

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)

РОЗРОБЛЕННЯ МЕТОДУ КЛАСИФІКАЦІЇ НА ПІДСТАВІ
ТРАНСФОРМАЦІЇ СТРУКТУРНОГО ОПИСУ ЗОБРАЖЕНЬ
(тема)

Виконав:
студент 4 курсу, групи ІТІНФ-20-3

Вечірська А.Д.
(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 Вечірській Анні Данилівні
(прізвище, ім'я, по батькові)1. Тема роботи Розроблення методу класифікації на підставі трансформації структурного опису зображень

затверджена наказом університету від 20 травня 2024 року № 464 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 24 травня 2024 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, бібліотека комп'ютерного зору з відкритим кодом OpenCV, мова програмування Python, редактор коду Microsoft Visual Studio Code.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Огляд методів і засобів класифікації зображень.

2. Математична модель структурних методів класифікації.

3. Моделі трансформації структурного опису зображень.

4. Програмна реалізація і оцінювання ефективності трансформацій описів у структурних методах класифікації.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми, постановка задачі, методи класифікації з використанням структурного опису, моделі грануляції описів, тестові зображення, результати програмного моделювання.

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 | 08.04.24-15.04.24 | |
| 3 | Аналіз літератури з досліджуваної проблеми | 16.04.24-18.04.24 | |
| 4 | Аналіз технічних засобів | 19.04.24-23.04.24 | |
| 5 | Розробка методу | 24.04.24-11.05.24 | |
| 6 | Програмна реалізація | 12.05.24-20.05.24 | |
| 7 | Оформлення пояснювальної записки | 21.05.24-23.05.24 | |
| 8 | Перевірка на плагіат | 26.05.24 | |
| 9 | Рецензування | 27.05.24 | |
| 10 | Підготовка презентації та доповіді | 28.05.24-02.06.24 | |
| 11 | Занесення роботи в електронний архів | 03.06.24 | |
| 12 | Попередній захист кваліфікаційної роботи | 03.06.24 | |

Дата видачі завдання 8 квітня 2024 р.

Студент _____
(підпис)

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

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 68 с., 14 табл., 23 рис., 51 джерело.

КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, СТРУКТУРНИЙ МЕТОД, РЕЛЕВАНТНІСТЬ ОПИСІВ, ТРАНСФОРМАЦІЯ ОПИСУ, ДЕСКРИПТОРИ, ГРАНУЛЮВАННЯ.

Об'єктом роботи є методи класифікації зображень з використанням структурного опису у формі множини дескрипторів ключових точок.

Метою роботи є розроблення швидкісних методів класифікації на підставі трансформації структурного опису шляхом грануляції даних, аналіз їх точності та швидкодії.

Програмно реалізовано модифіковані методи класифікації з використанням грануляції даних, у порівняльному аспекті оцінено їх ефективність з традиційним методом.

Результативність методів з грануляцією оцінено для бази зображень з використанням геометричних перетворень та зображень, що не відносяться до жодного класу.

Результати моделювання показали зростання швидкодії класифікації пропорційно ступеню скорочення опису без зниження показника точності.

IMAGE CLASSIFICATION, STRUCTURAL METHOD, RELEVANCE OF DESCRIPTIONS, TRANSFORMATION OF DESCRIPTIONS, DESCRIPTORS, GRANULATION.

The object of the work is image classification methods using a structural description in the form of a set of descriptors of keypoints.

The aim of the work is the development of high-speed classification methods based on the transformation of the structural description by data granulation and the analysis of their accuracy and speed.

Modified classification methods using data granulation were implemented in software, and their efficiency was evaluated in comparison with the traditional method.

The performance of methods with granulation was evaluated on a dataset of images using geometric transformations and images that do not belong to any class.

The simulation results showed an increase in the speed of classification proportional to the degree of reduction of the description without a decrease in the accuracy rate.

ЗМІСТ

| | |
|--|----|
| Перелік умовних позначень, символів, одиниць, скорочень і термінів | 6 |
| Вступ | 7 |
| 1 Огляд методів і засобів розпізнавання зображень | 9 |
| 1.1 Прикладні задачі розпізнавання зображень | 9 |
| 1.2 Апарат ключових точок та дескриптори | 12 |
| 1.3 Постановка задачі | 19 |
| 2 Класифікація з використанням структурного опису зображення | 20 |
| 2.1 Етапи і моделі аналізу даних у структурному методі класифікації .. | 20 |
| 2.2 Оцінка ефективності методу класифікації | 22 |
| 2.3 Створення бази зображень для тестування | 25 |
| 2.4 Структурні методи з використанням медоїду | 28 |
| 2.5 Застосування методів грануляції даних | 31 |
| 3 Програмне моделювання методів класифікації зображень | 35 |
| 3.1 Обґрунтування вибору середовища програмної реалізації | 35 |
| 3.2 Особливості програмної реалізації | 38 |
| 3.3 Результати класифікації із застосуванням трансформації структурного опису та їх аналіз | 54 |
| Висновки | 62 |
| Перелік джерел посилання | 63 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

КТ – ключова точка

SIFT – Scale Invariant Feature Transform (інваріантна до масштабування трансформація особливостей)

SURF – Speeded up Robust Features (прискорені стійкі особливості)

BRISK – Binary Robust Invariant Scalable Keypoints (бінарні стійкі інваріантні масштабовані ключові точки)

ORB – Oriented FAST and Rotated BRIEF (орієнтований FAST та BRIEF з врахуванням обертання)

FAST – Features from Accelerated Segment Test (особливості на основі швидкого сегментного тесту)

BRIEF – Binary Robust Independent Elementary Features (бінарні стійкі незалежні елементарні особливості)

ВСТУП

Комп'ютерний зір – обширна галузь, що включає в себе різноманітні технології, спрямовані на те, щоб дозволити комп'ютерам працювати з візуальною інформацією, аналізувати, інтерпретувати зображення та відео. Це міждисциплінарна область, що стоїть на перетині інформатики та штучного інтелекту, і постійно розвивається.

З розвитком обчислювальної техніки, технології комп'ютерного зору стали застосовуватися в багатьох сферах, в тому числі в таких як медицина, сільське господарство, автомобільна промисловість, безпека, охорона дикої природи [1–5].

Одним з ключових напрямків комп'ютерного зору є розпізнавання зображень, що включає класифікацію.

Класифікація зображень може бути виконана багатьма методами. Однією категорією з них є методи на основі структурного опису. Цей підхід використовує описи зображень у вигляді числових векторів, які описують певні властивості фрагментів об'єкта. Вони є інваріантними до геометричних перетворень. Класифікація відбувається за рахунок обчислення відстаней чи подібності між множинами векторів описів зображень та визначення схожості на основі цих обчислень, зазвичай, використовуючи голосування [2, 3].

Із впровадженням сучасних технологій збільшується обсяг інформації, що має бути аналізована, час процесу обробки інформації має бути зведений до мінімального. Традиційні методи, що працюють із структурним описом, при великій кількості даних вимагають значного обсягу обчислень. Тому ці методи мають необхідність в прискоренні для запобігання значних часових витрат та ресурсів на процес класифікації [4–8].

Трансформація структурного опису, а саме зменшення кількості дескрипторів, є варіантом збільшення швидкодії методів на основі структурного опису. За рахунок скорочення опису зменшується потрібний

обсяг обчислень. Зазвичай для забезпечення достатньої точності необхідна достатньо велика кількість дескрипторів. Проте, залишаючи для подальшого процесу класифікації найбільш інформативні ознаки, що найкраще характеризуватимуть структуру і зміст конкретного зображення, такий підхід може дозволити зберегти баланс між швидкістю методу класифікації та точністю вихідних результатів.

Актуальність кваліфікаційної роботи полягає у нагальній необхідності розроблення засобів прискорення при аналізі даних у системах класифікації зображень. Через постійне зростання обсягів багатовимірних даних, кількості цифрових зображень з однієї сторони, та збільшення вимог до точності та швидкодії оброблення та класифікації зображень, сучасні методи розпізнавання зображень, включаючи методи на підставі структурного опису, мають великий потенціал для подальшого вдосконалення [3, 4].

1 ОГЛЯД МЕТОДІВ І ЗАСОБІВ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ

1.1 Прикладні задачі розпізнавання зображень

Розпізнавання зображень передбачає знаходження та ідентифікацію конкретних об'єктів на зображеннях. Це одне з основних завдань комп'ютерного зору.

Сьогодні розроблено статистичне, структурне та синтаксичне розпізнавання образів. Це дуже важливі напрями, що мають багато сфер практичного застосування, такі як обробка відео, розпізнавання облич, розпізнавання рукописів, класифікація [6].

У статистичному розпізнаванні образів дані найчастіше представлені як фіксовані числові вектори ознак. Вони збираються відповідно до завдання, і далі аналізуються за допомогою певних методів, та класифікуються. Після цього зазвичай оцінюють результати.

У синтаксичному та структурному об'єкти представляють за допомогою символічних структур, таких як графи. Таким чином, зберігаються не тільки ознаки, а й взаємозв'язки між ними [6].

Структурні та синтаксичні методи враховують контекст, на відміну від статистичних. Проте статистичні методи є ефективнішими у використанні ресурсів, оскільки працюють з числовими значеннями, та є простішими, їх легше інтерпретувати [6].

До розпізнавання зображень відносяться такі завдання як сегментація зображень, виявлення об'єктів, розуміння сцени, класифікація зображень [4].

Сегментацією зображень називають процес розподілення зображення на певні області.

Виявлення об'єктів – це ідентифікація об'єкту, розуміння його локалізації.

Під розумінням сцени мається на увазі розуміння контексту зображення, зв'язків між об'єктами. Класифікацією зображення називають процес призначення мітки зображенню [1].

В контексті розпізнавання зображень дослідники виділяють два простори даних [7]:

- простір сигналів, де об'єкти, що мають розпізнаватись, формуються, таким чином задаючи класи;

- простір ознак, де відбувається процес прийняття рішення щодо визначення класу зображення.

Під час класифікації зображенням надаються попередньо визначені класи на основі їх візуального вмісту, і вхідні зображення розподіляються на категорії. Зазвичай, процес класифікації зображень має такі кроки:

- підготовка даних – збір зображень, позначення їх відповідними класами. Іноді також включає обробку зображень для покращення якості або підготовки зображення за відповідними параметрами;

- виділення ознак – виділення характеристик (це може бути колір, форма, текстура чи інші характеристики) для класифікації;

- вибір моделі – вибір моделі класифікації, яка найкраще відповідає поставленій задачі та зображенням;

- тестування й оцінка тестування – спроби класифікації обраною моделлю на деякій обраній частині зображень;

- класифікація всіх зображень у базі за обраною моделлю [1, 3, 5].

Класифікацію зображень можна здійснити за допомогою різноманітних методів і технік. Під класифікацією з застосуванням структурних описів мається на увазі знаходження ступеня відповідності між описами зображень у формі множини векторів [2].

Загальну схему системи класифікації можна зобразити блок-схемою на рисунку 1.1 [1]. На схемі позначення z – це вектор чи множина ознак вхідного зображення, s позначає кількість ознак, f – функція, що виконує певні

обчислення щодо вхідного вектора ознак та певної інформації K , отриманій на етапі навчання, m – кількість класів, k – результат, номер визначеного класу.

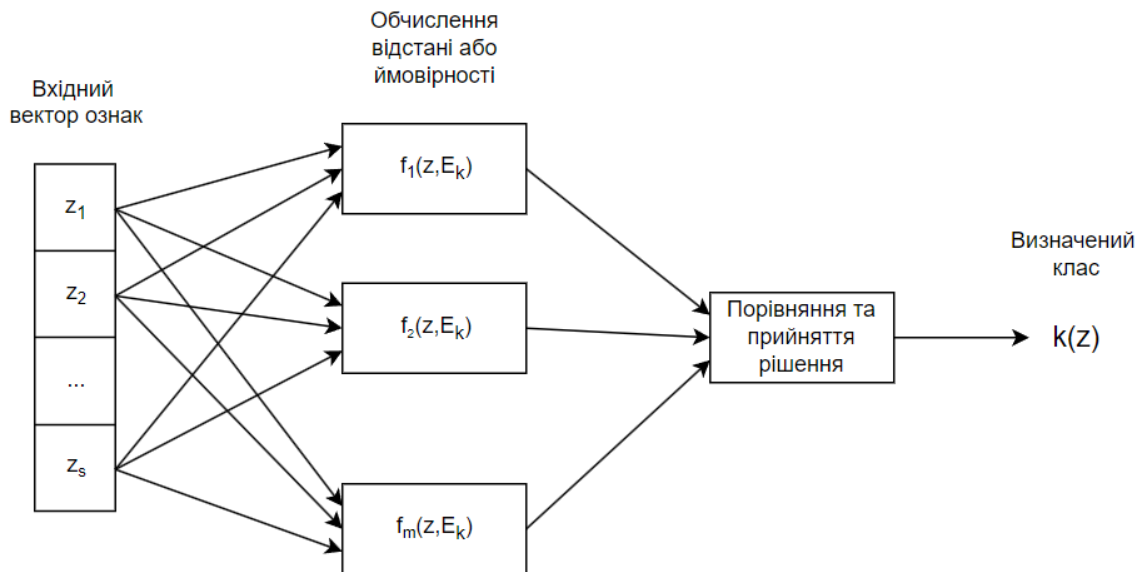


Рисунок 1.1 – Схема системи класифікації

Основними характеристиками для оцінювання результативності системи класифікації є коефіцієнт помилок, швидкодія. Помилка класифікації – віднесення об'єкту до класу k_i , при тому, що насправді класом об'єкту є k_j , $i \neq j$ і k_i не є класом всіх об'єктів, що не було віднесено до жодного класу [1].

