінка пози людини на зображенні.

Класифікація зображень – це завдання, що полягає у присвоєнні вхідному зображенню певної заздалегідь описаної мітки, тобто віднесення його до певного класу (групи зображень) [1]. Сам принцип роботи класифікації зображень полягає у порівнянні вхідного зображення з базою заздалегідь помічених зображень.

У класифікації зображень існує надзвичайно важливе підготовче завдання, що забезпечує надійність отриманих даних – виділення ознак [4, 5]. Виділення ознак – процес, що полягає у знайденні найбільш важливих ознак із вхідних даних та представлення їх у більш спрощеному вигляді для використання класифікаторами [5]. До виділення ознак, дані є надто великими для обробки та містять занадто багато нерелевантної інформації, що відображається не тільки на ефективності та якості класифікації, а і можливості її проведення. При роботі з зображеннями, такими ознаками є ключові точки та їх дескриптори.

Ключові точки (keypoints) – це точки, що вказують на помітні та важливі зони зображення. Дескриптори ключових точок – це вектори, що описують інформацію, отриману з ключових точок. Для отримання ключових точок та дескрипторів існують детектори ключових точок.

В ідеальному варіанті, детектори ключових точок повинні знаходити такі ключові точки, що можна відслідкувати на зображенні, на якому виконані різні геометричні перетворення (масштаб, зміщення, поворот), а дескриптори

цих точок повинні містити якомога більш важливу та відмінну інформацію, щоб її можна було впізнати при порівнянні.

1.2 Аналіз детектора BRISK для виявлення ключових точок

BRISK – алгоритм виявлення та опису ключових точок, розроблений на основі алгоритму виявлення кутів зображення FAST. Метою розробки алгоритму BRISK було створення швидкого алгоритму генерації надійних ключових точок, який можна використовувати у застосунках, що виконують розрахунки у реальному часі [5].

Такі алгоритми як SIFT є надзвичайно надійними, але потребують достатньо велику кількість часу на обчислення. З іншого боку, такий підхід як комбінація виявлення КТ FAST та опису КТ BRIEF дають швидкодію ціною недостатньої точності отриманих даних.

BRISK розшифровується як Binary Robust Invariant Scalable Keypoints, тобто бінарні надійні інваріантні масштабовані ключові точки. Інваріантність ключових точок у детекторах – це властивість, що вказує на те, що незважаючи на звичайні геометричні перетворення, ключові точки є постійними та можуть бути легко знайдені на оригінальному зображенні. Слід зазначити, що попри назву, точки отримані BRISK не є повністю інваріантними до змін у масштабі та ротації, але мають механізми на етапах виявлення та опису точок, що зменшують ефект цих перетворень на якість отриманих описів зображень.

BRISK має два основних етапи своєї роботи – знаходження та опис ключових точок. Критерієм визначення важливості точки є оцінка s алгоритму FAST. Алгоритм FAST вишукує кути на зображенні. Кутом вважається коло з 16 суміжних пікселів (вигляд кола зображений на рисунку 1.1), у якому n з 16 пікселів є світліші за суму інтенсивності центру та порогу t , або темніші за різницю інтенсивності центру та порогу t [6].

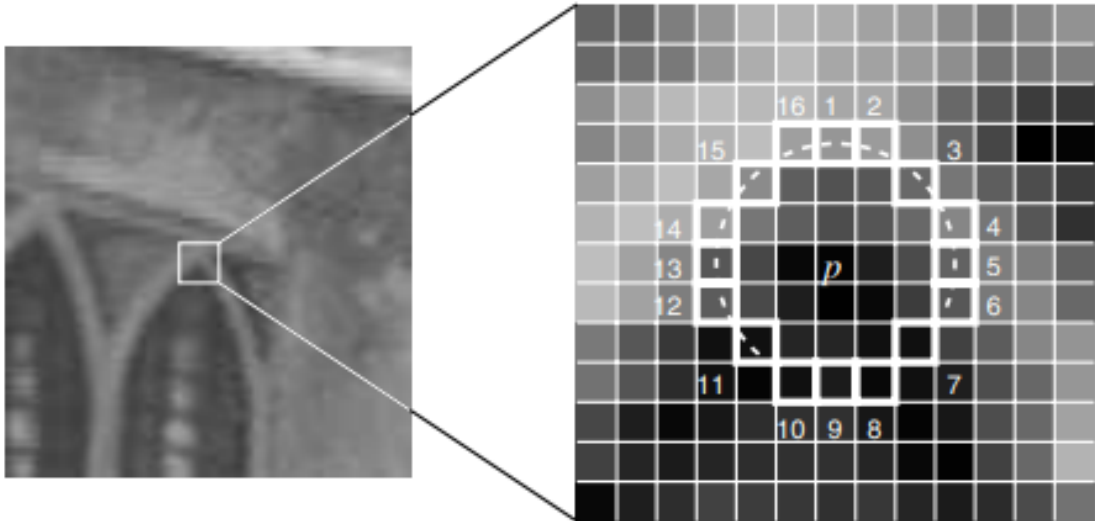


Рисунок 1.1 – Центр та коло кута-кандидата

Алгоритм FAST для $n = 12$ працює так:

Крок 1. Обираємо центр кута-кандидата p . Надалі важливим параметром є його інтенсивність I_p . Інтенсивність це яскравість пікселя від 0 до 255.

Крок 2. Розглядаємо коло розміром в 16 пікселів навколо обраного пікселя.

Крок 3. Розглядаємо пікселі 1, 5, 9 та 13 (середні верхній, правий, нижній та лівий пікселі на колі). Якщо хоча б 3 з них виконують умову, що їх інтенсивність більша за $I_p + t$, або менша за $I_p - t$, то це коло є кутом, тобто, ключовою точкою. Інакше – відкидаємо піксель-кандидат.

Крок 4. Повторюємо для кожного пікселя на зображенні.

Проте BRISK розраховує оцінку FAST не тільки для початкового зображення, а й для простору масштабів, щоб отримати відносну інваріантність до масштабування зображення.

Простір масштабів є пірамідою, що складається з n октав c_i (де $i = \{0, 1 \dots n - 1\}$) та n внутрішніх октав d_i – вхідне зображення різних масштабів. Множник кожної октави, що показує на масштаб розраховується як $t(c_i) = 2^i$ та $t(d_i) = 2^i \cdot 1,5$. BRISK спочатку використовує маску 9-16 на кожній з октав для виявлення потенційно важливих точок, тобто на колі з 16 пікселів, 9 з них повинні мати необхідну інтенсивність ($n = 9$). Після цього,

відбувається відкидання немаксимальних точок. Точка відкидається, якщо вона не отримує максимальну оцінку FAST серед 8 сусідніх точок навколо неї на своїй та сусідніх октавах.

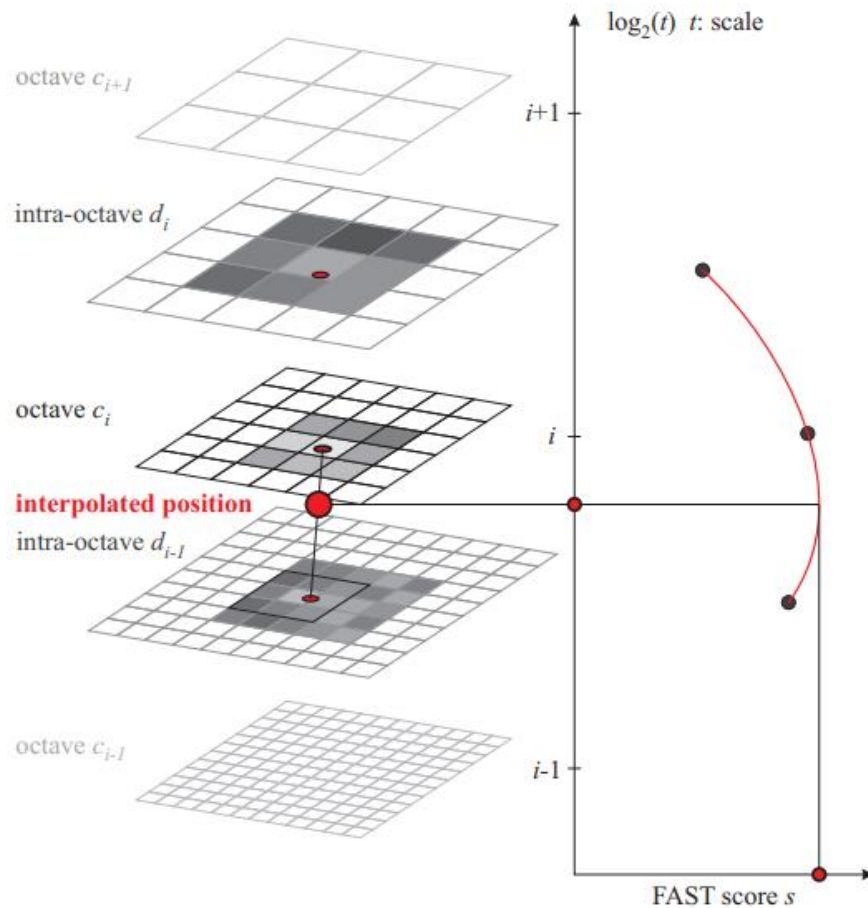


Рисунок 1.2 – Знаходження ключових точок BRISK на просторі масштабів

Передостаннім етапом є субпіксельне та безперервне уточнення масштабу, що полягає в визначенні локальних максимумів помітності на трьох сусідніх октавах та оцінці реального масштабу ключової точки завдяки підгонці результатів під одновимірну параболу. Останнім етапом є повторна інтерполяція позиції ключової точки.

Другий етап – це опис ключових точок, тобто процес створення їх дескрипторів. Дескриптори BRISK бінарні та складаються з 64 байтів (512 бітів), що дозволяє легко їх порівнювати використовуючи відстань Геммінга. Початковий етап побудови дескриптора – знаходження вибірки

точок навколо КТ. BRISK робить це за патерном, зображеним на рисунку 1.3. Сині точки на рисунку – точки вибірки, а червоні окружності – дисперсія Гаусівського фільтра, який застосовується на зображенні для згладжування.

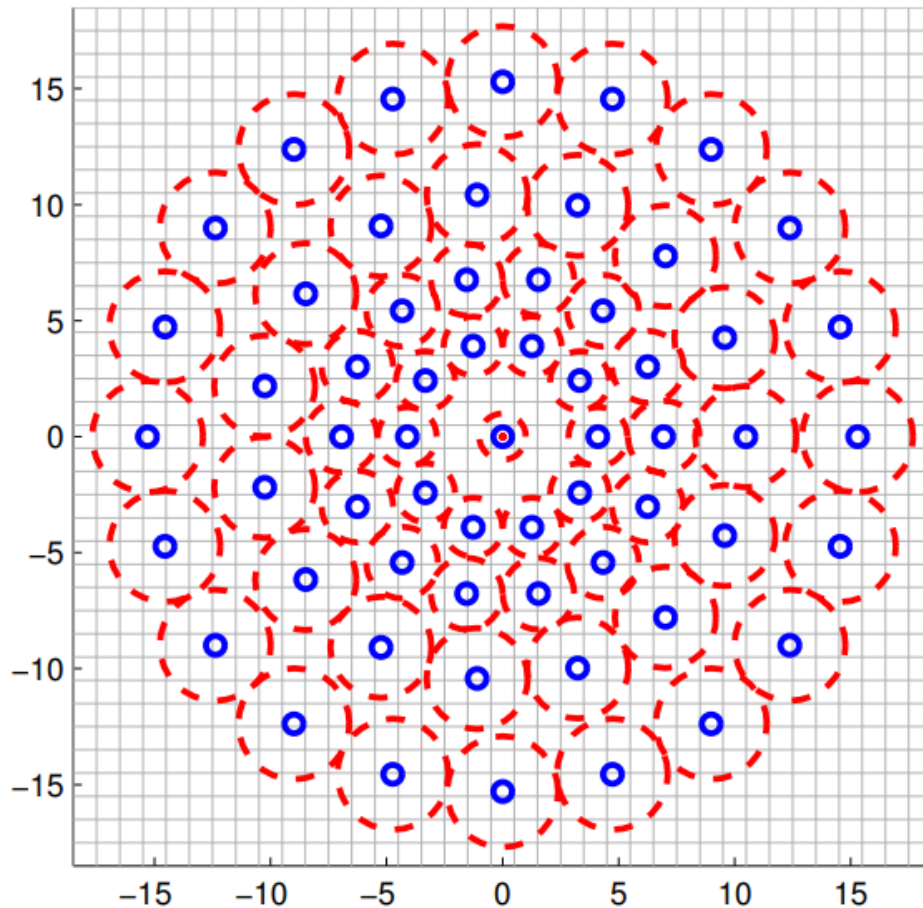


Рисунок 1.3 – Патерн знаходження точок вибірки BRISK

Після цього йде етап оцінки повороту ключової точки та її нормалізації за кутом повороту. Для цього визначаються множини дальніх та ближніх пар точок. Ближні пари – пари точок, відстань яких менша за певний поріг δ_{max} , а дальні – такі, що їх відстань більша за δ_{min} .

