

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 другий (магістерський)

**ДОСЛІДЖЕННЯ ПРОДУКТИВНОГО МЕТОДУ КЛАСИФІКАЦІЇ  
ЗОБРАЖЕНЬ ІЗ ВИКОРИСТАННЯМ ЗАСОБІВ ДЛЯ ОЦІНЮВАННЯ  
ВІДСТАНЕЙ У ПРОСТОРІ ДАНИХ**  
(тема)

Виконав:  
студент 2 курсу, групи ІНФМ-22-2

Рибалка М. О.  
(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 другий (магістерський)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 Рибалці Михайлу Олексійовичу  
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження продуктивного методу класифікації зображень із використанням засобів для оцінювання відстаней у просторі даних

затверджена наказом по університету від 3 листопада 2023 року № 1280Ст

2. Термін подання студентом роботи до екзаменаційної комісії 22 12 2023 р.3. Вихідні дані до роботи математичні моделі класифікації зображень, теоретичні відомості про методи класифікації зображень, теоретичні відомості про методи визначення дескрипторів ключових точок, тестові зображення, програмне забезпечення для розробки.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Ознайомлення з новітніми методами структурної класифікації зображень.2. Розгляд математичних моделей структурних методів класифікації.3. Моделювання модифікованих та традиційних методів з використання прикладної бази.4. Аналіз результатів дослідження методів класифікації зображень.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) ілюстративні зображення до огляду методів класифікації, візуалізація роботи методів, аналіз результатів роботи, тестові зображення.

---



---



---



---



---



---



---

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	03.11.2023	
2	Аналіз завдання, підбір літератури	03.11.23-05.11.23	
3	Аналіз літератури з досліджуваної проблеми	06.11.23-10.11.23	
4	Аналіз методів класифікації	10.11.23-25.11.23	
5	Розробка та порівняння методів	25.11.23-10.12.23	
6	Програмна реалізація	10.12.23-20.12.23	
7	Оформлення пояснювальної записки	20.12.23-25.12.23	
8	Перевірка на плагіат	26.12.2023	
9	Рецензування	27.12.2023	
10	Підготовка презентації та доповіді	28.12.2023	
11	Занесення роботи в електронний архів	01.01.2024	
12	Попередній захист кваліфікаційної роботи	02.01.2024	

Дата видачі завдання 3 листопада 2023 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

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

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 68 с., 6 табл., 9 рис., 53 джерела.

КОМП'ЮТЕРНИЙ ЗІР, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, КЛЮЧОВА ТОЧКА, ДЕТЕКТОР ORB, ДЕСКРИПТОР КЛЮЧОВОЇ ТОЧКИ, МЕДОЇД, ВІДСТАНЬ ХЕМІНГА, ТОЧНІСТЬ КЛАСИФІКАЦІЇ, ШВИДКОДІЯ КЛАСИФІКАЦІЇ.

Об'єктом дослідження є методи класифікації зображень із використанням засобів для оцінювання відстаней від точки до множини.

Метою є дослідження ефективності та оцінювання швидкодії традиційних та модифікованих структурних методів класифікації зображень. Опис зображення подається у формі множини дескрипторів ключових точок ORB. Застосовано і досліджено новітні модифікації методів, що значно прискорюють класифікацію шляхом впровадження процедури оцінювання при встановленні належності дескриптора до класу.

COMPUTER VISION, IMAGE CLASSIFICATION, KEYPOINT, ORB DETECTOR, KEYPOINT DESCRIPTOR, MEDOID, HAMMING DISTANCE, CLASSIFICATION ACCURACY, CLASSIFICATION SPEED.

The object of research is image classification methods using images for estimating distances from a point to a set.

The goal is to research the effectiveness and speed evaluation of traditional and modified structural image classification methods. The image description is provided in the form of a set of ORB keypoint descriptors. The latest modifications of the methods, which significantly speed up the classification by implementing the evaluation procedure when establishing the descriptor's belonging to the class, have been applied and studied.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів.....	6
Вступ.....	7
1 Структурні методи класифікації зображень та способи їх вдосконалення...	9
1.1 Аналіз літературних джерел за темою роботи.....	9
1.2 Класифікація за множиною дескрипторів.....	10
1.3 Дескриптори ORB для опису зображень.....	12
1.4 Застосування моделей нейронних мереж.....	16
1.5 Постановка задачі дослідження.....	20
2 Розроблення методів класифікації з використанням оцінювання.....	21
2.1 Реалізація методу класифікації.....	21
2.2 Застосування апарату хеш-функцій.....	24
2.3 Метод оцінювання відстані до класу.....	28
2.4 Оцінка результативності класифікації.....	34
3 Результати комп'ютерного моделювання та дослідження методів.....	37
3.1 Обґрунтування вибору програмних засобів.....	37
3.2 Особливості здійснення комп'ютерного моделювання.....	41
3.3 Результати дослідження та їх аналіз.....	53
Висновки.....	62
Перелік джерел посилання.....	63