У певних ситуаціях необхідне знаходження відповідності зображень (або об'єктів на зображеннях) до конкретних заданих еталонних зображень. Це також називають зіставленням зображень. До прикладу, така техніка може застосовуватись при виявленні дублікатів, відслідковуванні змін на знімках, об'єднанні зображень, відстеженні об'єктів.

Однією з категорій методів знаходження відповідності зображень є методи на основі структурного опису зображення у формі множини дескрипторів ключових точок [2].

Метод на основі ознак має на увазі виділення (знаходження) та зіставлення ознак зображення.

Такі методи включають декілька етапів:

- знаходження ознак – на цьому кроці на зображенні виявляються місця, точки, що могли б збігатися з іншими зображеннями, або навпаки вирізнити зображення серед інших;

- опис ознак – опис інформації області навколо кожної знайденої точки (ключової точки). Зазвичай ця інформація записується у формі дескриптора;

- зіставлення ознак – пошук відповідностей між ключовими точками шляхом порівняння дескрипторів зображень [3, 5].

Різні методи класифікації (зіставлення) зображень можуть бути кращими для різних задач, в залежності від структури, характеристик зображення та конкретної проблеми. Розуміння того, який метод є найкращим для конкретного зображення, конкретної задачі, є дуже важливим.

У класифікації зображень вже досягнуто значних результатів, проте деякі задачі залишаються проблемними, наприклад такі як наявність шумів на зображеннях, різниця освітлення, геометричні перетворення зображення або об'єкту, такі як масштабування (збільшення або зменшення розмірів зображення), обертання (обертання зображення на певний кут), зміщення та інші.

Враховуючи велику кількість інформації, в тому числі і візуальної, що має бути проаналізованою, актуальними є також розробки й дослідження щодо покращення швидкодії методів. Одним з можливих варіантів пришвидшення роботи методів є трансформація структурного опису і використання оцінювання для даних.

1.2 Апарат ключових точок та дескриптори

У методах розпізнавання зображень на основі структурного опису структурний опис – це сукупність ключових точок, що представлені векторами-дескрипторами [2].

Опис Z зображення представляється як

$$Z = \{z_v\}_{v=1}^s, \quad (1.1)$$

де z_v – дескриптор у вигляді числового вектора;

s – кількість дескрипторів ключових точок [8].

Ключова точка (КТ) – така точка зображення, яка виділяється певними особливостями, ознаками.

Приклад зображення знаходження ключових точок наведено на рисунках 1.2, 1.3.



Рисунок 1.2 – Оригінальне зображення

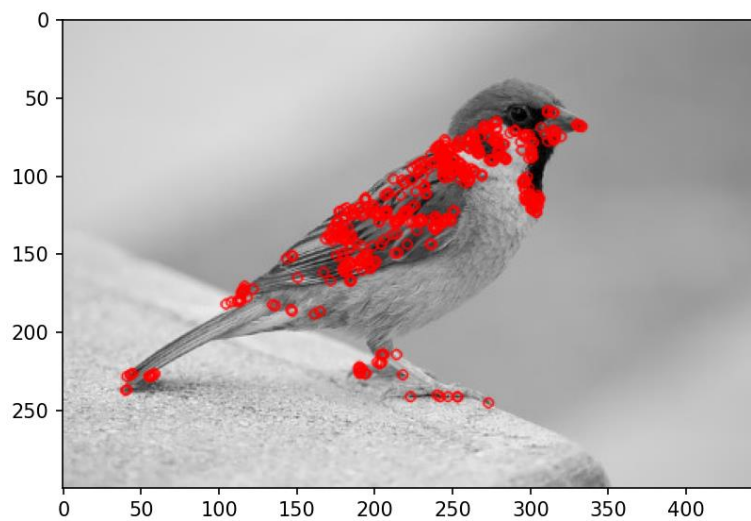


Рисунок 1.3 – Ключові точки зображення

В ідеальному випадку ключова точка має не змінюватися при різних трансформаціях зображення, відповідаючи певним характеристикам:

- унікальність – точка бути відмінною від інших сусідніх;
- інваріантність – незмінність щодо обертань, зміщень та інших геометричних перетворень;
- стабільність – незначні шуми не мають впливати на розміщення точки;
- стійкість – ключова точка не має залежати від зміни освітлення, яскравості [9, 10].

Для кожної ключової точки обчислюється дескриптор – вектор ознак, що описує оточуючу її область зображення. Дескриптори є дуже важливим інструментом в області комп'ютерного зору, вони використовуються для вирішення багатьох задач, в тому числі, для задачі розпізнавання образів.

За ключовими точками й дескрипторами можна знаходити відповідності між зображеннями (рис. 1.4).

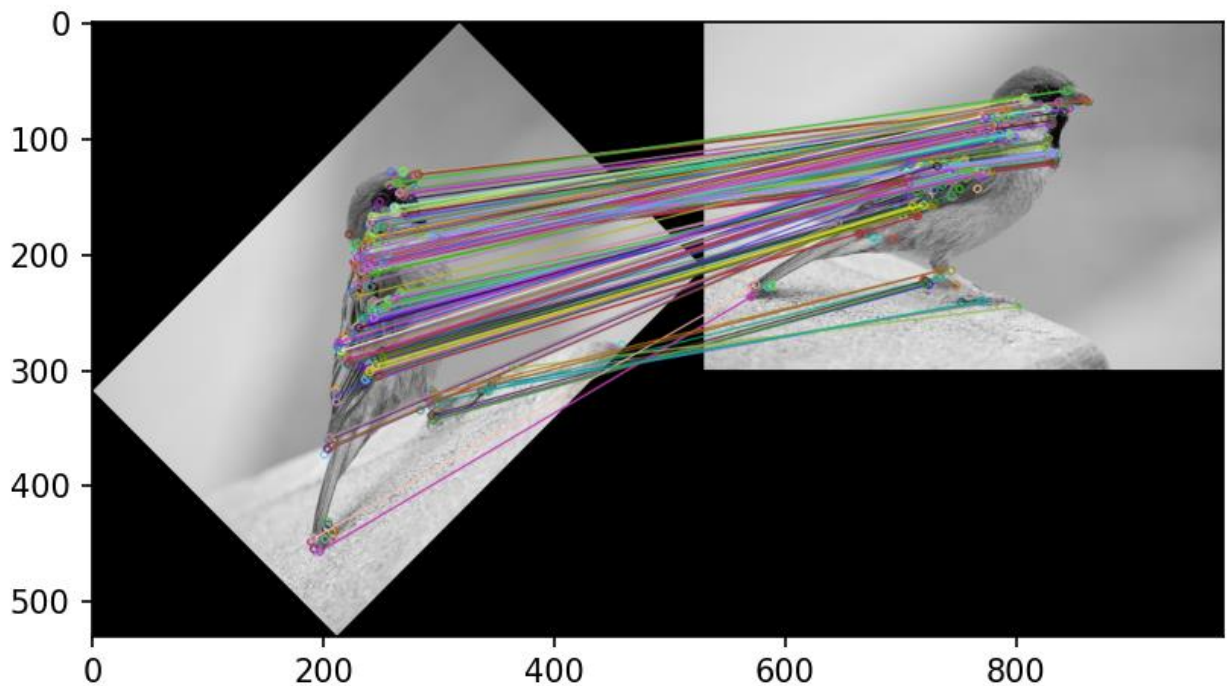


Рисунок 1.4 – Зображення знайдених відповідностей між ключовими точками зображень

Зіставлення ознак є фундаментальним аспектом в розпізнаванні та класифікації зображень. Ця задача є основою багатьох завдань комп'ютерного зору [5].

Ключові точки та дескриптори можуть бути обчислені за допомогою різних алгоритмів, як, наприклад, SIFT (Scale Invariant Feature Transform), SURF (Speeded up Robust Features), BRISK (Binary Robust Invariant Scalable Keypoints), ORB (Oriented FAST and Rotated BRIEF).

Детектор SIFT для визначення ключових точок використовує різницю гаусіан. Після виявлення ключових точок обчислюються дескриптори враховуючи орієнтацію та величини пікселів, що знаходяться навколо. На виході отримується 128-вимірний вектор ознак.

SURF є більш швидкою альтернативою SIFT, дескриптори SURF будуються за допомогою іншого підходу, на виході зазвичай отримують 64-вимірний вектор ознак [11, 12].

BRISK та ORB є альтернативами для SIFT і SURF. На відміну від SIFT і SURF, які використовують градієнтні підходи, BRISK та ORB використовують бінарні дескриптори. Це робить їх більш швидкими та ефективними щодо використання пам'яті.

В цій роботі використовується метод ORB [11].

Детектор ORB був розроблений в 2011 році в лабораторіях OpenCV. ORB використовує алгоритм FAST (Features from Accelerated Segments Test) для знаходження ключових точок, і BRIEF (Binary Robust Independent Elementary Features) для дескрипторів.

Алгоритм FAST є алгоритмом виявлення кутів, який використовується для пошуку ключових точок. Він працює порівнюючи інтенсивність між пікселями.

Обирається піксель p , що буде розглядатись, визначається інтенсивність цього пікселю.

Розглядаються пікселі навколо нього. Якщо серед них є n пікселів, інтенсивність яких більша або менша за інтенсивність пікселя p , то p є кутом і, відповідно, ключовою точкою.

$$S_{p \rightarrow x} = \begin{cases} d, & \text{якщо } I_{p \rightarrow x} \leq I_p - t \\ s, & \text{якщо } I_p - t < I_{p \rightarrow x} < I_p + t, \\ b, & \text{якщо } I_p + t \leq I_{p \rightarrow x} \end{cases} \quad (1.2)$$

де $S_{p \rightarrow x}$ – статус пікселя (сусіднього до p);

d – темніший;

s – схожий;

b – яскравіший;

$I_{p \rightarrow x}$ – інтенсивність пікселя, що оцінюється;

I_p – інтенсивність пікселя p ;

t – порогове значення [11].

На рисунку 1.5 добре помітно, що більшість ключових точок знайдено там, де достатньо значно змінюється яскравість пікселів. В цьому випадку це зміна кольору пір'я або «границя» пташки, яка значно відрізняється від фону.

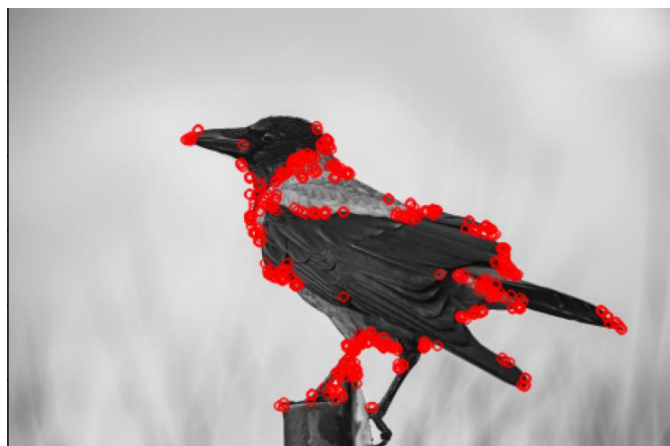


Рисунок 1.5 – Приклад ключових точок на зображенні

BRIEF є методом отримання бінарних дескрипторів ознак зображення. Спочатку вибирається набір пар координат. Після цього порівнюються

інтенсивності пікселів цих пар. Якщо перший піксель має меншу інтенсивність, записується 1, в іншому випадку записується 0. Відповідно, на виході отримується бітовий рядок розміром, що відповідає розміру набору пар (зазвичай це 256) [11].

BRIEF має певні недоліки щодо обертання, оскільки інтенсивності порівнюються не враховуючи орієнтацію. Проте під час виконання алгоритму ORB, визначені шаблони BRIEF зберігаються попередньо, а потім можуть бути адаптованими відповідно до орієнтації ключових точок.

Таким чином, кожен дескриптор ORB становить 32 байти, а у вигляді двійкового рядка 256 бітів. Порівняно з іншими детекторами, ORB вирізняється швидкістю та ефективністю використання пам'яті [9, 11].

Після знаходження описів зображень, набори дескрипторів цих зображень можна порівнювати між собою для визначення ступені схожості між ними.

Існують різні підходи для визначення ступеня релевантності даних, одним з яких є порівняння за відстанню. Це можна зробити з використанням різних метрик, включаючи Евклідову відстань, Манхетенську відстань, відстань Геммінга.

Евклідова відстань базується на теоремі Піфагора. Ця відстань вимірює відстань між двома точками на прямій, вона дуже часто використовується в різних сферах.

Манхетенська відстань визначає суму відстаней між координатами точок [13]. Тобто, якщо x та y – два вектори, то відстань $\rho(x, y)$ між ними можна обчислити таким чином:

$$\rho(x, y) = \sum_{i=1}^n |x_i - y_i|. \quad (1.3)$$

У цій роботі використовується відстань Геммінга для оцінювання близькості у векторному просторі. Відстань Геммінга використовується для підрахунку відстані між двома бінарними векторами або рядками, рівними за

довжиною. Значення відстані Геммінга відображає кількість бітів, що відрізняються у двох вхідних бінарних векторах. Для бінарних векторів (1.3) – це відстань Геммінга. Також відстань Геммінга можна записати як

$$\rho(x, y) = \text{sum}(x \neq y). \quad (1.4)$$

Важливими перевагами відстані Геммінга є такі:

- робота з бінарними векторами;
- ефективність – розрахунок представляє собою порозрядну операцію XOR і підрахунок кількості, що робить цей метод достатньо простим та швидким;
- зрозумілість – відстань Геммінга вказує на кількість відмінностей, що робить цю метрику легкою в розумінні та подальшому використанні.

У традиційних методах структурного зіставлення зображень наявний великий об'єм обчислень відстаней. Основна ідея таких методів – порівняння дескрипторів зображення, що класифікується, з дескрипторами бази еталонних зображень за допомогою деякої метрики відстані. Відстань Геммінга обчислюється відносно швидко, проте, коли зображень та дескрипторів багато, час класифікації стає великим.