Для знаходження напрямку ключової точки розраховується сума локальних градієнтів між точками з множини ближніх пар. Локальні градієнти при цьому розраховуються за такою формулою:

$$g(p_i, p_j) = (p_j - p_i) \cdot \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|^2},$$

де p_i та p_j – точки відповідної пари, а $I(p, \sigma)$ – інтенсивність відповідної точки з урахуванням дисперсії Гаусівського фільтру, застосованого до точки. Далі на зображення знов накладається патерн з рисунку 1.3, але повернутий на кут α , рівний арктангенсу координати y вектору напрямку КТ, поділену на координату x . Саме це і повинно нормалізувати КТ за кутом повороту для отримання відносної інваріантності до нього.

На останньому кроці будується дескриптор. Принцип побудови дескриптора дещо схожий на знаходження кутів у FAST та виконується за формулою:

$$b = \begin{cases} 1, & I(p_j^\alpha, \sigma_j) > I(p_i^\alpha, \sigma_i) \\ 0, & \text{інакше} \end{cases}.$$

Точки p_i та p_j у цьому випадку беруться з множини ближніх пар з повернутого на α патерну.

Враховуючи всі ці особливості знаходження КТ та побудови дескрипторів, BRISK демонструє високий результат повторюваності КТ на зображеннях з різними геометричними перетвореннями (рисунок 1.4). Повторюваність – відношення між кількістю відповідних КТ до мінімальної загальної кількості КТ, видимих на обох зображеннях [5].

У порівнянні з детектором SURF, результати BRISK є достатньо схожими з, в середньому, невеликим відривом повторюваності у алгоритму SURF, проте BRISK в середньому витрачає на знайдення однієї точки в 11,5 разів менше часу, що є важливим фактором, особливо для програм, що потребують отримані від детектора дескриптори для виконання розрахунків в реальному часі.

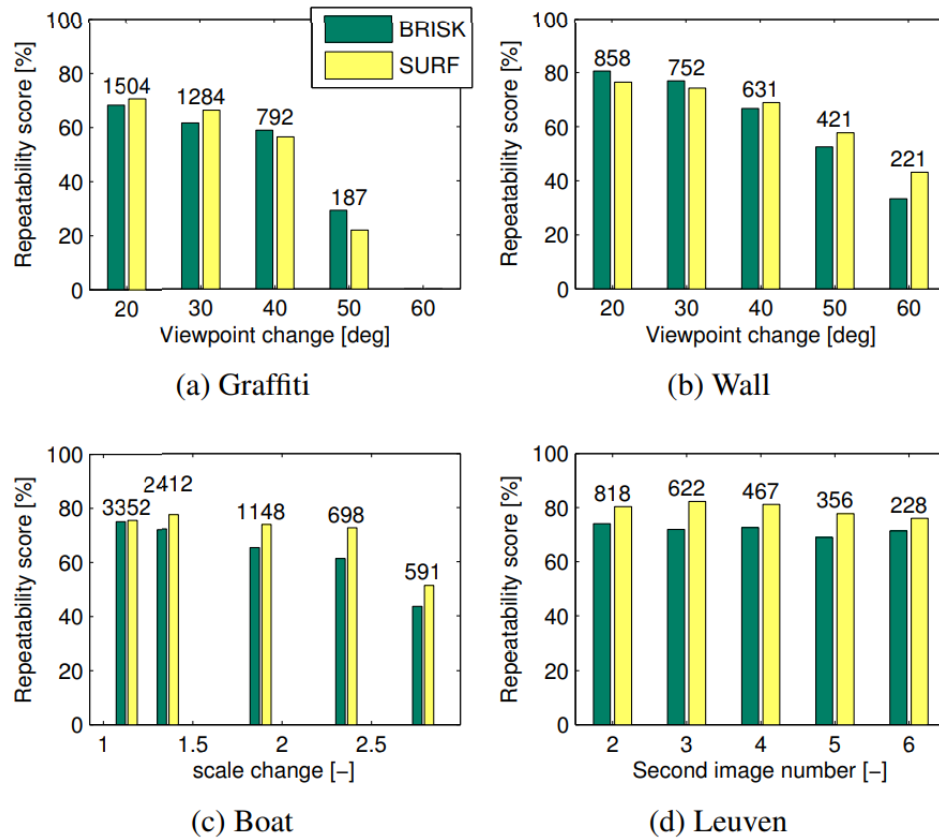


Рисунок 1.4 – Порівняння повторюваності КТ отриманих BRISK та SURF

Отже, через високий рівень інваріантності до зміни масштабу та повороту, високу швидкодію, а також простоту порівняння дескрипторів, є доцільним використання детектора BRISK для виділення та опису ключових точок для подальшого використання в задачі класифікації.

1.3 Аналіз алгоритму кластеризації k -means

Кластеризація – задача, що полягає у розподілі даних на групи (кластери) на основі певних властивостей. В результаті кластеризації, кожний кластер повинен містити схожі елементи.

Використання кластеризації у області комп'ютерного зору є достатньо поширеним явищем. Прикладом задачі комп'ютерного зору, у якій використовують кластеризацію є сегментація зображень.

Сегментація зображень – завдання знаходження груп пікселів, що поєднуються одне з одним, тобто є частиною певного окремого сегменту зображення [2].

Ідея використання кластеризації у завданні класифікації зображень полягає в кластеризації дескрипторів КТ та використанні центрів кластерів у якості векторів для порівняння для зменшення обчислювальної вартості та пришвидшення процесу класифікації.

Метод k -середніх (k -means clustering) – один з найпоширеніших методів кластеризації. Алгоритм k -means не потребує жодних імовірнісних розрахунків. Замість цього, алгоритму задається кількість кластерів k , які він повинен знайти, після чого він ітеративно оновлює місцезнаходження центрів кластерів на основі найближчих до відповідних центрів елементів. Поширеність методу k -середніх пов'язана зі швидкістю алгоритму при кластеризації великих наборів даних [7]. Швидкість алгоритму для різних метрик відстані для різних за обсягом наборів випадкових точок у двомірному просторі показана на рисунку 1.5.

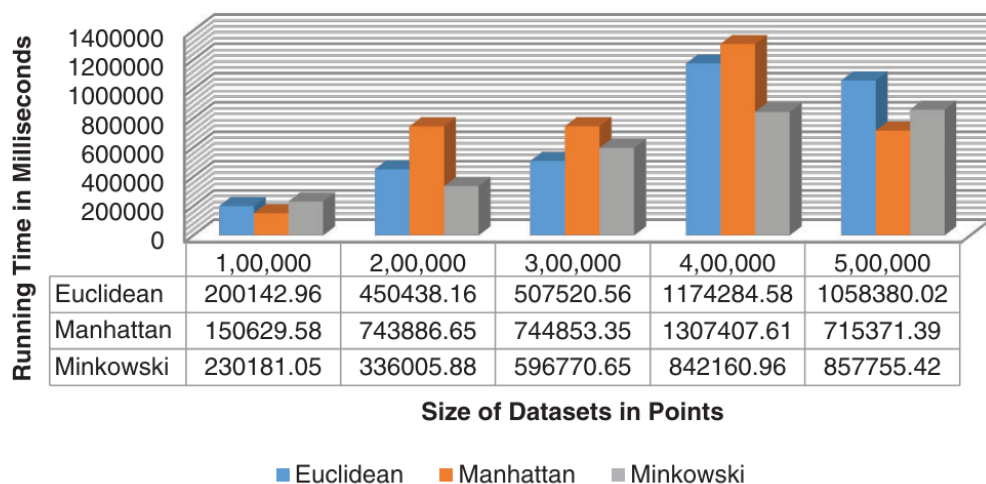


Рисунок 1.5 – Результати кластеризації для різних метрик відстані методом k -середніх на 10 кластерів

Враховуючи простоту роботи, легкість модифікації методу для різних даних і метрик відстані та велику швидкість, для завдання кластеризації є

доцільним використання методу k -середніх. Для отримання центрів у вигляді бінарних векторів, були зроблені модифікації методу, описані в частині 2.

1.4 Постановка задачі

Розроблення методів класифікації зображень із використанням параметрів кластерного подання для опису зображень є актуальною задачею у системах комп'ютерного зору.

Метою роботи є підвищення швидкодії методів класифікації зображень на підставі впровадження кластеризації структурних описів сукупно для бази зображень та окремо для кожного еталону.

Для досягнення мети необхідно вирішити такі завдання:

- виконати аналіз існуючих методів класифікації зображень;
- здійснити моделювання традиційного методу класифікації з лінійним пошуком;
- реалізувати методи класифікації зображень на основі кластерних центрів повної бази зображень і окремо для кожного з еталонів;
- сформувати набір тестових зображень для здійснення комп'ютерного моделювання;
- провести тестування реалізованих методів та оцінити доцільність використання розроблених методів на основі параметрів точності та швидкодії.

2 МОДЕЛІ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ ІЗ ВИКОРИСТАННЯМ КЛАСТЕРНОГО ПОДАННЯ

2.1 Формалізація структурного методу класифікації зображень

Нехай $E_k = \{e_v(k)\}_{v=1}^s$ – множина дескрипторів певного еталону k з бази еталонів $E = \{E_1, E_2, \dots, E_N\}$, де $s = \text{card } E_k$ кількість елементів множини дескрипторів еталону. Нехай $Z = \{z_v\}_{v=1}^s$ – множина дескрипторів вхідного зображення, s – однакове з кількістю дескрипторів еталону для спрощення аналізу.

Традиційний метод класифікації працює у два етапи у вигляді $R = R_2 R_1$: перший (R_1) – локальна класифікація окремих дескрипторів, другий (R_2) – прийняття остаточного рішення щодо класу зображення, що аналізується, на основі голосів, отриманих з першого етапу.

На першому етапі класифікація окремих дескрипторів відбувається завдяки принципу лінійного пошуку, порівнюючи кожний дескриптор з множини вхідного зображення з усіма дескрипторами з множини дескрипторів кожного еталону за принципом формули (2.1) [8, 30].

$$R_1: z_v \rightarrow \{1, \dots, N\}. \quad (2.1)$$

Визначення класу k , до якого відноситься відповідний дескриптор вхідного зображення, відбувається завдяки знаходженню класу найближчого дескриптора із множини дескрипторів еталонів по відношенню до нього за принципом формули (2.2) [30, 35].

$$R_1: k = \arg \min_{i=1, \dots, N; d=1, \dots, s} \rho(z_v, e_d(i)), \quad (2.2)$$

де $\rho(z_v, e_d(i))$ – відстань між дескриптором вхідного та еталонного зображення. Такою метрикою у цьому випадку є відстань Геммінга через ефективність обчислення у бінарному просторі.

Відстань Геммінга – число бітів у двох бітових векторів, що є відмінними одне від одного. По суті, розрахунок відстані Геммінга є бітовою операцією виключного «або» з подальшим підрахунком бітів [5, 24].

$$\rho(b_k, z_v) = \sum_{a=1}^n 1(b_{k,a}, z_{v,a}), \quad (2.3)$$

де функція $1(b_{k,a}, z_{v,a})$ порівнює біти з номером a для обох векторів та дорівнює одиниці, якщо біти не співпадають, а інакше – нулю.