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

ORB – Oriented FAST and Rotated BRIEF (орієнтований FAST і BRIEF з  
можливістю повороту)

CRC – Cyclic Redundancy Check (циклічний надлишковий код)

НМ – нейронна мережа

КТ – ключова точка

YOLO – You Only Live Once (ти живеш тільки раз)

Prec – Precision (точність)

Rec – Recall (повнота)

Асс – Accuracy (правильність)

КЗ – комп'ютерний зір

## ВСТУП

Напряом розпізнавання образів посідає одне з найбільш важливих місць у галузі обробки зображень, так як пов'язаний із сучасними впровадженнями систем штучного інтелекту. Прикладні розробки даного напряму використовуються у торгівельній, охоронній, науковій, військовій та інших сферах діяльності людства.

Однією з найбільш необхідних у наш час розділів розпізнавання образів є класифікація зображень [1 – 4].

Керуючись потребами прикладних завдань, сучасна наука працює над розробленням і вдосконаленням програмних систем, здатних класифікувати зображення за певними ознаками. Для підвищення точності і швидкодії класифікації розробляються нові, більш сучасні і досконалі алгоритми, та вдосконалюється апаратна частина, що використовується для класифікації зображень.

Одними із найбільш популярних ознак, що використовується для класифікації, є ключова точка зображення та її дескриптор. Їх популярність зумовлена простотою знаходження чи використання та достатньо високою точністю класифікації за даними ознаками [3, 5 – 10].

Класифікація зображень за допомогою множини дескрипторів використовує опис зображення у формі набору векторів – дескрипторів ключових точок. Важливою прикладною особливістю даного методу є стійкість дескрипторів до геометричних перетворень зображення.

Основними детекторами для формування дескрипторів ключових точок є SURF, BRISK, FAST, SIFT, BRIEF та ORB. Дескриптори, знайдені за допомогою кожного з методів, мають свої конкретні особливості, проте суть дескрипторів одна – виділити основні ознаки зображення і певним чином закодувати зображення задля ефективного аналізу [11].

Враховуючи необхідність удосконалення методів класифікації зображень, було розроблено велику кількість модифікацій. Серед них

окремою групою можна виділити методи, що використовують оцінювання відстаней у просторі даних. Під простором даних у даній роботі мається на увазі загалом нескінченна множина дескрипторів ключових точок. Оцінювання передбачає приблизне визначення відстані між множиною дескрипторів ключових точок вхідного зображення та кожною із множин дескрипторів еталонів за одним або декількома дескрипторами.

Зазвичай дескриптори для оцінювання обираються у еталонах за певними критеріями, що мінімізують відхилення відстані від множини дескрипторів ключових точок до обраних для оцінювання точок.

У рамках даної кваліфікаційної роботи проведено дослідження методів класифікації з використанням оцінювання і проаналізовано отримані результати між собою, а також з іншими методами прискорення процесу класифікації та з традиційним методом класифікації [12].

Актуальність дослідження полягає у впровадженні нових методик прискорення процесу класифікації зображень за множиною дескрипторів ключових точок та пошуку оптимального методу класифікації з урахуванням таких критеріїв, як точність та швидкодія класифікації.

Моделювання та експерименти проведені із довільними зображеннями.



# 1 СТРУКТУРНІ МЕТОДИ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ ТА СПОСОБИ ЇХ ВДОСКОНАЛЕННЯ

## 1.1 Аналіз літературних джерел за темою роботи

Проаналізуємо найбільш важливі джерела за даною тематикою.

У джерелі [1] досліджено основні методи аналізу даних та їх застосування для класифікації зображень. Особливу увагу приділено методам аналізу ключових точок та передобробці даних для аналізу. У результаті дослідження було визначено основні методи класифікації зображень, що будуть використані у роботі.

У літературі [2] проведено вивчення методів машинного навчання. Метою дослідження є перевірка основних методів дослідження та обробки даних. У результаті дослідження було визначено основні підходи і методи, що будуть використані у роботі.