Задля зменшення часу виконання, обчислювальних витрат процесу класифікації, існують такі перспективні розробки як:

- методи з оцінюванням, де обчислюється відстань не до всіх дескрипторів еталонів, а до одного;
- методи грануляції, за якими до того, як класифікувати зображення, кількість дескрипторів скорочується за близькістю ознак.

Гранулювання тут означає розбиття дескрипторів зображення на невеликі групи, гранули, об'єднані за близькістю один до одного. Це дозволяє прибрати еквівалентні дескриптори, залишивши найбільш значущі.

Грануляція дескрипторів може забезпечити:

- необхідне зменшення кількості ознак;

- оптимальне перетворення простору ознак;
- створення підмножин найінформативніших ознак [2, 4].

1.3 Постановка задачі

Покращення швидкодії класифікаторів зображень є актуальним напрямком удосконалення продуктивності розпізнавання.

Об'єктом роботи є методи класифікації зображень з використанням структурного опису у формі множини дескрипторів ключових точок.

Метою роботи є розроблення швидкісних методів класифікації на підставі трансформації структурного опису шляхом грануляції даних, аналіз їх точності та швидкодії.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз існуючих методів класифікації зображень з використанням структурного опису;
- реалізувати традиційні методи та проаналізувати їх;
- розглянути метод із застосуванням оцінювання відстані між множинами векторів;
- розробити метод грануляції даних та застосувати його до множини дескрипторів;
- проаналізувати результати, порівняти розроблені та традиційні методи за критеріями точності та швидкодії.

2 КЛАСИФІКАЦІЯ З ВИКОРИСТАННЯМ СТРУКТУРНОГО ОПИСУ ЗОБРАЖЕННЯ

2.1 Етапи і моделі аналізу даних у структурному методі класифікації

Створення бази еталонних зображень та побудова структурного опису для них проводиться до попередньому підготовчому етапі процесу класифікації виходячи із суті прикладної задачі [4].

Звичайний найпростіший структурний метод класифікації можна описати способом, зображеним на рисунку 2.1.



Рисунок 2.1 – Діаграма послідовності кроків базового структурного методу класифікації

Першим етапом є формування опису вхідного зображення. Для зображення виявляються ознаки і формується опис – ключові точки, дескриптори.

Другий етап – визначення ступеня відповідності. Між кожним описом еталонів і описом класифікованого зображення обчислюється значення відповідності за допомогою деякої метрики чи процедури голосування.

Третій – визначення класу зображення. На основі кількісних значень відповідностей визначається клас, якому з еталонів вхідне зображення відповідає найбільше [7].

В методах класифікації на основі структурного опису зображення для зіставлень ознак необхідно обчислювати відстань від кожного дескриптора зображення, що класифікується, до кожного дескриптора всіх еталонних зображень. Такі методи порівняння також називають методами перебору, або «Brute force» [10].

Якщо позначити множину дескрипторів еталону як E_k , v – номер дескриптора в еталоні, то

$$\{E_k\}_{k=1}^N = \{\{e_v(k)\}_{v=1}^s\}_{k=1}^N, \quad (2.1)$$

де k – клас еталону;

s – число дескрипторів в еталонному зображенні

$$s = \text{card}E_k. \quad (2.2)$$

Відстань ρ від дескриптора зображення, що класифікується, до множини дескрипторів одного з еталонів знаходиться за формулою

$$\rho(z, E_k) = \min_{v=1, \dots, s} \rho(z, e_v(k)), \quad (2.3)$$

де z – дескриптор аналізованого зображення.

Тобто обчислюється відстань між дескриптором зображення і кожним дескриптором еталону, і обирається найменша з них. І так з кожним дескриптором вхідного зображення.

Клас k зображення в структурних методах класифікації визначається за найбільшою кількістю голосів, зазвичай використовуючи поріг для відсіювання зображень, що не відносяться до жодного класу:

$$k = \arg \max_{i=1, \dots, N} h_i \mid h_k \geq \delta_h, \quad (2.4)$$

де δ_h – граничне мінімальне значення кількості голосів;

$\{h_i\}_{i=1}^N$ – вектор з накопиченим числом голосів за номерами класів.

У методах з роздільним способом аналізу знаходяться відстані від дескрипторів класифікованого зображення до еталонних окремо. У такому підході до відстані застосовується граничний поріг δ_ρ для того, щоб брати до уваги лише достатньо виражено близькі відстані [8].

$$\rho(z, E_k) \leq \delta_\rho. \quad (2.5)$$

Обчислюється кількість відстаней менших за граничний поріг δ_ρ для кожного еталону. Еталонне зображення з найбільшою кількістю відстаней вважається таким, до класу якого належить вхідне зображення за моделлю (2.4).

Такі методи є достатньо точними, проте мають великий обсяг обчислень під час класифікації.

2.2 Оцінка ефективності методу класифікації

Після створення методу класифікації зазвичай існує необхідність проведення оцінки результатів задля визначення ефективності моделі та можливого покращення подальших результатів.

Є декілька підходів до оцінювання класифікації. Найпростішим і найбільш популярним варіантом є обчислення відношення кількості правильно класифікованих зображень r_p до загальної кількості класифікованих зображень r .

$$pr = r_p/r. \quad (2.6)$$

Такий показник може бути застосований не тільки стосовно зображень, що класифікуються, а й стосовно дескрипторів. Ці показники пов'язані між собою, проте їх окремий аналіз також може бути інформативним.

Існує також поняття критерію впевненості, що, у випадку класифікації на основі структурного методу може застосовуватись відносно голосів еталонів. Він розраховується як нормована різниця між числом голосів обраного класу h_{max1} і числом голосів наступного класу, який є найближчим до максимуму h_{max2} . Одним з варіантів нормування є ділення на h_{max1} , інший спосіб – ділення на максимальне число голосів (сума всіх голосів за всіма класами)

$$\Delta = \frac{(h_{max1} - h_{max2})}{sum(h_i)}. \quad (2.7)$$

Цей показник показує міру впевненості моделі у класифікації певного об'єкту [8].

Також інструментом оцінювання є матриця помилок (невідповідностей). Матрицею помилок називають таблицю (або матрицю), де рядки відповідають фактичним класам, а стовпці – класам, визначеними моделлю. Таким чином матриця невідповідностей візуалізує відповідність між фактичними та прогнозованими класами.

Матриця невідповідностей має такі елементи:

- TP (true positive – істинно позитивний) – кількість правильно класифікованих об'єктів;
- FN (false negative – хибно негативний) – кількість не класифікованих об'єктів, що мали бути класифіковані;
- FP (false positive – хибно позитивний) – кількість неправильно класифікованих об'єктів;

– TN (true negative – істинно негативний) – кількість об’єктів, що не відносяться до жодного класу, і не були класифіковані моделлю до жодного з них [5, 14].

На рисунку 2.2 [5] A_1 та A_2 належать класу A (позначений колом), B_1 та B_2 – ні. На цьому рисунку A_1 є істинно позитивним елементом, A_2 – хибно негативним, B_1 – хибно позитивним, B_2 – істинно негативним [5].

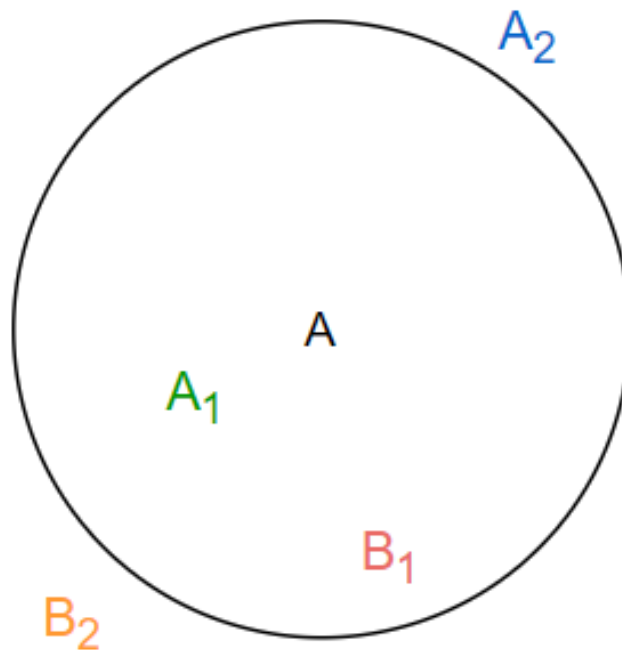


Рисунок 2.2 – Графічне зображення елементів матриці невідповідностей

В залежності від конкретної проблеми, що вирішується, граничне значення класу в схожих ситуаціях до проілюстрованого може бути збільшене. Розглядаючи рисунок 2.2, якщо радіус кола збільшити, A_2 буде класифікуватися правильно, стане істинно позитивним, але, в цьому випадку, B_2 стане хибно позитивним [1, 5].

Якщо класів два (зазвичай, це еталонний клас та клас всіх зображень, що не належать еталонному), матриця містить відповідно 4 елемента: TP, FN, FP, TN. Для багатьох класів розташування елементів матриці невідповідностей зображено на рисунку 2.3 [14].

| | | Прогнозовані класи | | | |
|----------------|---|--------------------|----|----|----|
| | | Класи | A | B | C |
| Фактичні класи | A | TN | FP | TN | TN |
| | B | FN | TP | FN | FN |
| | C | TN | FP | TN | TN |
| | D | TN | FP | TN | TN |

Рисунок 2.3 – Матриця невідповідностей для багатьох класів

За даними матриці помилок також можна обчислити такі оцінки:

– відсоток справжніх позитивних результатів $TPR = TP / (TP + FN)$ вимірює частку правильно класифікованих об'єктів від фактично правильних;

– відсоток несправжніх позитивних результатів $FPR = FP / (FP + TN)$ вимірює ймовірність визначення негативного значення як позитивний;

– позитивне прогнозоване значення $PPV = TP / (TP + FP)$ вимірює частку правильно класифікованих позитивних результатів поміж усіх позитивних.

В ідеальному випадку відсоток справжніх позитивних результатів має бути близьким до 1, а відсоток несправжніх позитивних результатів – до 0. Це буде вказувати на те, що класифікатор добре виявляє позитивні результати і мінімізує кількість помилкових позитивних припущень [5].

2.3 Створення бази зображень для тестування

Для тестування методів класифікації необхідна база тестових зображень, що буде включати зображення, які відносяться до еталонних зображень, та зображення, що не відносяться до жодного з них.

У цій роботі для аналізу ефективності методів класифікації на підставі трансформації структурного опису використовуються зображення, сформовані за допомогою афінних перетворень. До еталонних та не еталонних зображень було застосовано поворот та масштабування.

Афінні перетворення зберігають паралельність прямих, та залишають пропорційність зображення [5, 15–**Error! Reference source not found.**].

Поворот відносно точки та масштабування є широко застосованими перетвореннями на площині. Це лінійні перетворення [1, 21].

Для повороту двовимірної точки $P = [x, y]$ на певний кут θ проти годинникової стрілки відносно початку координат, створюється матриця повороту

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \cos \theta - y \sin \theta \\ x \sin \theta + y \cos \theta \end{bmatrix}, \quad (2.8)$$

де $[x', y']$ – нові координати точки P .

Поворот можна також описати формулою

$$R\theta([x, y]) = R\theta(x[1,0] + y[0,1]) = x[\cos \theta, \sin \theta] + y[-\sin \theta, \cos \theta] = [x \cos \theta - y \sin \theta, x \sin \theta + y \cos \theta]. \quad (2.9)$$

R позначає операцію повороту, $[1,0]$ та $[0,1]$ – це базисні вектори, до яких застосовується поворот [1].

Для повороту зображень використовується така модель для кожного пікселя. Приклад повороту зображення наведено на рисунку 2.4.

Масштабування також є популярним перетворенням на практиці. Воно має на увазі зміну розміру об'єкта змінюючи координати. За однорідного масштабування розмір змінюється пропорційно, збільшуючи або зменшуючи координати за коефіцієнтом масштабування [23–29].

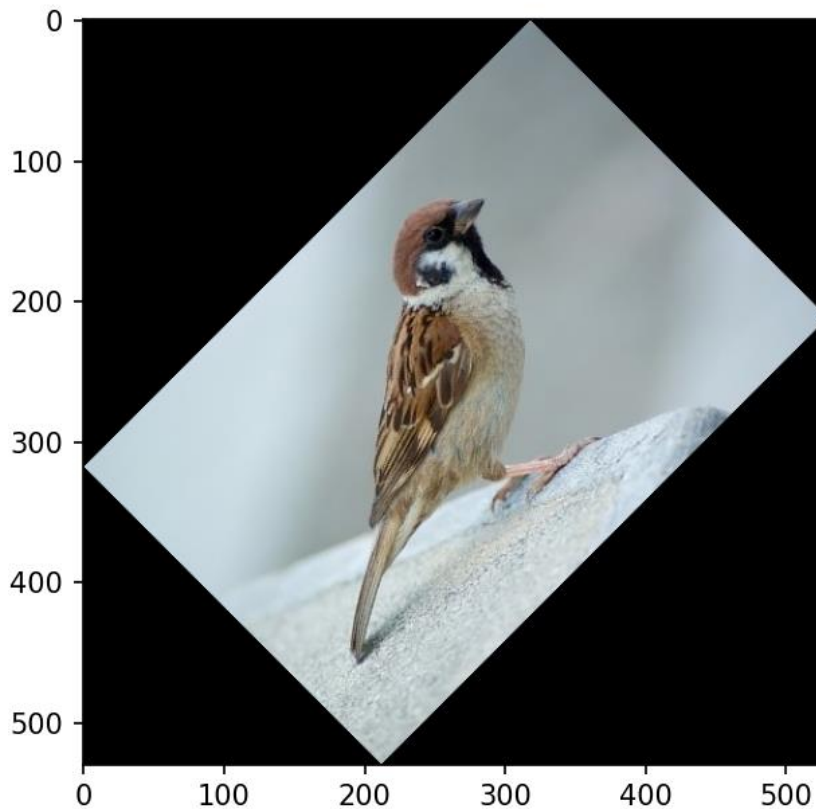


Рисунок 2.4 – Приклад обертання зображення

Якщо результуючі координати точки $P = [x, y]$ мають однакові коефіцієнти масштабування c , зміну розміру точки можна описати таким чином:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} c & 0 \\ 0 & c \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = c \begin{bmatrix} x \\ y \end{bmatrix}. \quad (2.10)$$

У випадку, коли координати точки масштабуються за різними величинами, формула виглядає так:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} c_x & 0 \\ 0 & c_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c_x x \\ c_y y \end{bmatrix}, \quad (2.11)$$

де c_x та c_y – коефіцієнти масштабування координат x та y відповідно [1, 30].