Для реалізації другого етапу та визначення класу k вхідного зображення Z , створюється вектор $\{h_i\}_{i=1}^N$ цілих чисел з кількістю дескрипторів, віднесених до кожного класу. Маючи результати класифікації за R_1 , для кожного $z_v \in Z$, інкрементно збільшується лічильник $h_k = h_k + 1$. Після розрахунку $\{h_i\}_{i=1}^N$, клас об'єкту Z визначається як клас з максимальною кількістю голосів у векторі $\{h_i\}_{i=1}^N$.

$$R_2: Z \rightarrow E_k | (k = \arg \max_{i=1, \dots, N} h_i) \& (h_k \geq \delta_h), \quad (2.4)$$

де δ_h – деякий поріг для мінімально дозволеної кількості голосів. За невиконання умови $h_k \geq \delta_h$, клас об'єкту не визначається (зображенню відмовлено від класифікації через відсутність підстави віднести його до жодного із класів). Поріг δ_h визначається експериментально і, як правило, дорівнює мінімальному необхідному порогу для безперешкодної класифікації еталонних зображень. Схема роботи традиційного класифікатора продемонстрована на рисунку 2.1.



Рисунок 2.1 – Схема роботи класифікатора за множиною дескрипторів

2.2 Аналіз метрик для обчислення релевантності множин векторів

При порівнянні множин векторів важливим етапом є вибір підходящої для задачі метрики ρ . Метрика є кількісною величиною, що допомагає визначити еквівалентність множин або належність певних елементів до множини. У задачі класифікації обрана метрика повинна давати конкретне уявлення про віднесення елементу до класу [9–15].

Правильно обрана метрика для множини дескрипторів КТ дозволяє зменшити вплив завад як наявних на зображенні, так і завад, що виникають при генерації КТ та їх описі. Прикладом останніх є завади, що виникають у результаті неточностей при розрахунку куту повороту на етапі опису КТ у детекторі BRISK.

Найпростіша метрика для порівняння множин є метрика ізольованих точок. При порівнянні двох векторів a та c вона буде виглядати як на формулі (2.5) [8, 19, 45].

$$\rho(a, c) = \begin{cases} 0, & a = c, \\ 1, & a \neq c. \end{cases} \quad (2.5)$$

Метрика по суті є схожою на відстань Геммінга (2.3). Різниця між ними полягає у тому, що порівняння йде саме для визначення повної еквівалентності векторів, а не порівняння схожості їх окремих елементів. Враховуючи вигляд дескрипторів КТ, ця метрика не є ефективною для задачі порівняння множин описів, оскільки не дає жодного уявлення про їх подібність.

Варіантом поліпшення цієї метрики для врахування неточності при описі дескрипторів КТ є введення порогу δ при порівнянні. Формула такої метрики описана у (2.6) [8, 16–24].

$$\rho(a, c) = \begin{cases} 0, & O_\delta(a) \cap O_\delta(c) \neq \emptyset, \\ 1, & O_\delta(a) \cap O_\delta(c) = \emptyset \end{cases} \quad (2.6)$$

де $O_\delta(a) = \{x: \rho(a, x) \leq \delta\}$ – множина елементів, відстань яких до елемента a не перевищує зазначений поріг δ . Оновлену метрику можна уявити як пересічення шарів навколо точок a та c з радіусом δ . Тобто ця метрика вважає множини a і c еквівалентними якщо ці шари перетинаються, і тоді відстань дорівнює 0, в іншому випадку – 1.

Інша метрика, що допомагає встановити схожість множин на основі їх компонентів – Манхеттенська відстань (2.7). Ця метрика визначає схожість елементів a та c з множин завдяки розрахунку суми модулів різниць окремих значень цих елементів, вважаючи їх окремими координатами у просторі [8, 43].

$$\rho(a, c) = \sum_{k=1}^n |a_k - c_k|. \quad (2.7)$$

При цьому еквівалентність множин в залежності від задачі та даних може встановлюватись як повна рівність, тобто $\rho(a, c) = 0$, або відносна рівність, що не перевищує похибку δ , тобто $\rho(a, c) \leq \delta$. У випадку з множинами векторів дескрипторів підходящим є саме останній варіант, що використовує поріг [25–28].

Розглянемо метрики для множин у вигляді $A = \{a^1, a^2 \dots a^{N_A}\}$ та $C = \{c^1, c^2 \dots c^{N_C}\}$, де $N_A = \text{card } A$ та $N_C = \text{card } C$, тобто множин, що складаються з окремих векторів дескрипторів КТ, що належать підмножині \mathfrak{Z} загальної множини \mathfrak{Z}_A . При цьому, множини простору \mathfrak{Z}_A мають окрему метрику ρ_A . Такі метрики можуть виступати у якості альтернативи голосуванню для визначення найближчого класу до множини дескрипторів вхідного зображення.

Прикладом метрики простору \mathfrak{Z}_A є метрика Хаусдорфа, що наведена у формулі (2.8) [8, 29–33]:

$$\rho_A(A, C) = \max[\max_{a \in A} \min_{c \in C} \rho(a, c), \max_{c \in C} \min_{a \in A} \rho(a, c)], \quad (2.8)$$

де $\rho(a, c)$ є метрикою з простору \mathfrak{Z} (наприклад Манхеттенська відстань). Тобто відстань Хаусдорфа є максимальною відстанню серед найбільших найближчих відстаней простору \mathfrak{Z} від точок множини A до C і навпаки. Більш простою величиною, що є схожою на відстань Хаусдорфа є метрика найближчого сусіда, що вважає результатом найменшу відстань серед дескрипторів обох множин (2.9) [34–38].

$$\rho_A(A, C) = \min_{a \in A, c \in C} \rho(a, c). \quad (2.9)$$

Недоліком метрик (2.8) та (2.9) є те, що вони вважають важливими у визначенні результату лише деякі з відстаней між елементами множин, що призводить до нестійкості до деяких видів завад. Іншим варіантом метрики простору множин \mathfrak{Z}_A є метрика середнього зв'язку (2.10) [8, 39].

$$\rho_A(A, C) = \frac{1}{N_A N_C} \sum_{a \in A} \sum_{c \in C} \rho(a, c). \quad (2.10)$$

Такий варіант метрики виражає еквівалентність множин у вигляді середнього арифметичного відстаней усіх елементів від множини A до множини C та в інший бік. Величина, на відміну від (2.8), враховує відстані кожного елемента у результаті, що робить її більш стійкою до адитивних викривлень при генерації дескрипторів.

Однією з найефективніших метрик для порівняння множин дескрипторів та мультимножин є метрика Танімото [34, 44].

$$\rho_A(A, C) = \frac{A \Delta B}{A \cup B}. \quad (2.11)$$

Метрика Танімото порівнює множини завдяки знаходженню частки їх розбіжних елементів (симетрична різниця множин) та загальної кількості елементів множин. Метрика є схожою на метрику Геммінга. Їх схожість полягає у підрахунку кількості схожих елементів множин, але на відміну від відстані Геммінга, у (2.11) результат подається у вигляді числа від 0 до 1, а не як ціле число від 0 до потужності об'єднання множин $\text{card } A \cup B$.

Метрики (2.5–2.7) можуть бути використані безпосередньо для порівняння або векторів дескрипторів, або векторів, що є результатами голосування за кластери як у розділі 2.4. Тоді як метрики (2.8–2.11) можна використати для порівняння множин замість голосування. Результат залежить від точності та кількості сторонніх завад на етапі визначення КТ та їх опису [35, 40–43].

2.3 Моделі застосування мультимножин

Розглянемо множину $U = \{u_1, u_2, \dots, u_n\}$, що складається з n різних елементів u_i . Мультимножиною A називається підмножина U , що складається з елементів, що підпорядковуються правилу $A = \{k_A(u) \bullet u \mid u \in U, k_A(u) \in C_+\}$,

де C_+ – множина цілих невід’ємних чисел. При цьому k_A – функція, що показує скільки разів елемент $u \in U$ входить до A . Мультимножина вважається звичайною множиною, якщо у ній не зустрічаються жодних повторень елементів, тобто якщо для всіх елементів виконується умова з формули (2.12).

$$k_A(u) = \chi_A(u) = \begin{cases} 1, & u \in A \\ 0, & u \notin A \end{cases} \quad (2.12)$$

де χ_A – функція, що визначає відмінні елементи мультимножини. Іншими словами, мультимножиною називається така множина, різні елементи якої можуть входити до її складу декілька разів та хоча б один з них повторюється більше одного разу.

При розгляді мультимножин важливо враховувати характеристики, що описують їх, а саме: потужність $card(A) = \sum_{u \in U} k_A(u)$, розмірність $dim(A) = \sum_{u \in U} \chi_A(u)$ та висота $hgt(A) = \max_{u \in U} k_A(u)$. Потужність показує загальну кількість елементів у мультимножині, розмірність вказує на кількість неповторних елементів множини, а висота – на кількість повторень елемента мультимножини, що повторюється найбільше.

Використання апарату мультимножин є можливим у задачі класифікації зображень на основі описів КТ через допустиму схожість та повторюваність у множинах дескрипторів [9, 32, 35, 44]. Завдяки цьому задача класифікації зображень зводиться до максимізації параметру міри схожості \mathcal{G} опису зображень та еталонів (2.13).

$$j = \arg \max_{i \in [1, \dots, N]} \mathcal{G}[Z, E_i], \quad (2.13)$$

де N – кількість еталонів. Схема структурної класифікації на основі кластерного подання як варіант виконання задачі класифікації з використанням математичних принципів мультимножин зображена на рисунку 2.2.

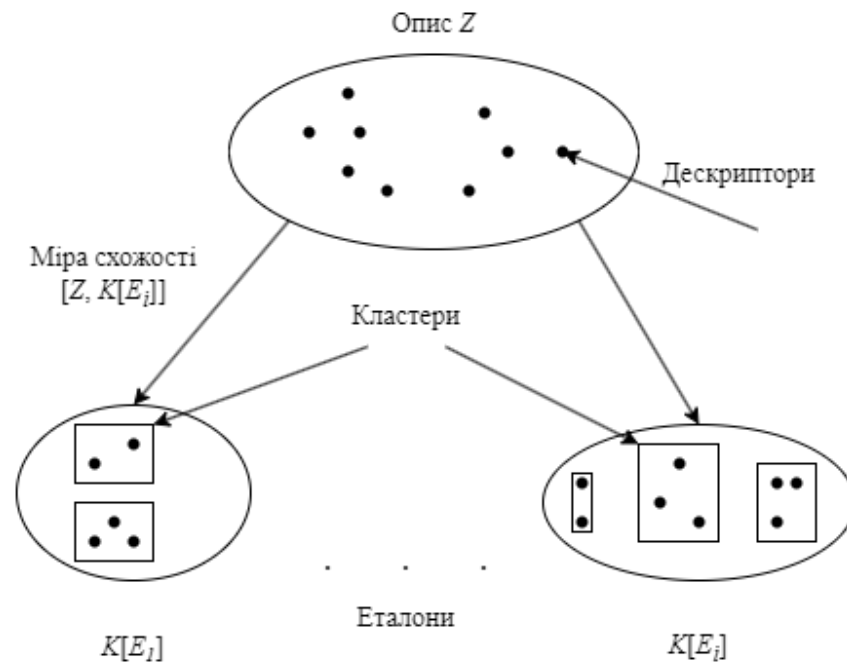


Рисунок 2.2 – Схема структурної класифікації на основі кластерного подання

При цьому міра схожості \mathcal{S} може розглядатись як міра схожості між множиною Z та кластером $K[E_i]$, або як міра схожості між точкою z_v та кластером $K[E_i]$. У першому випадку застосовуються метрики міри схожості, що визначають відстань між множинами як метрики простору \mathfrak{Z} , наприклад метрика Танімото (2.11). Слід зазначити, що розрахунок симетричної різниці мультимножин A та B відбувається за формулою (2.14).

$$A \Delta B = \{k_{A \Delta B}(u) \bullet u \mid k_{A \Delta B}(u) = |k_A(u) - k_B(u)|\}. \quad (2.14)$$

У другому варіанті у якості міри схожості обираються метрики для визначення належності точки до множини, ідентично до метрик простору \mathfrak{Z}_A .

2.4 Використання кластеризації у методі класифікації

Кластеризація є ефективним методом апроксимації для спрощення опису зображень та підвищення швидкодії класифікації. Виконаємо

відображення $E \rightarrow T$ із множини дескрипторів еталонних зображень у множини T непересічних кластерів. Кожний образ E_i еталону далі перетворюється у M непересічних підмножин $T_k(E_i)$ [40–42].

$$E_i = T(E_i) = \bigcup_{k=1}^M T_k(E_i), k = \overline{1, M}, T_k(E_i) \cap T_j(E_i) = \emptyset. \quad (2.15)$$

З підмножин, отриманих шляхом кластеризації, отримуємо центри кластерів $b_{k,i}$, що є ключовим параметром та будуть використовуватись далі для класифікації замість множин описів. Виграш у швидкодії при кластеризації з'являється завдяки зменшенню кількості дескрипторів еталонів для порівняння з s до $N \cdot M$, де N – кількість еталонів, а M – кількість кластерів.

У звичайному випадку алгоритм k -середніх обирає M точок, які визначає як центри, що визначають кластер. Далі кожний елемент, що підлягає кластеризації, відноситься до кластеру, відстань до центру якого є найменшою. Центри кластерів далі перераховуються до ітерації, на якій не відбувається жодної зміни центрів. Графічно результат роботи методу кластеризації k -середніх продемонстрований на рисунку 2.3 [39].

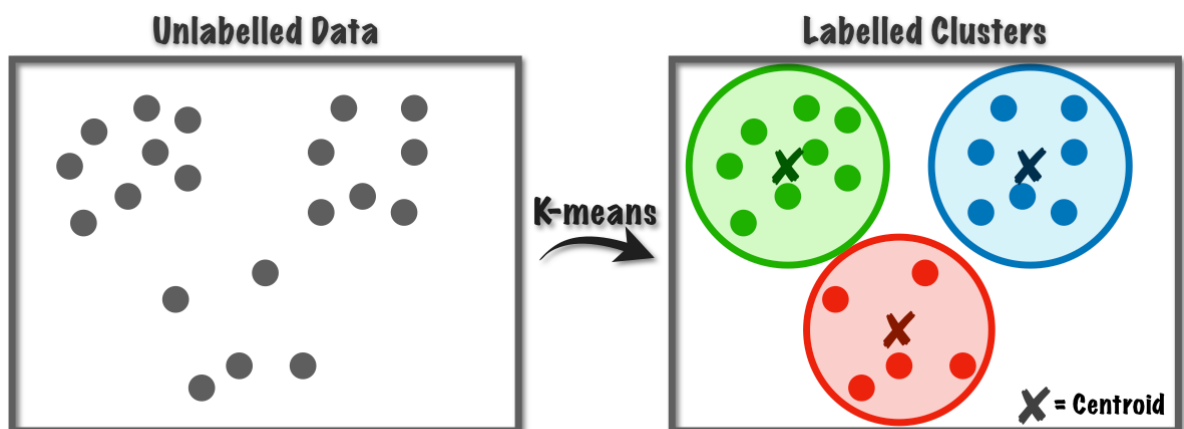


Рисунок 2.3 – Результат роботи методу k -середніх

Центри кластерів, отримані у результаті роботи алгоритму, можна вважати певним узагальненням множини елементів всього кластеру, через що

їх можна вважати валідними значеннями для апроксимації важливих елементів опису зображень за множиною їх дескрипторів.

У стандартному варіанті алгоритм k -середніх використовують для точок у багатовимірному просторі, де у якості метрики виступає Евклідова відстань, а розрахунок центрів кластерів на кожній ітерації відбувається з використанням координат точок кластерів [7, 42]. Такий підхід не є можливим для кластеризації одновимірних бінарних векторів, тому розрахунок кожного з центрів b_i кластерів у бінарному вигляді відбувається за формулою 2.16.

$$b_i(a) = \begin{cases} 1, & f(T^i, a) \geq s(T^i)/2, \\ 0, & \text{інакше,} \end{cases} \quad (2.16)$$

де $b_i(a)$ – значення біту на позиції a центру b_i , $f(T^i, a)$ – функція, що підраховує кількість одиниць на позиції a для кожного з елементів кластеру.

Похибка кластеризації при цьому розраховується як сума відстаней від елементів кластерів до відповідних центрів, поділена на кількість елементів, що приймали участь в кластеризації. Для представлення похибки у вигляді значення від 0 до 1, попереднє значення додатково ділиться на максимально можливу відстань Геммінга між цими векторами, що дорівнює їх розміру, тобто 512.

Як додатковий спосіб визначення якості кластеризації множини бінарних векторів дескрипторів, введений критерій відстані між центрами кластерів пар еталонних зображень β (це є допустимим лише у методі з роздільною кластеризацією через відсутність перемішування дескрипторів різних класів між собою на етапі кластеризації) [30].

$$\beta(A, B) = [\sum_{i=1}^M \sum_{j=1}^M \rho(a_i, b_j)]/M^2, \quad (2.17)$$

де a_i та b_j – центри з множини центрів пар еталонів $A = \{a_i\}_{i=1}^M$ та $B = \{b_j\}_{j=1}^M$.
Значення цього критерію знаходиться у межах відстані Геммінга для

дескрипторів BRISK (від 0 до 512). Зі збільшенням значення цього критерію збільшується різниця між класами A та B , що призводить до покращення можливості розрізнення класів. У практичній частині цей критерій використовуватиметься для визначення впливу зміни кількості кластерів для методів на точність класифікації.

2.5 Аналіз методів класифікації зображень з використанням кластерного подання опису

У кваліфікаційній роботі реалізовано два методи класифікації зображень, що спираються на кластерне подання описів зображення: метод зі спільною кластеризацією бази та метод із роздільною кластеризацією еталонів. Відмінність між методами полягає у різному підході до кластеризації множин дескрипторів еталонних зображень.

Для реалізації класифікації зображення з кластеризацією всієї бази даних вся множина дескрипторів $E = \{E_1, E_2, \dots, E_N\}$ розбивається на $N \cdot k$ кластерів методом k -середніх. Далі, кожне еталонне зображення та вхідне зображення окремо порівнюються з центрами кластерів, ідентично до формули (2.2) та формують вектори голосів $g(Z) = [g_1, g_2, \dots, g_M]$ ідентично до традиційного методу. Після цього, отриманий вектор голосів за кожен кластер від вхідного зображення порівнюється з кожним вектором голосів за кожен кластер від еталонних зображень та класифікує вхідне зображення за формулою (2.18), використовуючи іншу метрику – Манхеттенську відстань (μ).

$$R_2: Z \rightarrow E_v \mid v = \arg \min_{k=1, \dots, N} \mu(g(Z), g(E_k)). \quad (2.18)$$

Манхеттенська відстань μ (2.7) при цьому вираховується як сума модулів різниць голосів за окремі кластери. Формула для розрахунку Манхеттенської відстані для векторів голосів зазначена у формулі (2.19).

$$\mu(g(Z), g(E_k)) = \sum_{i=1}^M |g(Z)_i - g(E_k)_i|. \quad (2.19)$$

Потреба у створенні векторів голосів еталонних зображень за центри кластерів та використання Манхеттенської відстані для порівняння векторів голосів еталонів за центри кластерів з'являється через те, що центри, отримані в результаті кластеризації усієї бази еталонів утворюються з використанням дескрипторів з різних зображень, через що неможливо стверджувати, що будь-який зі сформованих центрів представляє собою апроксимоване уявлення про якийсь певний клас.

Схема принципу роботи методу класифікації зі спільною кластеризацією зображень показана на рисунку 2.4.



Рисунок 2.4 – Схема роботи методу зі спільною кластеризацією

Для виконання класифікації зображення з роздільною кластеризацією описів еталонів множина дескрипторів кожного еталону розбивається на k кластерів алгоритмом k -середніх, після чого поєднуються в один вектор з $N \cdot M$ центрів кластерів, з яким порівнюється вхідне зображення за відстанню Геммінга. У цьому випадку кожні M центрів описують своє відповідне

зображення, отже кількість голосів за клас є сумою голосів за M центрів кластерів, що відповідають зображенню.

Схема принципу роботи методу з роздільною кластеризацією зображена на рисунку 2.5.



Рисунок 2.5 – Схема роботи методу з роздільною кластеризацією

У порівнянні з методом зі спільною кластеризацією, цей метод уникає використання додаткової метрики та операції порівняння центрів кластерів з дескрипторами еталонів завдяки ізоляції даних, що описують еталони, у окремих операціях кластеризації. Через це, кожен M центрів у векторі центрів є результатом апроксимації даних відповідного зображення та можуть використовуватись напямую для класифікації вхідного зображення. Ціною уникнення цих додаткових операцій є необхідність у проведенні більшої кількості операцій кластеризації на підготовчому етапі.

Для методів зі спільною та роздільною кластеризацією, окрім порогу на Манхеттенську відстань δ_u та кількість голосів δ_h при класифікації, також використовується поріг на відстань Геммінга δ_p , що не пропускає до голосування дескриптори, відстань яких до найближчого дескриптору еталону є надто великою. Такий підхід дає змогу підвищити точність та завадозахищеність методів, що залежать від апроксимованих даних шляхом відкидання нерелевантних голосів.

Для оцінки ефективності класифікаторів під час здійснення тестування введено критерій точності pr , що визначається як відношення кількості q зображень, що були віднесені до коректного класу, до загальної кількості Q проведених експериментів.

$$pr = q/Q. \quad (2.20)$$

Критерій точності pr варіюється у межах значень від нуля до одиниці, де 1 – безпомилкове визначення класів усіх поданих зображень до класифікації методом, що випробовується.

3 РЕЗУЛЬТАТИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ

3.1 Вибір середовища програмної реалізації

У рамках кваліфікаційної роботи були розроблені два алгоритми для класифікації зображень з використанням параметрів кластерного подання для опису зображень. Для реалізації функціоналу програми була використана обгортка бібліотеки OpenCV для мови програмування C# OpenCVSharp [36].

OpenCV – бібліотека з відкритим кодом, розроблена компанією Intel, що містить велику кількість функцій та алгоритмів комп'ютерного зору. OpenCV налічує більше 2500 оптимізованих класичних та більш нових алгоритмів, включаючи детектор КТ BRISK. Функціонал OpenCV може бути використаний для завдань виявлення об'єктів, сегментації зображень, пошуку зображень на основі вмісту, аналізу руху та стеження за об'єктом, створення 3D моделей об'єктів, виконання перетворень зображень та відео. OpenCV також містить вбудовану бібліотеку для машинного навчання, що містить, наприклад, алгоритм k -найближчих сусідів, метод опорних векторів, наївний баєсівський класифікатор та випадковий ліс [37].

OpenCV спочатку була написана мовою програмування C++, після чого її інтерфейс був створений для мов C, Python, Java та MATLAB. Для використання з мовою C# потрібна обгортка бібліотеки, тобто бібліотека, що перетворює інтерфейс первинної бібліотеки на такий, що є або більш зручним, або сумісним для використання іншою мовою.

OpenCVSharp – одна з найпопулярніших обгортки OpenCV, написаних на C#. Вона побудована таким чином, що найбільше реплікує офіційний інтерфейс на C++. Перевагами OpenCVSharp є використання інтерфейсу C# IDisposable, що містить механізми для керування ресурсами, що більше не використовуються програмою, а також можливість конвертації об'єктів класу матриці Mat з OpenCV у об'єкти Bitmap та WriteableBitmap.

Причини використання мови програмування C# для програмної реалізації проєкту:

- код C# компілюється у проміжну мову (common intermediate language), що працює у середовищі віртуальної машини CLR (common language runtime), що дає можливість використовувати такі функції як автоматичне керування пам'яттю та обробку винятків.

- C# є мовою, що найповніше використовує можливості технології .NET. .NET є технологією, що дозволяє створювати кросплатформові програми та веб-застосунки.

- мова є повністю об'єктно-орієнтованою, тому має механізми успадкування, поліморфізму та інкапсуляції.

- синтаксис мови схожий на C++ та Java, що дозволяє імплементувати код, написаний на C#, у програми на C, C++ і Java, та навпаки, потребуючи мінімальну кількість змін.