При зменшенні або збільшенні зображення таким чином змінюється кожен піксель.

На рисунку 2.5 зображено масштабування зображення. За осями координат видно, що розмір зображення на рисунку 2.5 (а) 450×300 , а на рисунку 2.5 (б) 500×350 .

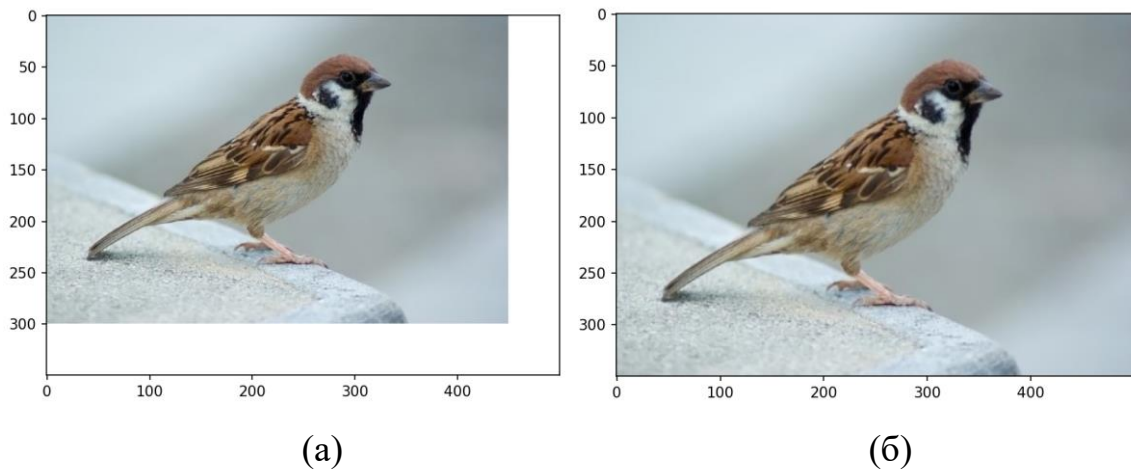


Рисунок 2.5 – Приклад збільшення зображення:

- (а) початкове зображення розміром 450×300 ;
- (б) масштабоване зображення розміром 500×350

Для зображень під час перетворень застосовується також інтерполяція, що визначається процесом створення поверхні плавної інтенсивності на основі існуючих точок зображення. Методи інтерполяції застосовуються до багатьох геометричних перетворень для покращення якості зображення, заповнення прогалів (наприклад, при збільшенні зображення) тощо [31].

2.4 Структурні методи з використанням медоїду

Одним з методів, що значно скорочує кількість обчислень і пришвидшує процес класифікації, є класифікація за медоїдом.

Медоїд – це така точка множини, що має мінімальну сумарну відстань до всіх інших точок. В цьому випадку, медоїдом є дескриптор з найменшою сумою відстаней до інших дескрипторів цього зображення [8, 32].

Щоб визначити медоїд, потрібно обчислити відстань між кожним дескриптором зображення e_v , та кожним іншим дескриптором цього ж зображення e_j , при чому $v \neq j$. І дескриптор, сума всіх відстаней якого виявиться мінімальною, визначається медоїдом [33–37].

$$m = \arg \min_{v=1, \dots, s} \sum_{j=1}^s \rho(e_v, e_j), \quad (2.12)$$

де ρ – відстань Геммінга.

Медоїд визначається для кожного еталонного зображення, і під час класифікації обчислюються відстані від кожного дескриптора зображення, що класифікується до кожного медоїда.

Таким чином, якщо s – кількість дескрипторів, знайдених на кожному еталоні (окремо), то безпосередньо під час класифікації одного зображення проводиться обчислень в s разів менше, ніж за традиційними структурними методами.

Також для зменшення кількості обчислень існують методи з використанням техніки оцінювання відстані [8, 38, 39]. Можливий підхід – використання методу оцінювання відстані від дескриптора зображення, що класифікується, до еталонного за допомогою правила трикутника.

Для цього для початку необхідно вибрати один дескриптор серед усіх описів кожного еталонного зображення. Ним запропоновано використовувати медоїд. Потрібно знайти відстань (запропоновано максимальну або мінімальну) від медоїду до іншого дескриптора цього ж зображення.

Під час класифікації треба знаходити відстані від кожного дескриптора вхідного зображення до множини медоїдів усіх еталонів.

Якщо ці відстані зобразити у вигляді трикутника (рис. 2.6 [8]), до них застосовується правило трикутника: сума довжин двох будь-яких сторін трикутника більша за довжину третьої сторони цього трикутника, а різниця – менша.

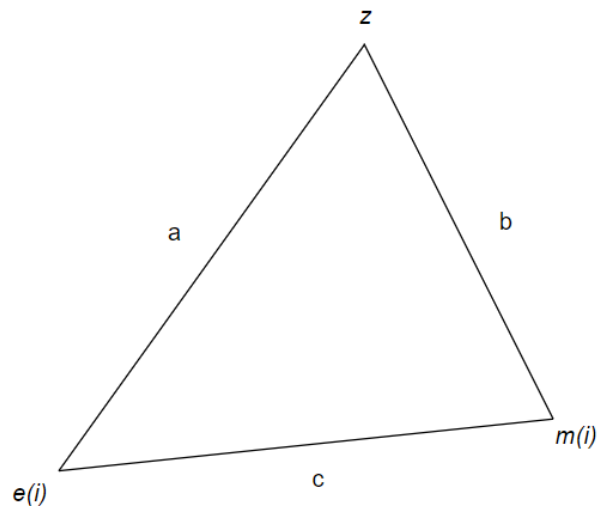


Рисунок 2.6 – Трикутник, утворений відстанями

Згідно з цим, отримано нерівність за якою можна оцінювати відстань:

$$|c - b| \leq a \leq c + b, \quad (2.13)$$

де a – відстань, що оцінюється;

b – відстань від медоїду еталонного зображення до дескриптора зображення, що класифікується;

c – відстань від медоїду до дескриптора того самого еталонного зображення.

В контексті простору дескрипторів (2.7) буде виглядати як

$$\rho(e(i), m(i)) - \rho(z, m(i)) \leq \rho(z, e(i)) \leq \rho(e(i), m(i)) + \rho(z, m(i)), \quad (2.14)$$

де $m(i)$ – медоїд еталонного зображення;

$e(i)$ – деякий дескриптор еталонного зображення, що не є $m(i)$;

z – дескриптор зображення, що класифікується (рис. 2.6);

ρ – функція відстані.

Відстань c може бути різною, запропоновано використовувати

$$c_{i,min} = \min_{m(i) \in E_i^*} \rho(e(i), m(i)) \quad (2.15)$$

або

$$c_{i,max} = \max_{m(i) \in E_i^*} \rho(e(i), m(i)). \quad (2.16)$$

Значення $c_{i,min}$ та $c_{i,max}$ отримуються до класифікації, тож на результати швидкодії процесу класифікації не впливає [8].

Методи з оцінювання відстані також збільшують швидкодію приблизно в стільки разів, скільки дескрипторів має кожний еталон.

2.5 Застосування методів грануляції даних

Одним із ефективних способів пришвидшення структурних методів є зменшення кількості дескрипторів. Можливе скорочення випадковим чином, проте це може значно знизити якість класифікації.

Процес зменшення кількості дескрипторів називається відбором інформативних або значущих ознак. Для цього необхідно створити підмножину з меншою кількістю векторів за допомогою деякого алгоритму відбору характеристик, таким чином сформувавши стиснений опис [2, 7, 38].

Способом зменшення кількості дескрипторів і, відповідно, збільшення швидкодії може бути грануляція (рис. 2.7).

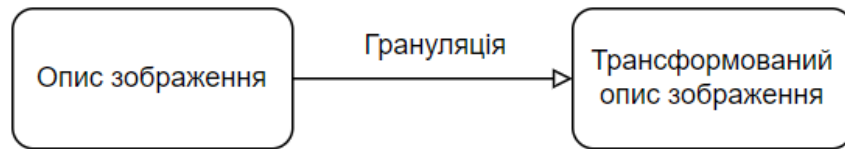


Рисунок 2.7 – Діаграма грануляції

Можливість гранулювати інформацію є важливою характеристикою інтелектуальних систем розпізнавання зображень.

Інформаційна гранула – це об’єднання елементів, що мають дуже схожі характеристики. Її можна визначити як частина універсуму (опису), де встановлені відношення еквівалентності (подібності) між елементами. Таким чином, опис може бути представлений множиною гранул. Гранулювання може бути здійснене чіткими або нечіткими підходами [2, 7].

У цій роботі розглядається гранулювання на підставі подібності дескрипторів.

Трансформація множини дескрипторів еталону визначається так:

$$T(E_k) \rightarrow E_k^*, \text{ card}(E_k^*) \ll s, \quad (2.17)$$

де E_k^* – трансформований опис, який отримується в результаті трансформації шляхом відбору підмножини $E_k^* \subset E_k$ за певним критерієм грануляції [39, 40].

Основою для грануляції може бути розгляд метричної близькості дескрипторів. Одним із варіантів такої грануляції може бути гранулювання за порогом еквівалентності. За таким методом для кожного дескриптора еталонних зображень $e_d(k) \in E_k$ обирається найближчий дескриптор $e_*(k)$

$$e_*(k) = \arg \min_{v \neq d} \rho(e_v(k), e_d(k)). \quad (2.18)$$

Після цього, перевіряється умова еквівалентності

$$\rho(e_d(k), e_*(k)) \leq \delta_\rho, \quad (2.19)$$

де δ_ρ – деяке порогове значення, що встановлює чи є дескриптори достатньо схожими.

Це значення є деяким відсотком від максимально можливого значення відстані. Воно підбирається експериментально, залежить від бази дескрипторів еталонів [7, 41].

Грануляція в такому способі представлена як функція Ω , що приймає два аргументи і повертає значення одного з них, якщо вони відповідають умові еквівалентності. В іншому випадку, якщо за граничним значенням δ_ρ , аргументи визначені як недостатньо схожі, залишаються обидва дескриптори. Функцію Ω можна описати такою формулою:

$$\Omega[e_a(k), e_b(k)] \rightarrow e_*(k), a \neq b, \quad (2.20)$$

де $e_*(k)$ є одним з аргументів, що визначено метрично подібними, і цей аргумент належить E_k^* .

На результат гранулювання таким способом впливає значення порогу δ_ρ та порядок розташування дескрипторів в структурному описі. Кількість дескрипторів в результаті не є фіксованою, тож може відрізнитися для кожного еталонного зображення.

Інший спосіб, грануляція з відсіюванням за найменшою відстанню, ґрунтується на аналізі повної матриці відстаней між дескрипторами E_k . Відбір дескрипторів відбувається фільтруванням найближчих дескрипторів, починаючи з значення мінімальної відстані [40].

За таким способом можна видаляти фіксовану кількість дескрипторів, такий метод не залежить від порогового значення. Цей алгоритм може бути дещо складнішим в реалізації, проте, з його переваг є те, що результат не залежить від порядку слідування дескрипторів у описі.

Після застосування гранулювання і отримання трансформованого структурного опису зображення можна застосовувати моделі традиційних структурних методів, описаними моделями (2.3), (2.4).

Трансформація структурного опису зображення реалізується до початку процесу класифікації, таким чином не збільшуючи час для реалізації зіставлення. За рахунок зменшення розміру структурного опису швидкодія розпізнавання збільшується на величину, що відповідає відношенню початкової кількості дескрипторів еталону до кількості дескрипторів після грануляції [2, 7].

3 ПРОГРАМНЕ МОДЕЛЮВАННЯ МЕТОДІВ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

3.1 Обґрунтування вибору середовища програмної реалізації

В рамках кваліфікаційної роботи були реалізовані базові методи класифікації на основі структурного опису зображення, методи класифікації з використанням медоїдів та методи класифікації на підставі трансформації структурного опису зображення.

Для розробки були обрані редактор коду Visual Studio Code, мова програмування Python та бібліотека OpenCV.

Visual Studio Code – це редактор вихідного коду розроблений компанією Microsoft. Ключові особливості:

- Visual Studio Code підтримує різні фреймворки, мови програмування, в тому числі Python;
 - має зручний інтерфейс;
 - містить багато різних розширень для покращення користувацького досвіду;
 - існує інтеграція з Git що дозволяє зручно працювати з версіями коду;
 - Visual Studio Code доступний на для Windows, macOS, Linux [42].
- Python – високорівнева мова програмування. Має такі характеристики:
- інтерпретована – виконується без необхідності компіляції в машинний код;
 - об'єктно-орієнтована – дає можливість створювати класи та об'єкти з властивостями та методами;
 - має динамічну семантику, що дає можливість зміни поведінки коду під час його виконання;
 - підтримка модулів та пакетів – дозволяє використовувати сторонні бібліотеки;

- простий, зрозумілий синтаксис;
- велика кількість бібліотек, що полегшує процес розробки [39, 43].