- можливість використання системи керування пакетами NuGet. NuGet дозволяє значно прискорити та полегшити процес встановлення сторонніх бібліотек для .NET застосунків. Замість мануального пошуку, завантаження та прописування шляхів розташування бібліотек, NuGet дає можливість знайти та встановити велику кількість бібліотек, а також оновити їх у декілька кліків;

3.2 Особливості програмної реалізації

Початковим етапом роботи програми є виділення ознак еталонних зображень та підготовка даних до використання класифікаторами. Використовуючи клас BRISK з OpenCVSharp, розраховуються ключові точки та дескриптори для кожного еталонного зображення, що попередньо перетворюються на напівтонові. Вектори КТ та дескрипторів зменшуються до розміру 500 для пришвидшення проведення подальших експериментів та

стандартизації даних. Зменшення відбувається завдяки видаленню останніх $N - 500$ елементів.

Дескриптори, отримані функцією `DetectAndCompute` класу `BRISK` подаються у вигляді об'єкту класу `Mat` з `OpenCVSharp`, що є n -вимірним щільним масивом. Через відносну незручність використання, дескриптори конвертуються спочатку в список байтів, а потім у список бітів у вигляді `List<bool>`.

Для порівняння використовуються два допоміжних класи: перший містить у собі інформацію про дескриптор та клас (зображення, до якого він відноситься), другий – клас, що містить дескриптор вхідного зображення, найближчий дескриптор еталону (у вигляді об'єкту першого класу) у результаті порівняння та відстань між ними.

У кожному з методів порівняння дескрипторів відбувається завдяки функції `CompareDescriptors`, що порівнює кожен дескриптор вхідного зображення кожним дескриптором з необхідною відповідно до контексту базою дескрипторів за відстанню Геммінга, та зберігає результат у вигляді об'єкту другого допоміжного класу. Програмний вигляд функції вказаний у лістингу 3.1.

Лістинг 3.1 Програмна реалізація функції `CompareDescriptors`:

```
public List<DistanceData> CompareDescriptors(List<Data> data,
List<List<bool>> exampleDescriptors)
{
    List<DistanceData> result = new List<DistanceData>();
    for (int i = 0; i < exampleDescriptors.Count; ++i)
    {
        int minDistance = 1000;
        Data minData = new Data();
        foreach (Data dataPiece in data)
        {
```

```

        if (HammingDistance(exampleDescriptors[i],
dataPiece.GetDescriptor()) < minDistance)
        {
            minDistance = HammingDistance(exampleDescriptors[i],
dataPiece.GetDescriptor());
            minData = dataPiece;
        }
    }
    result.Add(new DistanceData(exampleDescriptors[i], minData,
minDistance));
}
return result;
}

```

Для методів з кластеризацією існує окрема модифікація, яка зберігає лише ті найближчі дескриптори, відстань для яких не перевищує певний поріг. Для обох методів цей поріг прийнято рівним 192.

Програма реалізує традиційний метод, метод зі спільною кластеризацією та метод з роздільною кластеризацією відповідно до їх опису у пункті 2.2. Для роботи цих методів у програмі реалізовані функції підрахунку голосів та визначення класу, порівняння вектору голосів за Манхеттенською відстанню, конвертації векторів голосів з розміру $N \cdot M$ до N . Також для зручності аналізу додані функції виведення у консоль голосів та Манхеттенських відстаней, бінарного вигляду дескрипторів та номеру класу, що є результатом класифікації

Для тестування методів розроблені функції, що виконують класифікацію зображення з різними геометричними перетвореннями та оцінюють правильність класифікації. Для повного тестування також розроблені функції, що реалізують попередню функцію для кожного еталонного зображення та зображення поза базою та виводять повну точність класифікації.

3.3 Інструкція користувача

Для використання розроблених методів та тестування потрібно створити консольний проект C# на .NET Framework 6.0 та під'єднати до нього бібліотеку OpenCVSharp. Для пришвидшення завантаження та під'єднання бібліотеки можна використовувати систему керування пакетами NuGet. Для цього натискаємо Project → Manage NuGet Packages, відкриваємо вкладку Browse та шукаємо “OpenCVSharp”. Далі завантажте пакети, вказані на рисунку 3.1.

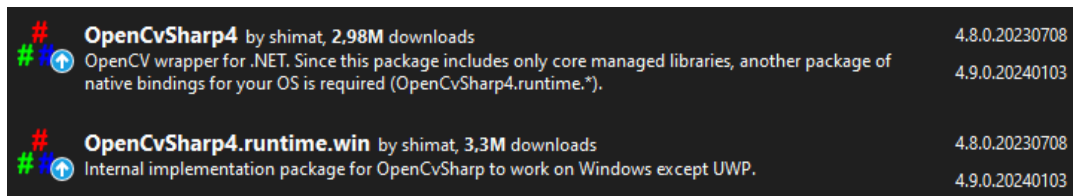


Рисунок 3.1 – Пакети, необхідні для роботи програми

У роботі використовувалась версія OpenCVSharp 4.8.0.20230708. Для керування версіями, у правій частині вікна NuGet, у дроббоксі Version оберіть версію 4.8.20230708 (рис. 3.2).

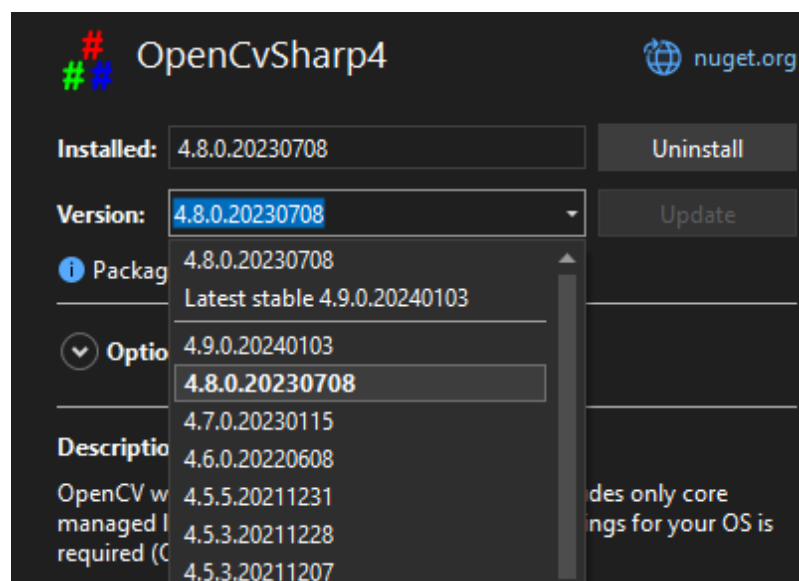


Рисунок 3.2 – Керування версіями у NuGet

Після встановлення OpenCVSharp, можна або підключити створену бібліотеку методів класифікації, або скопіювати створені класи у проєкт. Для підключення бібліотеки натисніть правою кнопкою по Solution → Add → Existing Project. Знайдіть та натисніть двічі на ClassificationMethods.csproj. Далі, натисніть правою кнопкою на Dependencies у створеному проєкті, Add Project Reference. Помітьте галочкою ClassificatonMethods, як показано на рисунку 3.3 та натисніть ОК.

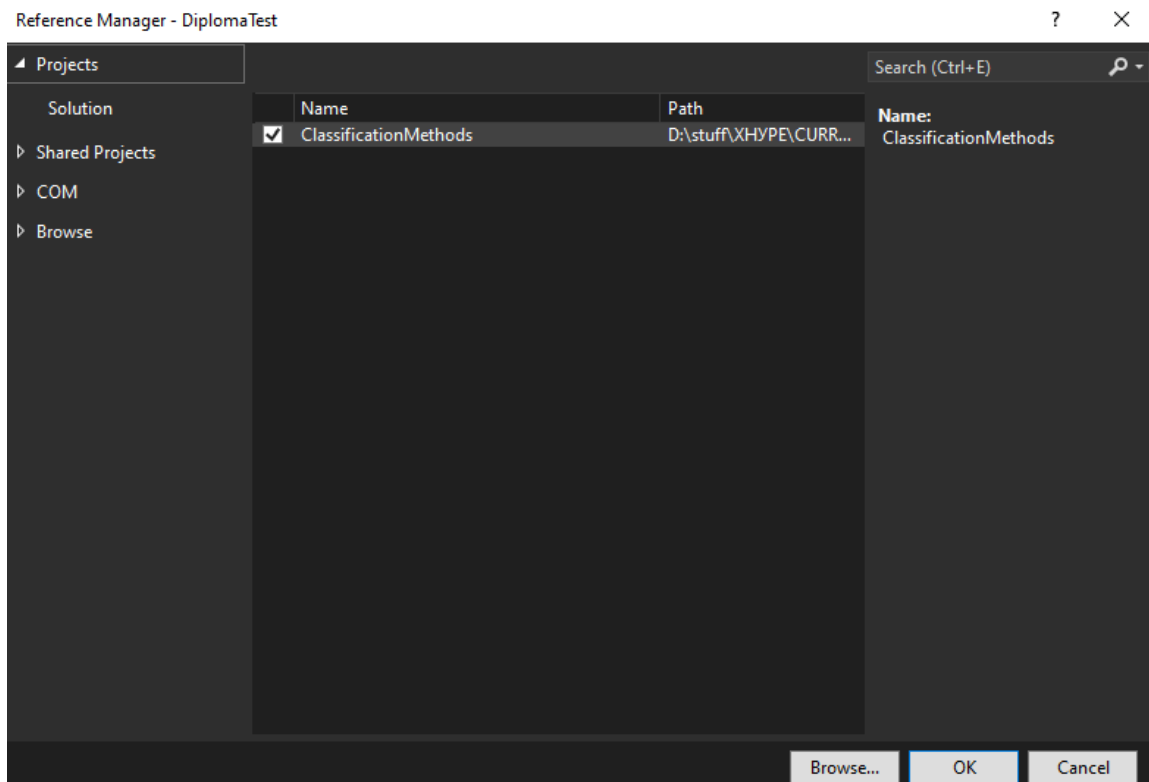


Рисунок 3.3 – Підключення бібліотеки розроблених класів до проєкту

Використовуючи директиви `using OpenCVSharp` та `using ClassificationMethods`, тепер є змога використовувати всі створені методи. Весь основний функціонал реалізований у класі `Solution`, об'єкт якого треба створити. Для створення бази дескрипторів еталонних зображень потрібно створити список виду `List<List<bool>>` та додавати дескриптори еталонних зображень так, як вказано на лістингу 3.2, вказуючи поріг BRISK та шлях до кожного зображення.

Лістинг 3.2 – Створення бази дескрипторів еталонних зображень

```

Solution solution = new Solution();
List<List<bool>> DescriptorPackage = new List<List<bool>>();

DescriptorPackage.AddRange(solution.GetDescriptors(@"D:\ e1.png",
123));
DescriptorPackage.AddRange(solution.GetDescriptors(@"D:\ e2.png",
156));
DescriptorPackage.AddRange(solution.GetDescriptors(@"D:\ e3.png", 56));
DescriptorPackage.AddRange(solution.GetDescriptors(@"D:\ e4.png", 67));
DescriptorPackage.AddRange(solution.GetDescriptors(@"D:\ e5.png",
90));

List<Data> data = solution.MakeData(5, 500, DescriptorPackage);

```

Функція `MakeData` у цьому випадку створить базу еталонних зображень у вигляді списку об'єктів додаткового класу `Data` з 5 класами, по 500 дескрипторів в кожному, зі списку дескрипторів `DescriptorPackage`. Для методів, що використовують кластеризацію також потрібно створити одновимірний масив цілих чисел, у якому вказані центри кластерів, які алгоритм k -середніх візьме як початкові. Значення центрів не повинні перевищувати 499.

Для тестування класифікації зображень розробленими методами можна або просто викликати функцію бажаного методу (`ClassicMethod`, `FullDBClusteringMethod` та `DistinctImageClusteringMethod`), використовуючи об'єкт класу `Solution`, або викликати функції тестування методів, що класифікують вхідне зображення з різними геометричними перетвореннями та показують точність класифікації. Такими функціями є `TestClassicMethod`, `TestFullDBClusteringMethod` та `TestDistinctImageClusteringMethod`. Приклад

вигляду класифікації одного зображення з використанням традиційного методу зображено на рисунку 3.4.

```
Classic method:
Votes:

Class 1: 370
Class 2: 38
Class 3: 32
Class 4: 38
Class 5: 22
Result class: 1
Time elapsed: 5982ms
```

Рисунок 3.4 – Вигляд класифікації зображення традиційним методом

3.4 Аналіз результатів тестування

Для тестування та аналізу розроблених методів були використані вісім картин української художниці Марії Примаченко [38]. Ці картини були розбиті на базу еталонних зображень, що складається з 5 зображень (рис. 3.5), а також 3 зображення поза базою (рис. 3.6).

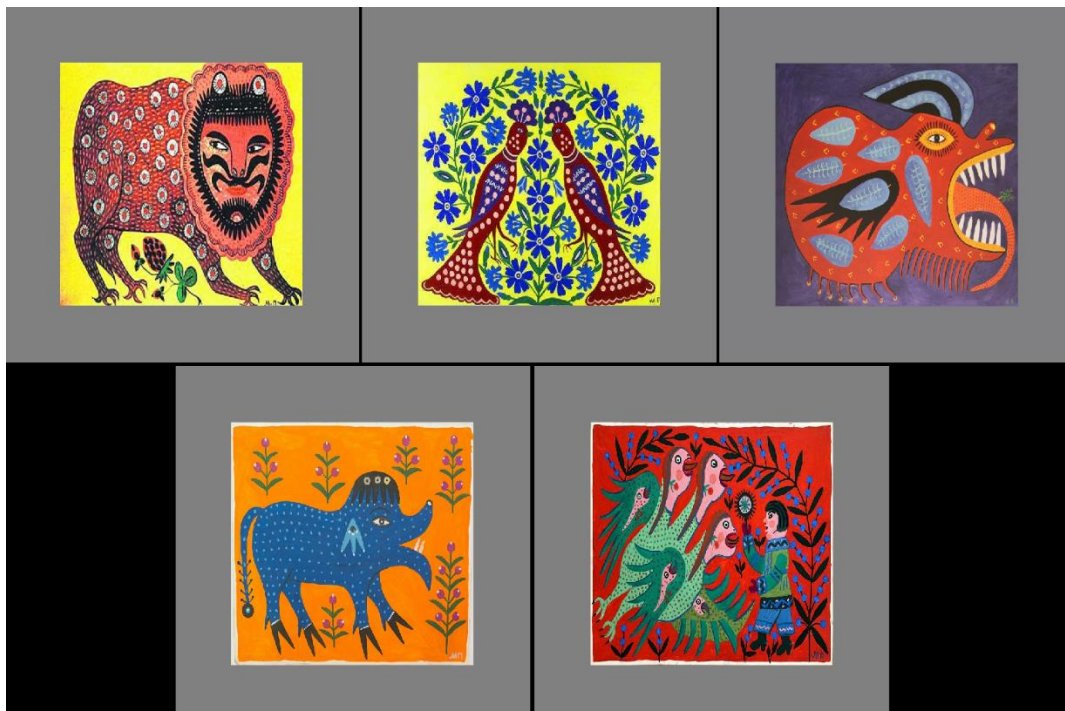


Рисунок 3.5 – База еталонних зображень



Рисунок 3.6 – Експериментальні зображення за межами бази

Кожне зображення представлено у форматі jpg та має розміри 512x512 пікселів. Самі оригінальні зображення зменшені до розмірів 350x350 та розміщені у центрі на сірому фоні розміром 512x512. Такий підхід має збільшити схожість отриманих ключових точок на етапі виявлення при повороті та масштабуванні відносно до вихідного зображення. Вигляд одного з еталонних зображень без та з КТ зображене на рисунку 3.7

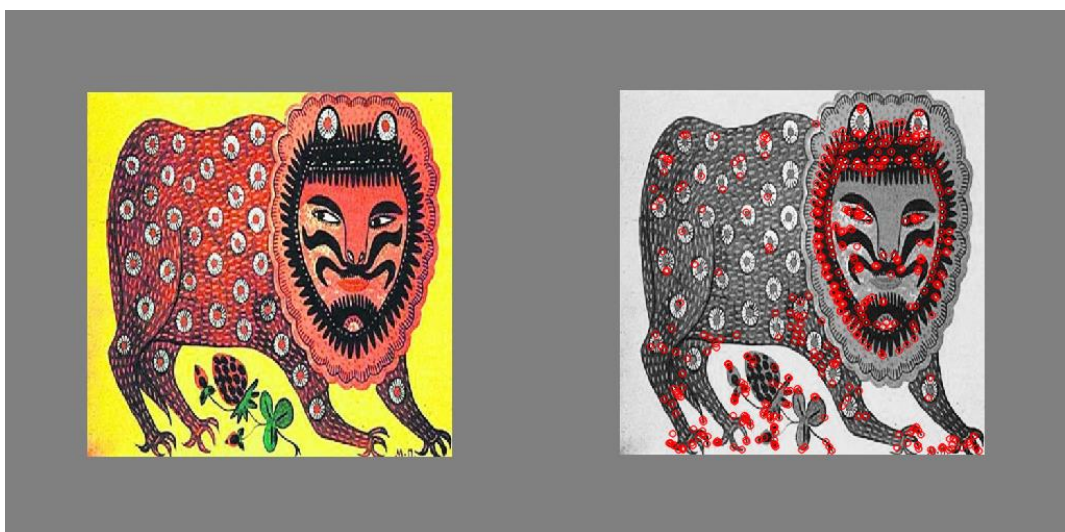


Рисунок 3.7 – Еталонне зображення та координати КТ

Зображення тестувались як у звичайному вигляді, так і з наступними геометричними перетвореннями: поворот (± 20 градусів), масштаб (0.9, 1.1) та зміщення (± 10 пікселів), тобто 8 комбінацій геометричних перетворень на кожне з зображень. Повна кількість зображень, що тестувались – 72. Вигляд зображень з застосованими геометричними перетвореннями та координатами КТ зображений на рисунку 3.8.



Рисунок 3.8 – Зображення з геометричними перетвореннями та координати КТ

Перед тестуванням методів експериментально визначені такі параметри, як поріг голосів для традиційного методу та методу з роздільною кластеризацією ($\delta_h=300$ та $\delta_h=140$), поріг максимально допустимої Манхеттенської відстані $\delta_\mu=180$ для методу зі спільною кластеризацією та поріг на максимальну відстань Геммінга для методів з кластеризацією $\delta_\rho=192$. Поріг на голоси та Манхеттенську відстань для методів визначався як мінімальний потрібний поріг для проходження усіх еталонних зображень $\pm 10\%$ від цього значення в залежності від результатів голосів або відстані зображень поза базою. Поріг на максимальну відстань Геммінга обирався

як 37,5% від максимально можливої відстані Геммінга між векторами ознак BRISK.

Традиційний метод класифікації продемонстрував себе у проведеному тестуванні як надзвичайно точний метод. З 72 зображень традиційний метод правильно класифікував кожне з еталонних зображень з великим відривом голосів та відмовив у класифікації кожному зображенню поза базою. У таблиці 3.1 наведені кількості голосів за кожен клас, отримані в результаті класифікації еталонного зображення, що у базі еталонів представлено класом 3. Жирним шрифтом виділено результат класифікації. Геометричні перетворення у лівій колонці йдуть у такому порядку: поворот, масштаб, зміщення.

Таблиця 3.1 – Результати класифікації еталонного зображення з номером 3 традиційним методом

перетв.\клас	1	2	3	4	5
(0,0,0)	0	0	500	0	0
(20,1.1,10)	27	19	417	24	13
(20,1.1,-10)	24	20	421	17	18
(20,0.9,10)	21	23	405	21	30
(20,0.9,-10)	29	27	383	29	32
(-20,1.1,10)	22	16	430	14	18
(-20,1.1,-10)	20	16	413	22	29
(-20,0.9,10)	32	30	385	26	27
(-20,0.9,-10)	39	27	371	20	43

Як видно із таблиці 3.1, жодне з поданих на класифікацію зображень не набрало менше порогових 300 голосів, у більшій кількості випадків навіть перевищуючи 400, що заявляє про велику впевненість класифікатора у обраному класі. Для порівняння у таблиці 3.2 наведено результат класифікації для зображення поза базою. Червоним кольором у таблиці позначені максимальні кількості голосів за класи зображення, якщо йому відмовлено у класифікації.

Таблиця 3.2 – Результати класифікації зображення поза базою традиційним методом

перетв.\клас	1	2	3	4	5
(0,0,0)	104	75	129	92	100
(20,1.1,10)	89	74	132	99	106
(20,1.1,-10)	77	78	137	89	119
(20,0.9,10)	115	68	140	80	97
(20,0.9,-10)	107	59	153	76	105
(-20,1.1,10)	94	75	156	81	94
(-20,1.1,-10)	97	78	146	90	89
(-20,0.9,10)	103	65	144	91	97
(-20,0.9,-10)	101	64	157	84	94

З таблиці 3.2 бачимо, що більшість зображень не набрали максимум голосів, що перевищує навіть половину необхідного порогу δ_h для отримання мітки класу.

Тестування для обох варіантів методів, що використовують кластеризацію, відбувалося на параметрі кількості кластерів M рівному 3 (15 кластерів для методу зі спільною кластеризацією) та 5 (25 кластерів для методу зі спільною кластеризацією). При цьому у якості початкових центрів кластерів для $M = 3$ обирались кожний 200, 300 та 400 дескриптор кожного еталонного зображення, тоді як для $M = 5$ – дескриптори еталонів з номерами 50, 150, 250, 350, 450. Для спрощення тестування алгоритм k -середніх виконує лише 5 ітерацій.

Похибка кластеризації розраховувалась у кожному методі. Значення похибки кластеризації для $M = 3$ та $M = 5$ для обох методів зазначені у таблиці 3.3.

Таблиця 3.3 – Похибки кластеризації для різних методів

Спільний, $M = 3$	0,305				
Спільний, $M = 5$	0,286				
Роздільний, $M = 3$	0,331	0,34	0,348	0,33	0,333
Роздільний, $M = 5$	0,315	0,311	0,324	0,317	0,311

За результатом моделювання метод із сумісною кластеризацією для бази зображень при $M = 3$ показав відносно високий результат точності $pr = 0,79$, правильно класифікувавши кожне з еталонних зображень та 12 з 27 зображень поза базою. Очікувано, що зі збільшенням кількості кластерів до п'яти точність класифікатора зростає. Отримане значення точності pr з кількістю кластерів $M = 5$ стало дорівнювати 0,88. Ідентично до тесту з $M = 3$, із жодним із еталонних зображень не виявлено помилок, але відбулось покращення стосовно відмови у класифікації зображень поза базою: класифікатор відхилив 18 з 27 зображень, що не належали до бази еталонів. Класифікація двох еталонів методом із сумісною кластеризацією наведена в таблицях 3.4 та 3.5.

Таблиця 3.4 – Результати класифікації еталонного зображення 2 методом зі спільною кластеризацією, $M = 5$

перетв.\клас	1	2	3	4	5
(0,0,0)	339	0	509	381	428
(20,1.1,10)	313	146	465	295	406
(20,1.1,-10)	320	157	478	304	413
(20,0.9,10)	294	161	456	308	387
(20,0.9,-10)	297	156	453	299	396
(-20,1.1,10)	316	119	468	306	419
(-20,1.1,-10)	313	120	489	311	424
(-20,0.9,10)	310	145	484	298	435
(-20,0.9,-10)	302	163	462	312	405

Таблиця 3.5 – Результати класифікації еталонного зображення 2 методом зі спільною кластеризацією, $M = 3$

перетв.\клас	1	2	3	4	5
(0,0,0)	216	0	267	307	254
(20,1.1,10)	198	138	269	257	306
(20,1.1,-10)	194	130	253	255	312
(20,0.9,10)	178	110	233	241	286
(20,0.9,-10)	180	128	259	235	300
(-20,1.1,10)	192	102	237	261	300
(-20,1.1,-10)	189	119	268	256	319
(-20,0.9,10)	196	144	281	227	328
(-20,0.9,-10)	206	138	283	229	336

Таблиці 3.4 та 3.5 показують значення мінімальної Манхеттенської відстані, що в середньому є відносно далекими від порогу. Значення відстані для $M = 5$ в середньому є більшими за варіант $M = 3$, що в теорії повинно показувати меншу впевненість у класифікації, але відстані до найдалшого класу значно зросли зі збільшенням кількості кластерів, що свідчить про кращу здатність розрізнення класів. Для порівняння, результати класифікації зображення поза базою наведені у таблиці 3.6. Червоним кольором у таблиці позначені значення мінімальної Манхеттенської відстані для зображення, якщо йому відмовлено в класифікації.