Python є популярною мовою програмування та використовується у багатьох сферах завдяки своїй універсальності, легкості використання, великій кількості бібліотек за різними сферами, включаючи обробку даних, штучний інтелект, веброзробку та інші.

Під час розробки використовувалось інтерактивне середовище Jupyter Notebook. Воно дозволяє виконувати код частинами, і одразу отримувати результат конкретного фрагменту. Це спрощує та пришвидшує процес розробки, дозволяючи тестувати частини коду окремо, без повторного виконання всього попереднього скрипту, що є дуже зручним для експериментів з різними варіантами.

Головними перевагами Jupyter Notebook є:

- інтерактивність – виконання коду по частинах;
- візуалізація результатів фрагменту коду;
- можливість збереження результату виконання скрипту – дозволяє об'єднувати та зберігати код, описи, результати коду (в тому числі візуалізації) в одному документі [35, 44].

На рисунку 3.1 наведено приклад вигляду та застосування розширення Jupyter Notebook в VS Code. На ньому зображено дві клітинки коду з результатами виконання конкретно цих частин коду.

Наявність такого розширення дозволяє працювати з файлами Jupyter Notebook безпосередньо в Visual Studio Code. Тобто є можливість використання функціональності інтерактивного середовища та можливості редактору коду одночасно.

Клітинки коду можна виконувати по одній, групами, або всі. Їх також можна переносити, міняти місцями між собою. Нові клітинки можна додавати в будь-якому місці, на початку, в середині та в кінці файлу.

Є підтримка налагодження коду, а саме виставлення точок зупинок для контролю та аналізу роботи програми. Після завершення роботи, файл Jupyter

- доступна для багатьох мов програмування, включаючи Python;
- Бібліотека OpenCV може працювати як з зображеннями, так і з відео;
- має гарно описану документацію, що допомагає розробникам швидше розбиратись в її методах [11, 45].

Застосовувалась також бібліотека SciPy. Це бібліотека у мові програмування Python з відкритим вихідним кодом, що має багато алгоритмів для наукових обчислень. Вона не є спеціалізованою бібліотекою для роботи з зображеннями, проте містить модуль `scipy.ndimage`, де є деякі функції обробки зображень. А також модуль `scipy.spatial` з алгоритмами різних операцій у просторі, в тому числі роботи з точками [46].

Для візуалізації даних існує декілька бібліотек. Було використано бібліотеку Matplotlib, яка є однією з найпопулярніших. Вона має модуль `matplotlib.pyplot`. Ця бібліотека дозволяє виводити графіки, діаграми. А також підтримує різні формати зображення. Їх можна виводити з координатними осями, що може бути зручним для розуміння масштабу [47].

3.2 Особливості програмної реалізації

Для початку роботи над реалізацією методів необхідно було обрати набір зображень для подальшого аналізу, які будуть еталонними та визначати класи. Об'єкти мають бути високої якості, мати однаковий розмір.

Спочатку було обрано 5 еталонних зображень пташок однакового розміру (450×300). Еталонні зображення 1–5 відображені на рисунках 3.2 (а), 3.2 (б), 3.2 (в), 3.2 (г), 3.2 (д). Зображення були взяті з ресурсів [48–50].

Зображення завантажуються використовуючи функцію бібліотеки OpenCV `cv2.imread()`, яка використовується для зчитування зображення з файлу, що передається. Ця функція використовує колірну модель BGR (синій, зелений, червоний), тому якщо є необхідність використовувати функції, що

працюють з RGB (червоний, зелений, синій), потрібно застосовувати функцію конвертації. До прикладу, функції виводу Matplotlib використовують RGB.

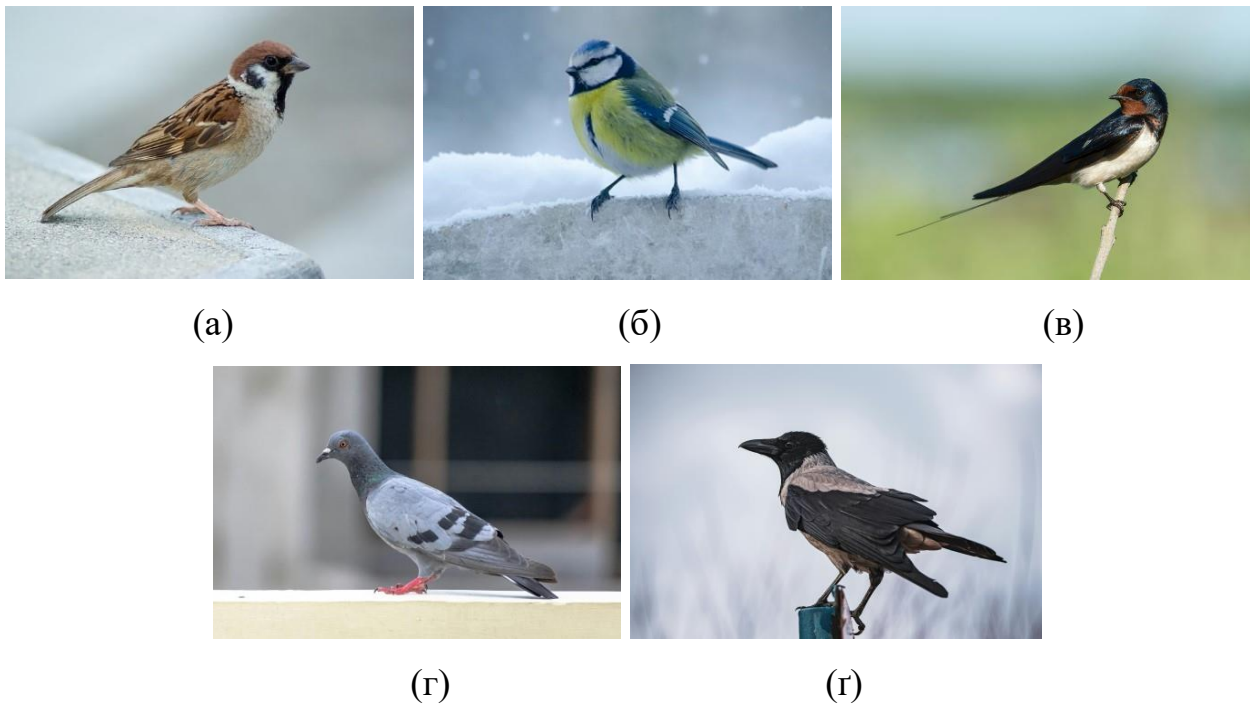


Рисунок 3.2 – Еталонні зображення:

- (а) еталонне зображення 1; (б) еталонне зображення 2;
 (в) еталонне зображення 3; (г) еталонне зображення 4;
 (e) еталонне зображення 5

Для кожного зображення було знайдено дескриптори (до 500) за допомогою детектора ORB. Для цього було створено об'єкт ORB features за допомогою `cv2.ORB_create(500)` та знайдено ключові точки та дескриптори методом `features.detectAndCompute()`. На рисунках 3.3 (а), 3.3 (б), 3.3 (в), 3.3 (г), 3.3 (e) зображені еталони 1–5 з ключовими точками.

В результаті роботи цих функцій отримуються вектори дескрипторів, що містять 32 числа, кожне з яких позначає певну характеристику ключової точки. Чисел в векторі 32, оскільки довжина дескрипторів ORB за замовчуванням 32 байти. Це відповідає 256 бітам. Дескриптори було переведено в бінарний вигляд. Таким чином, для кожного зображення отримується по приблизно 500 бінарних векторів довжиною 256.

Така процедура застосовується до кожного зображення. Тому функції для її реалізації були віднесені в окремий модуль для того, щоб не повторювати однаковий код багато разів.

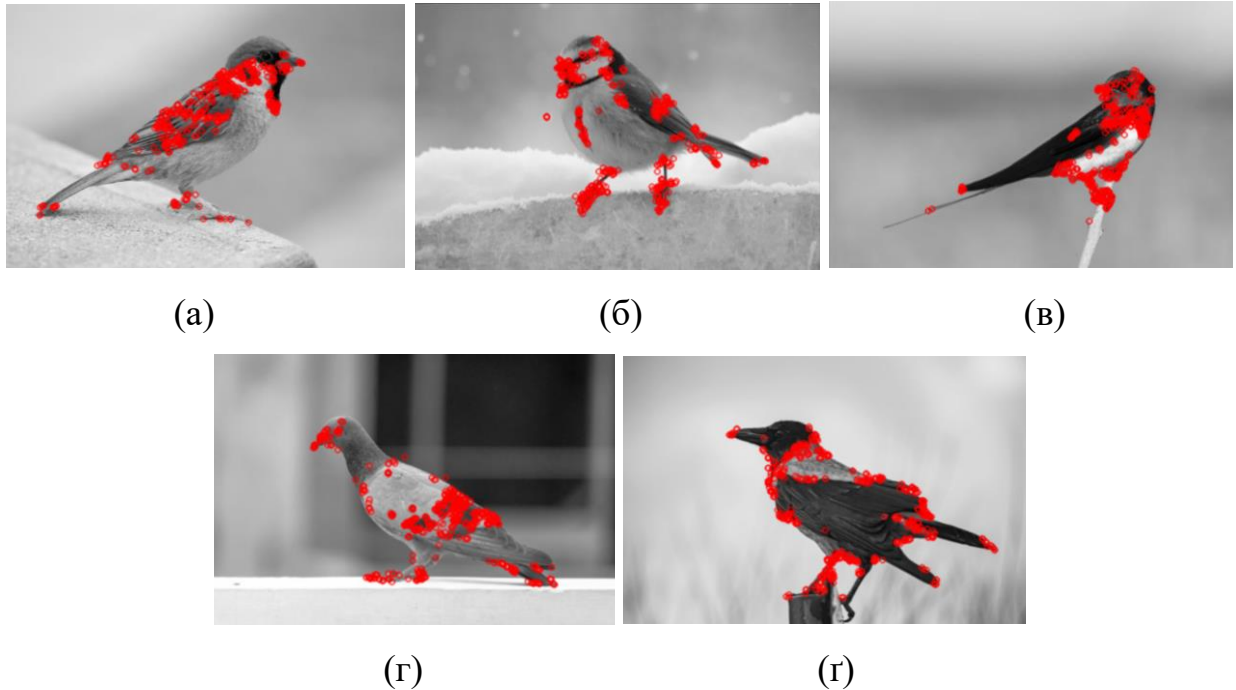


Рисунок 3.3 – Еталонні зображення з ключовими точками:

а) еталонне зображення 1; б) еталонне зображення 2;

в) еталонне зображення 3; г) еталонне зображення 4;

д) еталонне зображення 5

Для зручного, ефективного програмного коду прийнято використовувати патерни проєктування. Патерни (шаблони) проєктування – стандартний спосіб вирішення найрозповсюдженіших задач проєктування архітектури програми. Вони представляють собою загальний опис для вирішення проблеми і можуть мати різну безпосередню реалізацію [51].

Патерни проєктування поділяються на такі основні групи:

- патерни створення (породжуючі патерни) – визначають способи створення об'єктів;

- структурні патерни – надають механізми для створення продуктивних, практичних зв'язків між об'єктами;

– патерни поведінки – надають різні способи організації взаємодії між об'єктами [51].

В цій роботі використовується поведінковий шаблон «Стратегія». Цей патерн допомагає організувати класи зі схожими алгоритмами, що мають схожі або однакові функції, часто змінюються або розширюються, так, щоб це було найбільш ефективно. Це реалізується за допомогою наступних класів:

- контекст – має поле, що зберігає вибір стратегії та взаємодіє з конкретними стратегіями через абстрактну стратегію;
- абстрактна стратегія – абстрактний клас або інтерфейс, що оголошує загальний метод, який використовує контекст;
- конкретні стратегії – класи, що реалізують абстрактну стратегію, надаючи конкретну реалізацію методу [51].

В даній роботі цей шаблон використовується окремо для реалізації методів класифікації та методів зменшення кількості дескрипторів. Для кожної реалізації патерну було створено клас контекст, абстрактний клас для стратегій та класи стратегій.

Для прикладу на рисунку 3.4, наведено діаграму класів методів грануляції.

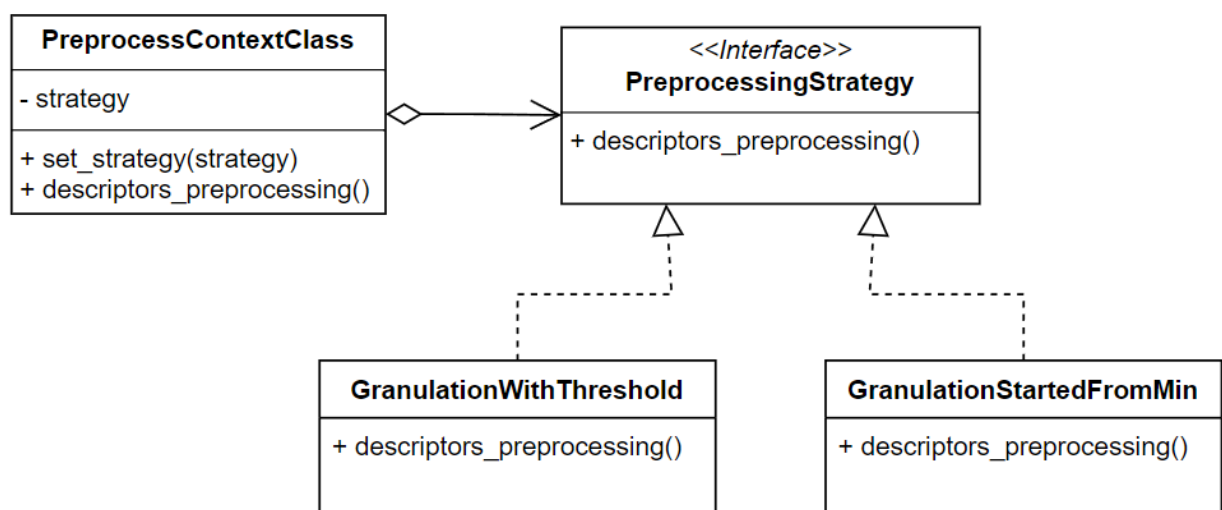


Рисунок 3.4 – Діаграма класів методів грануляції

Цей патерн був особливо корисним при тестуванні методів на великій базі зображень за рахунок простоти зміни стратегії під час виконання коду, а також уникання дублювання, оскільки весь код, що відповідає алгоритму методу знаходиться у відповідному класі.

Класи `GranulationWithThreshold` та `GranulationStartedFromMin` імплементують `PreprocessingStrategy` (рис. 3.5), надаючи свою реалізацію алгоритму. `PreprocessContextClass` (рис. 3.6) зберігає вибір стратегії, може викликати конкретну.

```
class PreprocessingStrategy:
    def descriptors_preprocessing(self, bit_des_etalons):
        pass
```

Рисунок 3.5 – Приклад абстрактного класу стратегії

```
class PreprocessContextClass:
    def __init__(self, strategy):
        self.strategy = strategy

    def set_strategy(self, strategy):
        self.strategy = strategy

    def descriptors_preprocessing(self, bit_des_etalons):
        return self.strategy.descriptors_preprocessing(bit_des_etalons)
```

Рисунок 3.6 – Приклад класу контексту

Приклад використання різних стратегій в контексті методів грануляції наведено на рисунку 3.7.

```
granulation1 = GranulationWithThreshold()
granulation2 = GranulationStartedFromMin()
imgpreprocessor = PreprocessContextClass(granulation1)

reduced_descset_etalons1 = imgpreprocessor.descriptors_preprocessing(bit_des_etalons)

imgpreprocessor.set_strategy(granulation2)
reduced_descset_etalons2 = imgpreprocessor.descriptors_preprocessing(bit_des_etalons)
```

Рисунок 3.7 – Приклад застосування патерну Стратегія

Кожному реалізованому методу було виміряно час виконання класифікації. Це було виконано застосовуючи модуль `time`, що містить функції для вимірювання часу. Використовувалась функція `time.time()`. Вона повертає час в секундах. Визначався час початку і час завершення процесу, після чого обчислювалась різниця між ними.

На час виконання процесів може впливати складність алгоритму, складність реалізації, а також операційна система, фізичні характеристики комп'ютера та навантаження системи.

Під час розроблення першого традиційного методу класифікації, необхідно знаходити відстані Геммінга від дескрипторів зображення, що класифікується, до дескрипторів кожного еталону. Для цього для кожного дескриптора вхідного зображення необхідно шукати дескриптор еталонного зображення з мінімальною відстанню [15–19].

Обчислення відстані Геммінга було реалізовано за допомогою функції `hamming()` модуля `scipy.spatial.distance`. Ця функція повертає нормалізоване значення відстані Геммінга, через що результат функції множиться на довжину вектора дескриптору для отримання відстані в вигляді кількості позицій, що відрізняються.

Реалізація функції знаходження відстаней між дескрипторами зображень відбувається так:

Крок 1. В функцію передаються дескриптори двох зображень (класифікованого та еталонного).

Крок 2. Ініціалізується вектор `hammingdistances`, де будуть збережені відстані.

Крок 3. Починається цикл `for` для проходження по вектору дескрипторів першого зображення.

Крок 4. Ініціалізується змінна `min_distance = 256`, де буде зберігатись поточна мінімальна відстань.

Крок 5. Починається цикл `for` для проходження по вектору дескрипторів другого зображення.

Крок 6. Обчислюється відстань Геммінга між дескрипторами `distance`.

Крок 7. Визначається чи є ця відстань меншою за `min_distance`, якщо так, то `min_distance = distance`.

Крок 8. Закінчується другий цикл, значення змінної `min_distance` додається до вектору `hammingdistances`.

Крок 9. Закінчується перший цикл.

Після проходження циклу по всім дескрипторам, функція повертає вектор `hammingdistances`.

Цей вектор було збережено до вектору відстаней з усіма еталонами `allhammingdistances`, там чином `allhammingdistances` містить 5 векторів.

Далі підготовлюються дані для голосування. Для кожного дескриптора вхідного зображення необхідно знаходити найближчий серед відповідних п'яти еталонних. Ця відстань та номер еталону зберігаються до відповідних векторів `minhammingdistances` та `closestetalon`.

Голосування відбувається додаванням одиниці до відповідного елемента вектору, попередньо перевіряючи відстань граничним значенням. В прикладі реалізації, лістингу 3.1, граничне значення є рівним 64. `votesnumber` в цьому скрипті – результуючий вектор з голосами.

Лістинг 3.1 Реалізація підрахунку голосів:

```

votesnumber = []
for x in range(len(allhammingdistances)):
    count = 0
    for i in range(len(closestetalon)):
        if (closestetalon[i] == x):
            if(minhammingdistances[i]<=64):
                count = count + 1
    votesnumber.append(count)

```

Спочатку метод було випробувано на оригінальних зображеннях. В такому випадку, зображення класифікує всі дескриптори правильно: фактичний клас отримує максимально можливу кількість голосів, інші класи – жодного. Точність такої класифікації 1.

Далі метод був застосований до еталонних зображень повернутих на кут 45 градусів. Зображення були обернені застосовуючи функцію `scipy.ndimage.rotate()`, що забезпечує обертання зображення на заданий кут [17–**Error! Reference source not found.**, 46].

Здійснювалися спроби застосування методу без порогового значення та з ним. Результати класифікації кожного еталонного зображення оберненого на 45 градусів без порогового значення наведені в таблиці 3.1 у вигляді матриці невідповідностей. За рядками таблиці наведено фактичні номери відповідності зображень, які були класифіковані, еталонним, а стовпці позначають розподіл голосів за еталонами.

Таблиця 3.1 – Результати класифікації традиційним методом без порогового значення

| | | Число голосів еталонів | | | | |
|--------------------------------|---|------------------------|-----|-----|-----|-----|
| | | 1 | 2 | 3 | 4 | 5 |
| Фактичні номери еталонів | 1 | 429 | 22 | 15 | 13 | 21 |
| | 2 | 22 | 443 | 16 | 9 | 10 |
| | 3 | 7 | 13 | 463 | 9 | 8 |
| | 4 | 11 | 11 | 15 | 452 | 11 |
| | 5 | 21 | 37 | 35 | 52 | 355 |

Сума голосів кожного рядка результатів без граничного значення дорівнює 500 – кількості дескрипторів знайдених на вхідному зображенні.

З використанням порогу декілька відстаней, які більші граничного значення, не враховуються при голосуванні. Відповідно, сума елементів рядків

може бути меншою 500. В таблиці 3.2 наведені результати традиційного методу з пороговим значенням 64.

Таблиця 3.2 – Результати класифікації традиційним методом з пороговим значенням 64

| | | Голоси за номерами еталонів | | | | |
|--------------------------------|---|-----------------------------|-----|-----|-----|-----|
| | | 1 | 2 | 3 | 4 | 5 |
| Фактичні номери еталонів | 1 | 426 | 20 | 13 | 11 | 19 |
| | 2 | 15 | 437 | 14 | 7 | 8 |
| | 3 | 2 | 8 | 449 | 2 | 8 |
| | 4 | 8 | 9 | 11 | 444 | 9 |
| | 5 | 15 | 30 | 26 | 31 | 347 |

Оскільки найбільші кількості голосів за номерами еталонів знаходяться на головній діагоналі матриці, метод класифікує правильно.

В цьому випадку граничне значення майже не вплинуло на результат, але за голосами точність є кращою з порогом (табл. 3.3).

Таблиця 3.3 – Критерії впевненості традиційного методу без граничного значення та з ним

| Номер еталонного зображення | Критерій впевненості методу без граничного значення | Критерій впевненості методу з граничним значенням 64 |
|-----------------------------|---|--|
| 1 | 0,814 | 0,871 |
| 2 | 0,842 | 0,877 |
| 3 | 0,9 | 0,94 |
| 4 | 0,874 | 0,9 |
| 5 | 0,606 | 0,704 |

Далі всі процеси класифікації за цим методом проводяться з граничним значенням 64.

Наступним було розроблено метод з роздільним способом аналізу. Спочатку також обчислюються відстані між дескрипторами вхідного та кожного еталонного зображення окремо. Отримуються 5 векторів з відстанями. Потім в межах цих векторів окремо підраховується кількість відстаней, що є меншою за граничне значення. Тут граничним значенням також було обрано 64, що відповідає 25% від максимально можливої відстані 256.

Цей метод також було перевірено на зображеннях еталонів обернених на 45 градусів (табл. 3.4).

Таблиця 3.4 – Результати класифікації роздільним методом з пороговим значенням 64

| | | Голоси за номерами еталонів | | | | |
|--------------------------------|---|-----------------------------|-----|-----|-----|-----|
| | | 1 | 2 | 3 | 4 | 5 |
| Фактичні номери еталонів | 1 | 476 | 263 | 220 | 188 | 201 |
| | 2 | 277 | 462 | 249 | 199 | 239 |
| | 3 | 207 | 207 | 461 | 165 | 183 |
| | 4 | 158 | 177 | 167 | 459 | 164 |
| | 5 | 198 | 225 | 189 | 172 | 391 |

В результатах цього алгоритму кожен елемент рядка може отримати значення до 500, оскільки відстані до еталонів розглядаються роздільно. Метод також працює правильно, швидкодія такого алгоритму є приблизно такою як і в базового.

Наступними розглянутими методами були метод за відстанню до медоїда та метод з оцінюванням за правилом трикутника, що також використовує медоїд.

Тож спочатку необхідно було знайти медоїди кожного набору дескрипторів еталонних зображень.

Програмна реалізація пошуку медоїдів була організована двома циклами, що йдуть по вектору дескрипторів еталону та обчислюють суму відстаней кожного дескриптора з усіма іншими. На виході функції отримується медоїд – певний дескриптор та відповідну йому суму відстаней (рис. 3.8).

```
[1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1,
1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1,
1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0,
1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0,
1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1,
1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1,
1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0,
1, 1, 1, 1, 1, 1, 1, 1] 52882.0
```

Рисунок 3.8 – Медоїд еталону 1 та відповідна йому сума відстаней

На наведеному рисунку 3.9 представлено стовпчасту діаграму відстаней до медоїда від інших дескрипторів в межах еталонного зображення 1, де вісь ординат – значення величин відстаней.

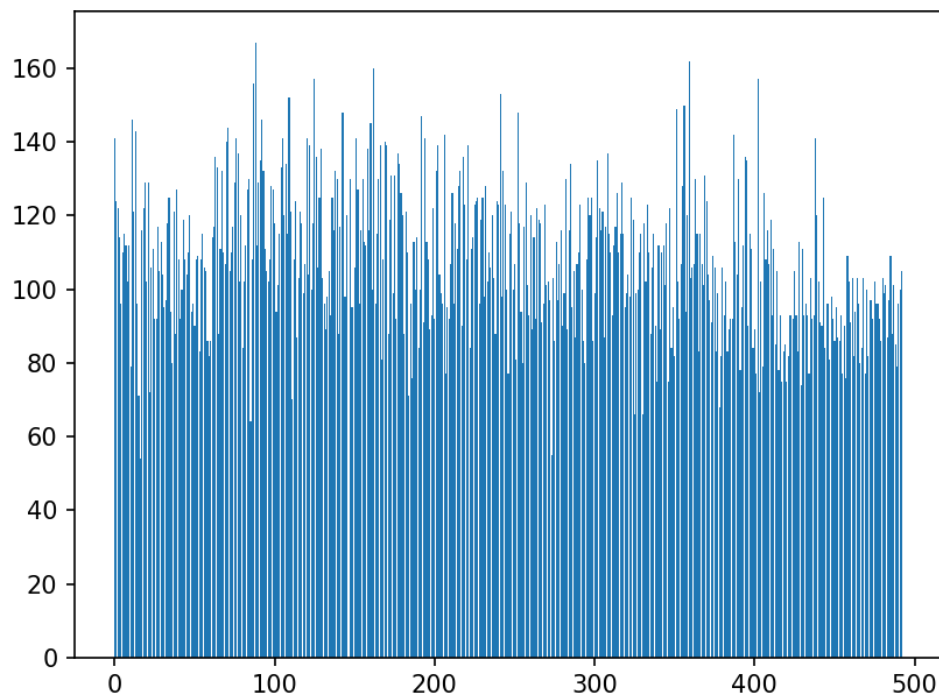


Рисунок 3.9 – Діаграма величини відстаней до медоїда

Процес класифікації за відстанню до медоїда ґрунтується на базовому методі, відповідно реалізація є такою ж, за винятком функції обчислення відстаней, оскільки тут необхідне обчислення відстаней лише до одного дескриптора еталонного зображення. Результати зіставлення обернених зображень з еталонними наведені в таблиці 3.5.

Таблиця 3.5 – Результати класифікації за відстанню до медоїда

| | | Голоси за номерами еталонів | | | | |
|--------------------------------|---|-----------------------------|-----|-----|-----|-----|
| | | 1 | 2 | 3 | 4 | 5 |
| Фактичні номери еталонів | 1 | 107 | 123 | 134 | 42 | 87 |
| | 2 | 76 | 123 | 125 | 47 | 111 |
| | 3 | 79 | 90 | 196 | 57 | 63 |
| | 4 | 62 | 68 | 160 | 132 | 59 |
| | 5 | 93 | 86 | 107 | 100 | 106 |

Результати класифікації виявились неправильними. В усіх випадках найбільшу кількість голосів отримує третій еталон.