Таблиця 3.6 – Результати класифікації зображення поза базою 3 методом зі спільною кластеризацією, $M = 5$

перетв.\клас	1	2	3	4	5
(0,0,0)	283	342	361	357	270
(20,1.1,10)	263	368	363	383	264
(20,1.1,-10)	263	372	353	373	274
(20,0.9,10)	223	298	357	323	314
(20,0.9,-10)	232	301	346	316	311
(-20,1.1,10)	264	353	360	342	245
(-20,1.1,-10)	287	344	369	361	242
(-20,0.9,10)	216	281	344	296	321
(-20,0.9,-10)	206	285	316	274	327

З таблиці 3.6 можна зробити висновок, що через те, що деякі з відстаней знаходяться на відносно недалекій відстані ± 25 від порогу, іноді можуть відбуватись ситуації відкидання еталонного зображення, або, як виявилось найчастіше, невідмова у класифікації зображенню поза базою. Оскільки усі еталонні зображення під час тестування завжди відносились до правильного класу, неточність методу полягає у достатньо великій чутливості методу до зміни порогу на Манхеттенську відстань δ_{μ} .

Метод з роздільною кластеризацією зображень під час тестування продемонстрував більш суттєву точність класифікації. При кількості кластерів $M = 3$, точність класифікації *pr* методу з роздільною класифікацією

склала 0,89, а для $M = 5 - 0,98$. При $M = 3$, окрім помилки невідмови у класифікації 7 з 27 зображень поза базою, була допущена помилка при відмові у класифікації 2 з 45 еталонних зображень з геометричними перетвореннями. Проте слід зазначити, що відмова для цих еталонних зображень сталась через нестачу одного голосу у першому випадку та п'яти у іншому (табл. 3.7), тобто ці випадки можна вважати винятковими.

Таблиця 3.7 – Результати класифікації еталонного зображення 3 методом з роздільною кластеризацією, $M = 3$

перетв.\клас	1	2	3	4	5
(0,0,0)	55	38	185	46	94
(20,1.1,10)	59	42	169	58	92
(20,1.1,-10)	63	46	158	58	95
(20,0.9,10)	59	49	148	58	95
(20,0.9,-10)	57	54	142	60	98
(-20,1.1,10)	61	46	147	59	103
(-20,1.1,-10)	55	46	148	57	111
(-20,0.9,10)	57	57	135	53	98
(-20,0.9,-10)	60	52	139	58	98

Як можна побачити, ще декілька зображень пройшли зі значенням максимальної кількості голосів за клас дуже близьким до порогового значення $\delta_h=140$. Інші ж еталонні зображення під час класифікації в більшості показували результат, що перевищує 150, іноді отримуючи результати більше за 200. Можливим рішенням неправильної класифікації еталонного зображення 3 є обрання інших центрів кластерів, оскільки ще більше заниження порогу голосів може призвести до зменшення точності через те, що деякі з зображень поза базою будуть отримувати мітку класу у ще більшій кількості за ту, що була отримана у результаті тестування, навіть для $M = 5$.

При розбитті бази дескрипторів еталонних зображень на 5 кластерів під час класифікації 72 зображень метод класифікації з роздільною кластеризацією помилився при класифікації лише одного зображення, тобто продемонстрував майже ідеальний результат.

Бачимо із таблиці 3.8, значення максимальної кількості голосів за п'яти кластерів є значно вищим за максимальні кількості голосів з розбиттям описів на 3 кластери. Жодне зі значень не є меншим за 180, що вже достатньо помітно перевищує поріг для отримання класу, а деякі з них взагалі перевищують 200.

Таблиця 3.8 – Результати класифікації еталонного зображення 3 методом з роздільною кластеризацією, $M = 5$

перетв.\клас	1	2	3	4	5
(0,0,0)	66	57	219	53	70
(20,1.1,10)	69	55	200	58	79
(20,1.1,-10)	64	60	200	60	75
(20,0.9,10)	65	48	182	72	84
(20,0.9,-10)	66	55	188	60	80
(-20,1.1,10)	65	50	181	63	92
(-20,1.1,-10)	59	59	185	63	87
(-20,0.9,10)	65	56	190	56	83
(-20,0.9,-10)	66	47	199	56	85

Помилка сталася з одним з зображень поза базою, де класифікатор не відмовив одному з зображень з геометричними перетвореннями. Максимум голосів цього зображення при цьому перевищив поріг δ_h всього на 4 голоси. Результат класифікації цього зображення з його перетвореннями зазначений у таблиці 3.9.

Таблиця 3.9 – Результати класифікації зображення поза базою методом з роздільною кластеризацією, $M = 5$

перетв.\клас	1	2	3	4	5
(0,0,0)	83	120	62	113	79
(20,1.1,10)	90	113	65	128	72
(20,1.1,-10)	108	111	59	126	63
(20,0.9,10)	100	108	69	114	82
(20,0.9,-10)	94	112	71	113	80
(-20,1.1,10)	80	107	56	144	78
(-20,1.1,-10)	93	105	59	124	83
(-20,0.9,10)	98	108	63	124	67
(-20,0.9,-10)	98	110	68	125	65

З таблиці 3.9 можна спостерігати, що навіть для такого «проблематичного» зображення більшість значень максимальної кількості голосів знаходиться у діапазоні, що не перевищує 130.

Під час класифікації, була заміряна швидкодія виконання класифікації кожного зображення кожним методом. Класифікація виконувалась на комп'ютері з процесором Intel Core I5-7600K (3.8GHz) та 16 Гб оперативної пам'яті. Зафіксований час виконання алгоритмів продемонстрований на гістограмі, зображеній на рисунку 3.9.

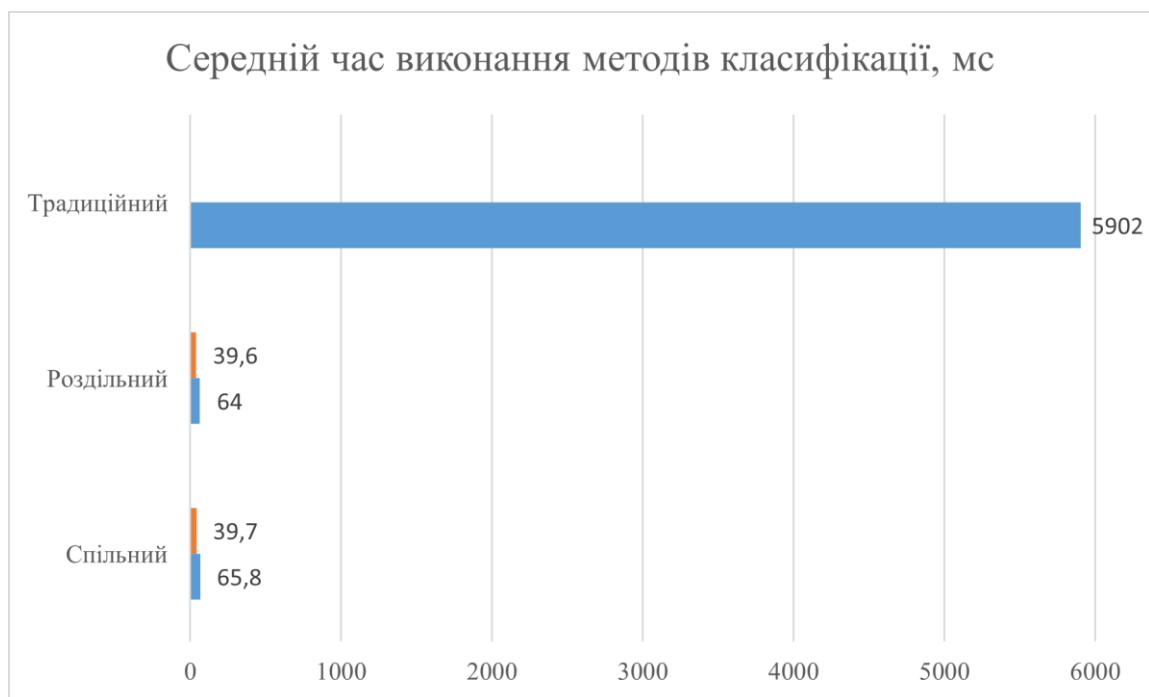


Рисунок 3.9 – Гістограма середнього часу виконання алгоритмів

Середній час класифікації для традиційного методу склав 5902 мс. Для методу зі спільною кластеризацією при $M = 3$ час склав 39,7 мс, а для $M = 5$ він дорівнював 65,8 мс. Метод з роздільною кластеризацією показав швидкість 39,6 мс при кількості кластерів $M = 3$ та 64 мс при $M = 5$. Показники середнього часу виконання методів класифікації зображень в результаті проведених експериментів вказані у таблиці 3.10.

Таблиця 3.10 – Середній час виконання кожного методу класифікації зображень

Метод	Час виконання, мс
Традиційний	5902
Спільний, $M = 3$	39,7
Спільний, $M = 5$	65,8
Роздільний, $M = 3$	39,6
Роздільний, $M = 5$	64

Значення точності алгоритмів класифікації для різного числа кластерів продемонстровані на гістограмі рисунку 3.10.

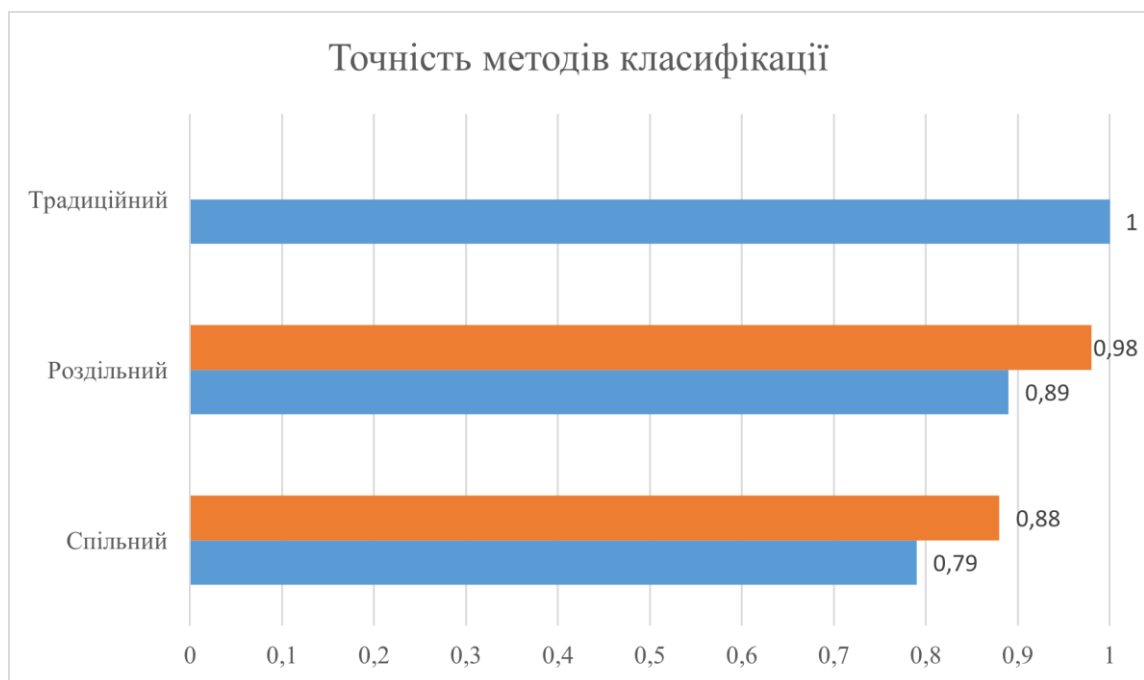


Рисунок 3.10 – Гістограма значень точності методів

У результаті проведених експериментів традиційний метод виявився найточнішим з точністю $pr = 1$, оскільки метод не апроксимує жодних даних та працює з ними напряму, але ціною помітного зниження швидкодії у порівнянні з розробленими модифікаціями.