Для методу з оцінюванням (2.14) потрібно було обчислити також мінімальну та максимальну відстані до медоїда в середині еталону (c_{min} та c_{max}). Після цього формуються нерівності. Програмно для цього було сформовані вектори, що містять два значення: перше – ліва частина нерівності (мінімум), друге – права частина (максимум).

Тестування відбувалось на оригінальних зображеннях еталонів. Були перевірені декілька варіацій методу.

Використовуючи c_{min} і порівнюючи ліві частини нерівностей, шукаючи мінімум, всі еталони були віднесені до класу першого еталонного зображення. Порівнюючи праві частини нерівностей і знаходячи мінімуми, всі еталони були віднесені до третього класу. Використовуючи c_{max} , провівши ті ж самі дії, всі зображення класифікувались як еталон 4.

За серединою інтервалу використовуючи c_{min} класифікація дала такий результат, як при класифікації за відстанню до медоїда. Дійсно, враховуючи, що відстань c_{min} завжди менше b , за (2.13), для нерівності за серединою інтервалу отримаємо

$$a \leq (b - c_{min} + b + c_{min})/2. \quad (3.1)$$

Після скорочень це буде відповідати оцінюванню за відстанню до медоїда. Таким способом, за серединою інтервалу використовуючи c_{max} отримується оцінювання за c_{max} , що також не дає потрібного результату.

Спробувавши додати нормування діленням на c_{max} всі зображення знову були віднесені до третього класу

$$a \leq (b + c_{min})/c_{max}. \quad (3.2)$$

Метод з нормуванням діленням на різницю c_{max} і c_{min} також відніс всі 5 зображень еталонів до класу третього еталону.

Методи оцінювання дійсно дуже пришвидшують процес класифікації, проте належного результату не було отримано. Результативність цих методів залежить від змісту дескрипторів [8].

Було виявлено, що для деяких варіантів методів з застосуванням медоїдів, однією з причин невірної класифікації могли бути значення c_{min} та c_{max} . У третього еталонного зображення c_{min} є найменшим з усіх, а c_{max} найбільшим. Також, ці параметри еталонів порівняно близькі. Можливо, для цих зображень коректніше підібрати інші дескриптори, замість максимально та мінімально віддалених від медоїда.

Щодо швидкодії таких методів, класифікація відбувається набагато швидше від традиційних методів. Одне зображення класифікувалось приблизно 0,14 – 0,16 секунд.

Наступною розроблювалась грануляція дескрипторів. Вона була реалізована двома способами.

Перший метод грануляції працює на основі порогового значення. Програма ітераційно проходить по вектору дескрипторів еталонного зображення, обчислюючи відстань Геммінга до кожного іншого дескриптора та знаходячи найменшу. Якщо найменша відстань менша або дорівнює граничному значенню 64, перший дескриптор пари видаляється.

Другий метод грануляції з відсіюванням за найменшою відстанню, було реалізовано таким чином:

Крок 1. Пошук найближчого дескриптора в межах кожного еталону для кожного його дескриптора і відстані між ними, запис результатів у вектор.

Крок 2. Сортування вектору з гранулами за зростанням за відстанями.

Крок 3. Початок циклу, який ітераційно проходить по кожній парі дескрипторів відсортованого вектору поки їх не залишиться половина від початкової кількості, або конкретна кількість.

Крок 4. Перевірка, що жоден з дескрипторів пари не був видалений раніше. Якщо був видалений – перехід на наступну ітерацію.

Крок 5. Видалення одного дескриптора з пари.

Вихідні дані зменшення кількості дескрипторів наведені в таблиці 3.6.

Таблиця 3.6 – Результати зменшення кількості дескрипторів

| Номер еталонного зображення | Повна кількість дескрипторів | Після першого методу грануляції | Після другого методу грануляції |
|------------------------------------|-------------------------------------|--|--|
| 1 | 493 | 263 | 250 |
| 2 | 482 | 246 | 250 |
| 3 | 485 | 258 | 250 |
| 4 | 481 | 264 | 250 |
| 5 | 492 | 257 | 250 |

Тут для другого методу зменшення набору дескрипторів було обране фіксоване значення 250.

За реалізацією методу не завжди можна видалити конкретну кількість відстаней, це залежить від структури зображення. Оскільки, якщо існує дескриптор, близький до багатьох інших, то після його видалення залишиться багато елементів без пар, які відповідно не будуть розглядатися.

Додатково було розглянуто видалення дескриптора з пари за підставою близькості до інших дескрипторів, проте це не вплинуло значно на результат.

Після методів гранулювання для класифікації було застосовано традиційний алгоритм з пошуком відстані кожного дескриптора вхідного зображення з кожним еталонних.

Спочатку тестування було проведено також на еталонних зображеннях обернених на 45 градусів.

Результати класифікації за трансформованим структурним описом за допомогою гранулювання за пороговим значенням та за найменшими відстанями наведені в таблиці 3.7.

Таблиця 3.7 – Результати класифікації після скорочення кількості дескрипторів методом грануляції за пороговим значенням та методом грануляції за найменшими відстанями

| | | Число голосів після грануляції за порогом | | | | | Число голосів еталонів після грануляції за найменшими відстанями | | | | |
|--------------------------|---|---|-----|-----|-----|-----|--|-----|-----|-----|-----|
| | | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
| Фактичні номери еталонів | 1 | 387 | 45 | 18 | 23 | 27 | 401 | 29 | 15 | 22 | 33 |
| | 2 | 38 | 395 | 25 | 18 | 24 | 20 | 438 | 14 | 14 | 14 |
| | 3 | 24 | 25 | 421 | 17 | 13 | 10 | 23 | 436 | 13 | 18 |
| | 4 | 18 | 30 | 21 | 410 | 21 | 17 | 20 | 20 | 424 | 19 |
| | 5 | 28 | 47 | 39 | 59 | 327 | 24 | 34 | 45 | 54 | 343 |

Якщо порівняти з класифікацією за початковим описом, число голосів трохи знизилось, проте результат залишився правильним, точність збереглась на найвищому рівні.

Для визначення того, чи покращує гранулювання результат порівняно з простим скороченням кількості дескрипторів, був проведений експеримент зі зменшеною кількістю дескрипторів. Це було виконано шляхом видалення елементів опису з вектору через один. Голоси класів після простого видалення дескрипторів наведені в таблиці 3.8.

Таблиця 3.8 – Голоси класів після простого зменшення кількості дескрипторів

| | | Голоси за номерами еталонів | | | | |
|--------------------------------|---|-----------------------------|-----|-----|-----|-----|
| | | 1 | 2 | 3 | 4 | 5 |
| Фактичні номери еталонів | 1 | 382 | 41 | 25 | 16 | 36 |
| | 2 | 30 | 416 | 23 | 11 | 20 |
| | 3 | 21 | 36 | 408 | 17 | 18 |
| | 4 | 24 | 27 | 21 | 415 | 13 |
| | 5 | 33 | 44 | 40 | 50 | 333 |

Виявилось, що, порівнюючи з результатами після гранулювання за порогом, деякі фактичні класи отримали більше голосів, деякі менше. Тож, в цьому випадку, деякі зображення можуть мати перевагу використовуючи гранулювання, а деякі ні. Проте при порівнянні з грануляцією з видаленням за найменшими відстанями, цей метод гранулювання однозначно дає кращі результати, ніж просте видалення.

3.3 Результати класифікації із застосуванням трансформації структурного опису та їх аналіз

Для базового методу з попередньою грануляцією за двома алгоритмами проводились тестування на більшій базі зображень. Було взято 5 еталонних зображень та 3 не еталонних зображення, що не мають бути віднесені до жодного класу. На рисунках 3.10 (а), 3.10 (б), 3.10 (в) наведено відповідно перше, друге та третє зображення, що не відносяться до еталонних. Зображення взяті з ресурсів [48–50].



(a)

(б)

(в)

Рисунок 3.10 – Зображення, що не є еталонними:

(а) не еталонне зображення 1; (б) не еталонне зображення 2;

(в) не еталонне зображення 3

Кожне зображення було повернуте на 20 та -20 градусів, збільшене та зменшене на 20%, і були виконані поєднання повороту та масштабування. Масштабування було реалізовано з використанням функції бібліотеки OpenCV `cv2.resize()`. Таким чином, разом з оригінальними зображеннями вибірка склала 72 зображення.

Для прикладу, результати класифікації зміненого третього еталонного зображення наведені в таблиці 3.9.

Таблиця 3.9 – Результати класифікації повного та скороченого набору дескрипторів еталонного зображення

| Перетворення зображення | Довжина опису | Без грануляції | Грануляція за пороговим значенням | Грануляція за мінімальними відстанями |
|--------------------------------|----------------------|-----------------------|--|--|
| Оригінальне | 485 | 0, 0, 485, 0, 0 | 8, 15, 443, 5, 3 | 4, 3, 458, 6, 2 |
| Поворот +20 | 500 | 3, 6, 451, 4, 6 | 11, 17, 401, 7, 8 | 6, 14, 413, 9, 5 |
| Поворот -20 | 500 | 9, 6, 451, 3, 6 | 21, 21, 396, 7, 10 | 16, 16, 408, 6, 6 |
| Збільшення на 20% | 500 | 9, 16, 458, 3, 3 | 13, 25, 399, 13, 17 | 15, 20, 413, 12, 12 |
| Зменшення на 20% | 433 | 7, 10, 369, 8, 15 | 13, 17, 323, 16, 18 | 10, 13, 338, 9, 16 |
| Поворот +20 та збільшення | 500 | 6, 23, 412, 4, 19 | 16, 26, 366, 11, 26 | 15, 28, 374, 11, 15 |
| Поворот +20 та зменшення | 490 | 9, 30, 375, 5, 17 | 22, 39, 337, 9, 16 | 13, 29, 340, 9, 18 |
| Поворот -20 та збільшення | 500 | 6, 20, 423, 3, 20 | 20, 29, 370, 9, 23 | 16, 26, 377, 7, 19 |
| Поворот -20 та зменшення | 495 | 15, 21, 381, 7, 25 | 27, 33, 327, 12, 26 | 18, 25, 337, 12, 26 |

Зображення були класифіковані вірно всіма методами. Відповідно, точність результатів класифікації (2.6) рівна 1.

В таблиці 3.10 наведено результати класифікації одного з зображень поза базою еталонів.

Таблиця 3.10 – Результати класифікації повного повного та скороченого набору дескрипторів зображення, що не відноситься до жодного класу

| Перетворення зображення | Довжина опису | Без грануляції | Грануляція за пороговим значенням | Грануляція за мінімальними відстанями |
|--------------------------------|----------------------|-----------------------|--|--|
| Оригінальне | 477 | 65, 117, 47, 74, 46 | 68, 89, 44, 54, 33 | 62, 81, 45, 61, 38 |
| Поворот +20 | 500 | 76, 121, 55, 61, 46 | 67, 101, 53, 47, 34 | 67, 104, 42, 55, 33 |
| Поворот -20 | 500 | 66, 113, 54, 58, 55 | 67, 99, 50, 43, 36 | 73, 88, 38, 54, 39 |
| Збільшення на 20% | 497 | 86, 130, 52, 62, 50 | 82, 111, 46, 39, 49 | 82, 92, 52, 58, 48 |
| Зменшення на 20% | 408 | 79, 87, 57, 47, 34 | 61, 72, 53, 37, 30 | 63, 77, 47, 42, 31 |
| Поворот +20 та збільшення | 500 | 76, 117, 61, 60, 48 | 70, 91, 60, 46, 49 | 64, 93, 51, 64, 38 |
| Поворот +20 та зменшення | 498 | 81, 94, 48, 64, 54 | 67, 77, 47, 47, 42 | 72, 81, 38, 57, 40 |
| Поворот -20 та збільшення | 500 | 89, 114, 53, 63, 46 | 80, 93, 49, 44, 41 | 75, 93, 39, 59, 42 |
| Поворот -20 та зменшення | 492 | 69, 103, 64, 55, 54 | 77, 86, 58, 44, 32 | 61, 79, 50, 58, 45 |

В результатах класифікації зображень, що не є еталонними, найбільшою кількістю голосів є значення 130. А для визначення класу еталонного зображення найменшою кількістю голосів є 302. Тож як поріг для відкидання класу можна визначити половину максимально можливої кількості голосів – 250.

За голосами впевненість класифікації зображень, що відносяться до певних класів, без застосування грануляції вища ніж з грануляцією. Критерій впевненості з застосуванням грануляції методом за найменшими відстанями

вищий за критерій впевненості методу за порогом майже в усіх випадках. В таблиці 3.11 наведено значення впевненості деяких трансформованих класифікованих зображень (розрахованої за (2.7)).

Таблиця 3.11 – Значення критерію впевненості за результатами класифікації деяких змінених еталонних зображень

| Без грануляції | Грануляція за пороговим значенням | Грануляція за мінімальними відстанями |
|---------------------------|--|--|
| 0,897 | 0,800 | 0,841 |
| 0,951 | 0,838 | 0,830 |
| 0,663 | 0,633 | 0,653 |
| 0,738 | 0,606 | 0,685 |
| 0,945 | 0,871 | 0,896 |
| 0,676 | 0,664 | 0,652 |
| 0,937 | 0,861 | 0,907 |
| 0,885 | 0,745 | 0,805 |

Якщо проаналізувати результати класифікації зображень поза базою, віднесення деяких зображень до класів з попередньою трансформацією структурного опису мають нижчий поріг відкидання зображення, ніж без. Тож після грануляції критерій впевненості нижчий для всіх зображень. Якщо для еталонних це не дуже добре, то для не еталонних це може бути навіть перевагою.

Деякі значення впевненості методів в класі зображення, що не відноситься до жодного, наведено в таблиці 3.12.

Таблиця 3.12 – Критерії впевненості за результатами класифікації деяких зображень поза базою

| Без грануляції | Грануляція за пороговим значенням | Грануляція за мінімальними відстанями |
|-------------------|---|---|
| 0,038 | 0,036 | 0,031 |
| 0,136 | 0,108 | 0,051 |
| 0,004 | 0,050 | 0,071 |
| 0,080 | 0,093 | 0,090 |
| 0,070 | 0,119 | 0,069 |
| 0,009 | 0,011 | 0,007 |
| 0,028 | 0,035 | 0,123 |

На рисунку 3.11 зображено графік за результатами обчислення впевненості за всіма зображеннями, де сірим кольором відображаються результати точності алгоритму без грануляції, синім та оранжевим – з грануляцією за порогом та за найменшими відстанями відповідно.

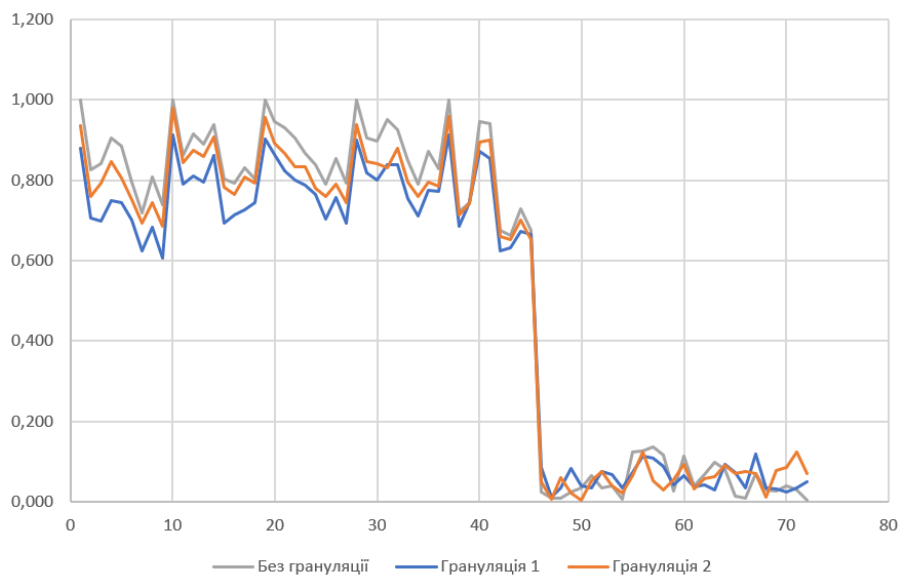


Рисунок 3.11 – Діаграма критерію впевненості класифікатора в обраному класі

Тож, за результатами тестування, після методу, де гранули сортуються за відстанями, класифікатор працює краще, ніж після методу з використанням граничного значення. Проте, хоча на швидкодію процесу класифікації це не впливає, метод за відстанями оброблює набір значень опису довше за рахунок сортування.

Грануляція прискорила процес класифікації в два рази: з повним набором дескрипторів алгоритм працював приблизно 57 секунд, зі зменшеним приблизно в два рази – 28 – 30 секунд.

До гранульованих даних можна повторити процес зменшення кількості дескрипторів ще раз, після чого буде отримано ще менший набір, що, відповідно, також зменшить час класифікації. В таблиці 3.13 наведено кількість дескрипторів після одного та двох процесів гранулювання за порогом 64, та після однієї та двох гранулювань за відстанями.

Таблиця 3.13 – Результати скорочення кількості дескрипторів

| Грануляція за пороговим значенням | | Грануляція за мінімальними відстанями | |
|--|-----------------------------------|--|-----------------------------------|
| Після першого застосування | Після другого застосування | Після першого застосування | Після другого застосування |
| 263 | 174 | 246 | 130 |
| 246 | 138 | 241 | 121 |
| 258 | 172 | 243 | 121 |
| 264 | 178 | 240 | 120 |
| 257 | 158 | 247 | 132 |

Після другої грануляції час класифікації зменшується по-різному, в кращому випадку час класифікації одного зображення зменшується в два рази. Це пояснюється тим, що кількість скорочених дескрипторів залежить від структури конкретного зображення. В середньому, час класифікації після

подвійної грануляції методом за порогом становить 18 – 24 секунд, методом за відстанями – 14 – 20 секунд.

Для методу з видаленням за відстанями фіксованим значенням обиралась половина від початкового набору, можна побачити що перший процес грануляції видалив рівно половину дескрипторів для кожного зображення. Друге гранулювання для третього і четвертого зображень видалив половину, для другого – майже половину, а для першого і п'ятого – менше, ніж половину дескрипторів.

Тестування двох процесів грануляції було проведено на першому еталонному зображенні (рис. 3.2 (а)) та його перетвореннях. Точності (2.7) наведено в таблиці 3.14. Перший рядок відповідає результату класифікації незміненого зображення, інші – зображенням з перетвореннями.

Таблиця 3.14 – Порівняння критеріїв впевненості після однієї та двох грануляцій за результатами класифікації зображень першого класу

| Після першого процесу скорочення | | Після другого процесу скорочення | |
|--|--|--|--|
| Грануляція за пороговим значенням | Грануляція за мінімальними відстанями | Грануляція за пороговим значенням | Грануляція за мінімальними відстанями |
| 0,879 | 0,93 | 0,738 | 0,769 |
| 0,706 | 0,753 | 0,606 | 0,592 |
| 0,697 | 0,785 | 0,648 | 0,66 |
| 0,748 | 0,835 | 0,683 | 0,708 |
| 0,745 | 0,805 | 0,627 | 0,678 |
| 0,701 | 0,746 | 0,607 | 0,608 |
| 0,625 | 0,691 | 0,567 | 0,587 |
| 0,684 | 0,741 | 0,605 | 0,618 |
| 0,606 | 0,685 | 0,521 | 0,605 |

Майже в усіх зображеннях критерій впевненості грануляції за мінімальними відстанями вище ніж грануляції за порогом, проте різниця стала меншою. В одному зображенні точність грануляції за пороговим значенням трохи переважає.

Порівнюючи результати других процесів зменшення дескрипторів гранулюванням та простим видаленням через один, критерії впевненості класифікації після грануляцій є кращими. Наприклад, критерій впевненості класифікації одного із зображень після грануляцій є рівним 0,607 та 0,608, а після простого видалення дескрипторів критерій впевненості класифікації того ж самого зображення дорівнює 0,493.

Загалом, гранулювання добре працює на заданій фіксованій множині зображень, точність результатів віднесення до класів залишається високою, критерій впевненості – прийнятним.

Для покращення методів гранулювання можна розглянути суміщення методів за найменшою відстанню та за граничним порогом, вони можуть покращити дію одне одного [38, 40].

Також можна додатково впровадити алгоритм вибору одного дескриптора із пари (гранули) для видалення за певною моделлю.

ВИСНОВКИ

У рамках кваліфікаційної роботи здійснено розроблення методів класифікації зображень на підставі трансформації структурного опису, заданого множиною дескрипторів ключових точок.

Для цього розглянуто традиційні методи класифікації зображень з використанням структурного опису, програмно реалізовано їх. Проведено аналіз результатів класифікації зображень на підставі апарату оцінювання і грануляції структурного опису.

Розглянуті методи проаналізовані за критеріями точності та швидкодії. Згідно з результатами, для обраних вхідних зображень методи оцінювання відстані із застосуванням медоїду та найближчих і найдальших дескрипторів від нього не показали гарних результатів.

Більш ефективними виявились методи гранулювання найближчих дескрипторів за схожістю. Вони суттєво покращують швидкодію, при цьому залишаючи прийнятну точність. Час класифікації після грануляції зменшується пропорційно до кількості дескрипторів, що залишились у трансформованому описі.

У відповідності до результатів тестування за збереженням точності віднесення до класу переважає метод грануляції, що реалізований видаленням дескрипторів, починаючи з гранули з найменшою відстанню.

Перспективи роботи можуть полягати у покращенні методу грануляції за відстанями шляхом введення обмежень за граничним значенням. Також можна проаналізувати способи вибору дескриптору з пари для видалення.

Результати роботи апробовано у вигляді тез доповідей під час Міжнародного молодіжного форуму [41] та статті [40], поданої до опублікування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Stockman, G., & Shapiro, L. G. (2001). Computer vision. Prentice Hall PTR.
2. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ, 124 с.
3. Ma, J., Jiang, X., Fan, A., Jiang, J., & Yan, J. (2021). Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision*, 129(1), 23-79.
4. 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.
5. Szeliski, R. (2022). Computer vision: algorithms and applications. Springer Nature.
6. Bai, X., Hancock, E. R., Wilson, R. C., & Ho, T. K. (2020). Special issue on recent advances in statistical, structural and syntactic pattern recognition. *Pattern Recognition Letters*, 131, pp. 46-48.
7. Гороховатський, В. О., & Гадецька, С. В. (2020). Статистичне оброблення та аналіз даних у структурних методах класифікації зображень.
8. 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.
9. Understanding Keypoints and ORB algorithm. URL: <https://medium.com/analytics-vidhya/understanding-keypoints-and-orb-algorithm-f260e6fb640c> (дата звернення 09.04.2024).
10. Тимчишин, Р. М., Волков, О. Є., Господарчук, О. Ю., & Богачук, Ю. П. (2018). Сучасні підходи до розв'язання задач комп'ютерного зору.

11. OpenCV: Feature Detection and Description. URL: https://docs.opencv.org/3.4/db/d27/tutorial_py_table_of_contents_feature2d.html (дата звернення 09.04.2024).
12. Tareen, S. A. K., & Raza, R. H. (2023, March). Potential of SIFT, SURF, KAZE, AKAZE, ORB, BRISK, AGAST, and 7 More Algorithms for Matching Extremely Variant Image Pairs. *2023 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1-6. IEEE.
13. Suwanda, R., Syahputra, Z., & Zamzami, E. M. (2020, June). Analysis of euclidean distance and manhattan distance in the K-means algorithm for variations number of centroid K. *Journal of Physics: Conference Series*, 1566(1). IOP Publishing.
14. Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for multi-class classification: an overview. arXiv preprint arXiv:2008.05756.
15. 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.
16. 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.
17. 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.
18. 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.

19. 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.
20. 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.
21. 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.
22. Gadetska, S. V., & Gorokhovatskyi, V. O. (2018). Statistical measures for computation of the image relevance of visual objects in the structural image classification methods. *Telecommunications and Radio Engineering*, 77(12).
23. 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.
24. Gorokhovatskiy, V. A. (2016). Efficient Estimation of Visual Object Relevance during Recognition through their Vector Descriptions. *Telecommunications and Radio Engineering*, 75(14).
25. 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), pp. 57-70.
26. 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.

27. 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.

28. Gadetska, S. V., Gorokhovatskyi, V. O., Stiahlyk, N. I., & Vlasenko, N. V. (2021). 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*, (4), pp. 58-68.

29. 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.

30. Гороховатський В., Творошенко І., Сидоренко Д. (2021) Класифікація зображень із використанням кластерного подання, Міжн. наук. симпозіум «Інтелектуальні рішення-С». *Обчислювальний інтелект. Теорія прийняття рішень* (Вересень 29, 2021). Київ – Ужгород, С. 44-45.

31. Gorokhovatskyi, O., Peredrii, O., Gorokhovatskyi, V., Vlasenko, N. (2023) Explanation of CNN Image Classifiers with Hiding Parts. In: J. Benoit-Pineau, R. Bourqui, D. Petkovic, G. Quenot (eds), *Explainable Deep Learning Artificial Intelligence*, pp. 125-146, Academic Press, 346 p.

32. 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.

33. Explain the concept of medoid in clustering. URL: <https://medium.com/@sujathamudadla1213/explain-the-concept-of-medoid-in-clustering-b93144887742> (дата звернення 20.04.2024).

34. 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.

35. 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.

36. Gorokhovatsky V.A., Putyatin Ye.P. (2009) Image Likelihood Measures of the Basis of the Set of Conformities. Telecommunications and Radio Engineering, 68 (9), pp. 763-778.

37. 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.

38. 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.

39. 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. 1 (47), с. 93–98.

40. 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).

41. Вечірська А. (2024). Класифікація зображень з використанням оцінювання даних. 28-ий Міжнародний молодіжний форум «Радіоелектроніка і молодь у XXI столітті», т. 7, с. 25–26.

42. Documentation for Visual Studio Code. URL: <https://code.visualstudio.com/docs#vscode> (дата звернення 25.04.2024).

43. What is Python? Executive Summary. URL: <https://www.python.org/doc/essays/blurb/> (дата звернення 15.04.2024).

44. Jupyter Notebook Documentation. URL: <https://jupyter-notebook.readthedocs.io/en/latest/> (дата звернення 29.04.2024).

45. Riba, E., Mishkin, D., Ponsa, D., Rublee, E., & Bradski, G. (2020). Kornia: an open source differentiable computer vision library for pytorch. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3674-3683.
46. SciPy documentation. URL: <https://scipy.org/> (дата звернення 19.04.2024).
47. matplotlib.pyplot — Matplotlib 3.5.3 documentation. URL: https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html (дата звернення 16.04.2024).
48. Beautiful Free Images & Pictures | Unsplash. URL: <https://unsplash.com/> (дата звернення 10.04.2024).
49. 4.2 million+ Stunning Free Images to Use Anywhere - Pixabay – Pixabay. URL: <https://pixabay.com/> (дата звернення 10.04.2024).
50. Free stock photos, Royalty free stock images & Copyright Free pictures · Pexels. URL: <https://www.pexels.com/> (дата звернення 10.04.2024).
51. Design Patterns. URL: <https://refactoring.guru/design-patterns> (дата звернення 01.05.2024).