Метод зі спільною кластеризацією виявився найменш точним зі значеннями $pr = 0,79$ та $pr = 0,88$. Метод з роздільною кластеризацією показав найвищу точність класифікації серед методів, що використовують кластеризацію. Навіть з трьома кластерами його точність більша за точність методу зі спільною кластеризацією при п'яти кластерах. Враховуючи відносно незначні втрати у точності методів з кластеризацією, у порівнянні з традиційним методом швидкість є значно більшою: у 149 разів для трьох кластерів та 92 рази для п'яти. Зміна кількості кластерів з трьох до п'яти достатньо помітно підвищує точність методів, витрачаючи приблизно на 25 мс більше на виконання класифікації. Розрахунки критерію відстані між центрами кластерів пар еталонних зображень β для першого та другого еталонного зображення при $M = 3$ та $M = 5$ показали результати 168,2 та 182,8, отже підвищення кількості кластерів збільшує відстань між класами, що сприяє результативності класифікації. Отримані значення точності за результатом моделювання кожного з методів наведені в таблиці 3.11.

Таблиця 3.11 – Точність методів класифікації зображень

Метод	Точність
Традиційний	1
Спільний, $M = 3$	0,79
Спільний, $M = 5$	0,88
Роздільний, $M = 3$	0,89
Роздільний, $M = 5$	0,98

Підвищення точності методів при збільшенні кількості кластерів відбувається через покращення якості апроксимованих даних: за більшої кількості кластерів створюється більше унікальних описів кожного зображення, що дозволяє краще їх розрізняти на етапі порівняння (про що свідчить зростання значення критерію β). Втрати швидкодії при збільшенні кількості кластерів пов'язані зі збільшенням кількості даних, які

порівнюються, що призводить до збільшення кількості ітерацій як безпосередньо на етапі порівняння, так і на етапі кластеризації.

Слід зазначити також, що отримана у експерименті оцінка точності класифікації на пряму залежить від використаного набору еталонних зображень та зображень поза базою. При застосуванні іншої бази зображень оцінка для значення точності може як зменшуватись, так і збільшуватись.

ВИСНОВКИ

У рамках кваліфікаційної роботи розроблені, реалізовані та досліджені два швидкодіючих та ефективних методи класифікації зображень з використанням кластерного подання для опису зображень.

Розроблені методи експериментально порівняні між собою та із показниками традиційного структурного методу класифікації зображень, що має за основу лінійний пошук. Виходячи з результатів, метод з роздільною кластеризацією має найкращі показники точності та швидкодії серед методів, що використовують кластерне подання. Обидва розроблені методи забезпечують суттєвий приріст швидкодії у порівнянні з традиційним методом класифікації та є майже рівноцінними у швидкодії у порівнянні одне з одним.

Розроблені методи можуть бути універсально застосовані для будь-яких баз зображень довільного розміру. При використанні методів треба звернути увагу на визначення значень порогів, що відповідають за відмову у класифікації зображенням поза базою для забезпечення максимальної точності класифікації.

Перспективами подальшого розглядання теми є практичне застосування методів на об'ємних наборах зображень з метою додаткового визначення параметрів ефективності та швидкодії.

Результати роботи апробовано у формі поданої до публікації статті [30] та тез доповідей під час Міжнародного молодіжного форуму [41].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is computer vision? URL: <https://www.ibm.com/topics/computer-vision> (дата звернення 11.04.2024).
2. Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*.
3. Chen, L., Li, S., Bai, Q., Yang, J., Jiang, S., & Miao, Y. (2021). Review of image classification algorithms based on convolutional neural networks. *Remote Sensing*, 13(22), 4712.
4. Kumar, G., & Bhatia, P. K. (2014, February). A detailed review of feature extraction in image processing systems. In *2014 Fourth international conference on advanced computing & communication technologies* (pp. 5-12). IEEE.
5. Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011, November). BRISK: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision* (pp. 2548-2555). Ieee.
6. Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9* (pp. 430-443). Springer Berlin Heidelberg.
7. Ghazal, T. M. (2021). Performances of k-means clustering algorithm with different distance metrics. *Intelligent Automation & Soft Computing*, 30(2), 735-742.
8. Гороховатський, В. О., & Гадецька, С. В. (2020). Статистичне оброблення та аналіз даних у структурних методах класифікації зображень.
9. Гороховатський В.О., Гадецька С.В., Стяглик Н.І., Власенко Н.В. (2020) Класифікація зображень на підставі ансамблю статистичних розподілів за класами еталонів для компонентів структурного опису. *Радіоелектроніка, інформатика, управління*, №4 , с. 85–94.
10. Gorokhovatsky, V.O. and Gadetska, S.V. (2019) Determination of Relevance of Visual Object Images by Application of Statistical Analysis of

Regarding Fragment Representation of their Descriptions, *Telecommunications and Radio Engineering*, 78 (3), pp. 211–220.

11. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, 33 (1), 113-125.

12. Gorokhovatskyi, V., Gadetska, S., & Stiahlyk, N. (2023). Accelerating Image Classification based on a Model for Estimating Descriptor-to-Class Distance. *International Journal of Computing*, 22(4), 485-492.

13. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, 11, 126938-126949.

14. Gorokhovatsky, V. (2014), *Structural Analysis and Intellectual Data Processing in Computer Vision*, SMIT, Kharkiv.

15. Gadetska, S.V., Gorokhovatsky, V.O. Statistical Measures for Computation of the Image Relevance of Visual Objects in the Structural Image Classification Methods. *Telecommunications and Radio Engineering*. – 2018, Vol. 77 (12), pp. 1041–1053.

16. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *Int. scientific symp. «Intelligent Solutions-S»*. Computational intelligence. Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine, 25-27.

17. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9), pp. 64-72.

18. Gadetska, S.V., Gorokhovatskyi, V. O., Stiahlyk, N. I., Vlasenko, N.V. Statistical data analysis tools in image classification methods based on the description as a set of binary descriptors of key points. *Radio Electronics, Computer Science, Control*, 2021, №4, pp. 58-68.

19. Gorokhovatskyi, V., Vlasenko, N. (2021). Редукція опису зображення у складі множини дескрипторів на основі метричного критерію інформативності. *Advanced Information Systems*, 5(4), pp. 10-16.
20. Гороховатський В., Творошенко І., Сидоренко Д. (2021) Класифікація зображень із використанням кластерного подання, Міжн. наук. симпозиум «Інтелектуальні рішення-С». *Обчислювальний інтелект. Теорія прийняття рішень* (Вересень 29, 2021). Київ – Ужгород, С. 44-45.
21. Gorokhovatskyi, O., Peredrii, O., Gorokhovatskyi, V., Vlasenko, N. (2023) Explanation of CNN Image Classifiers with Hiding Parts. In: J. Benois-Pineau, R. Bourqui, D. Petkovic, G. Quenot (eds), *Explainable Deep Learning Artificial Intelligence*, pp. 125-146, Academic Press, 346 p.
22. Tvoroshenko, I., & Zarivchatskyi, R. (2020). Analysis of existing methods for searching object in the video stream, in *Proc. VI Int. Sci. Practic. Conf. «About the problems of science and practice, tasks and ways to solve them»*, Milan, pp. 500-505.
23. Gorokhovatskyi V.A., Zamula A.A. (2016) Employment of Intelligent Technologies in Multiparametric Control Systems. *Telecommunications and Radio Engineering*. Vol. 75, No 19, p. 1775–1785.
24. V. Gorokhovatsky, Y. Putyatin and V. Stolyarov (2017) Research of Effectiveness of Structural Image Classification Methods using Cluster Data Model, *Radio Electronics Computer Science Control*, vol. 3, no. 42, pp. 78-85.
25. Gorokhovatsky V.A., Putyatin Ye.P. (2009) Image Likelihood Measures of the Basis of the Set of Conformities. *Telecommunications and Radio Engineering*, 68 (9), 763-778.
26. Gorokhovatsky, V.A., Putyatin, Y.P. (2008) Structural recognition of images on the basis of voting models of attributes of typical points, *Data recording, storage and processing*, 10(4), 75-85.
27. V. A. Gorokhovatskiy, (2011), Compression of descriptions in the structural image recognition, *Telecommunications and Radio Engineering*, vol. 70, no. 15, pp. 1363–1371, doi: 10.1615/TelecomRadEng.v70.i15.60.

28. Гороховатський В.О., Пупченко Д.В., Солодченко К.Г. (2018) Аналіз властивостей, характеристик та результатів застосування новітніх детекторів для визначення особливих точок зображення. Системи управління, навігації та зв'язку, 93–98.

29. Gorokhovatskyi V., Gadetska S., Stiahlyk N. (2020) Image structural classification technologies based on statistical analysis of descriptions in the form of bit descriptor set. In CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2020), 2608, pp. 1027-1039.

30. Gorokhovatskyi, V., Stiahlyk, N., Mazur, Y., Vechirska, A. (2024) Способи метричної грануляції для опису зображень у задачі класифікації. Системи управління, навігації та зв'язку, (подано до публікації)

31. Tvoroshenko I.S., and Gorokhovatsky V.O. (2019) Intelligent classification of biophysical system states using fuzzy interval logic, Telecommunications and Radio Engineering, 78(14), pp. 1303–1315.

32. Gorokhovatsky V.A. Efficient Estimation of Visual Object Relevance during Recognition through their Vector Descriptions. Telecommunications and Radio Engineering. – 2016, Vol. 75, No 14. – P. 1271–1283.

33. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, International Journal of Academic and Applied Research, 7(9), 57-70.

34. Gadetska S., Gorokhovatskyi V., Stiahlyk N., Vlasenko N. (2022) Aggregate Parametric Representation of Image Structural Description in Statistical Classification Methods. In CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2022), 3137, pp. 68-77.

35. Gorokhovatskyi V.A. (2018) Image Classification Methods in the Space of Descriptions in the Form of a Set of the Key Point Descriptors. Telecommunications and Radio Engineering, 77 (9), pp. 787-797.

36. OpenCV wrapper for .NET. URL: <https://github.com/shimat/opencvsharp> (дата звернення 16.04.2024).

37. Документація OpenCV. URL: <https://docs.opencv.org/> (дата звернення 17.04.2024).
38. WikiArt. Енциклопедія візуальних мистецтв. <https://www.wikiart.org/uk/mariya-primachenko> (дата звернення 12.04.2024).
39. K-means: a complete introduction. URL: <https://towardsdatascience.com/k-means-a-complete-introduction-1702af9cd8c> (дата звернення 25.04.2024).
40. Ibrahim, D. Y., Gorokhovatskyi, V., Tvoroshenko, I., & Zeghid, M. (2022). Cluster representation of the structural description of images for effective classification.
41. Мазур Є.В. Класифікація зображень на підставі кластерування. Радіoeлектроніка та молодь у ХХІ столітті: тези доповідей 28-го Міжнародного молодіжного форуму (Харків, 16–18 квітня 2024 р.). Харків: ХНУРЕ, 2024. Т. 7. С. 80-81.
42. Gorokhovatskyi O., Gorokhovatskyi V., Peredrii O. Analysis of Application of Cluster Descriptions in Space of Characteristic Image Features. *Data*. 2018. Vol. 3(4), 52. DOI: <https://doi.org/10.3390/data3040052>
43. Gorokhovatskyi, V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, *IEEE Access*, vol. 12, pp. 73376-73385.
44. Гороховатський В.О., Пономаренко Р.П. (2020) Класифікація зображень на підставі формування незалежної системи кластерів у складі структурних описів бази еталонів. *Сучасні інформаційні системи*, 4, С. 17-23.
45. Daradkeh, Y.I., Tvoroshenko, I., Gorokhovatskyi, V., Latiff, L.A., and Ahmad, N. (2021) Development of Effective Methods for Structural Image Recognition Using the Principles of Data Granulation and Apparatus of Fuzzy Logic, *IEEE Access*, 9, pp. 13417-13428.