Джерело [3] присвячене дослідженню дескрипторів ключових точок та впливу статистичних характеристик на результати класифікації за множинами ключових точок. Метою дослідження було вивчення впливу знайдених дескрипторів на результат класифікації. Під час аналізу роботи було уточнено основні характеристики, необхідні для оцінки дескрипторів.

У роботі [4] проведено дослідження основних методів машинного навчання та класифікації зображень. Метою дослідження було порівняння найбільш популярних методів машинного навчання. Під час аналізу даної роботи було уточнено реалізацію методів класифікації.

У джерелі [5] проведено дослідження сучасних базових методів аналізу даних. Під час виконання даної роботи було проведено аналіз базових методів класифікації.

У джерелах [6, 7] проведено дослідження основних методів знаходження ключових точок зображення. Мета досліджень – вивчення методів знаходження ключових точок та прискорення процесу класифікації

зображень. Під час аналізу даного джерела було уточнено метод пошуку ключових точок та подальше використання знайдених ключових точок.

У літературі [8, 9] проведено дослідження методів прискорення класифікації за рахунок використання кластерного аналізу. Основною метою дослідження є ознайомлення з методами прискорення класифікації зображень. Під час аналізу даної роботи вивчено альтернативні методів для прискорення структурної класифікації зображень.

У джерелі [10] проведено дослідження основних методів аналізу ключових точок. Найбільшу увагу приділено методу ключових точок ORB. Метою дослідження було визначення необхідних характеристик метода ORB. У результаті дослідження для проведення експериментів було обрано метод ORB для визначення дескрипторів ключових точок і формування опису зображення.

## 1.2 Класифікація за множиною дескрипторів

У якості опису вхідних зображень маємо набір дескрипторів ORB по 256 бітів кожен. Маємо два простори даних: простір сигналів, в якому формуються розпізнавані образи – зображення об'єктів, і простір ознак, де реалізується прийняття рішення про клас зображень. Класи об'єктів задані у просторі сигналів, а визначаються і розрізняються у просторі ознак. На рисунку 1.1 наведено приклад зображення і візуальне відображення ключових точок, сформованих детектором ORB. На зображенні ключові точки позначено фіолетовими кілочками.

Якщо  $\Omega$  – нескінченна множина зображень;  $\omega \in \Omega$  – зображення, що належить  $\Omega$ ;  $M[j] \in \{1, 2, \dots, n\}$  – множина номерів для класів, що не перетинаються образів  $\Omega_1, \Omega_2, \dots, \Omega_n$ ;  $g: \Omega \rightarrow M$  – невідома індикаторна функція, яка ставить у відповідність кожному образу  $\omega \in \Omega$  деякий номер  $j \in M$  класу, то можна розглянути  $Z = \{z | z \in R^n\}$  – у якості опису

зображення. Опис – це скінченна множина числових векторів. Кожному  $\omega \in \Omega$  можна поставити у відповідність певний опис  $Z$  [13].



Рисунок 1.1 – Зображення та його множина координат КТ

Для створення повноцінного класифікатора необхідно мати правило, яке множині  $Z$  ознак зображення ставить у відповідність номер класу зображення із переліку допустимих номерів класів. Процес класифікації – це ланцюжок перетворень даних із простору зображень у множину класів, має наступний вигляд:

$$\Omega \rightarrow \Theta(\Omega) \rightarrow Z \rightarrow g(Z) \rightarrow M. \quad (1.1)$$

Основна задача процесу (1.1) – побудова вирішального правила  $g$ , яке дозволяє встановити відповідність між множиною ознак  $Z$  та зображенням  $\omega$ . Вибір  $Z$  прямо впливає на ефективність правила  $g$ . Критерієм якості класифікації виступає результативність (правильність) прийняття рішення щодо елементів вихідного простору образів  $\Omega$ .

Ставлячи задачу розпізнавання зображень за множиною дескрипторів ключових точок, допускаємо можливість перетину зображень із простору  $\Omega$  та їх описів із простору  $Z$ .

Базовий метод класифікації полягає в наступному [1 – 3]:

- формуємо базу еталонних зображень та їхніх описів;
- будуємо опис  $O \in Z$  для розпізнаваного зображення;
- перевіряємо відповідність релевантності  $d_j(Z_j, O)$  для опису  $O$  і кожного із елементів множини  $Z_1, Z_2, \dots, Z_j$  описів еталонів;
- знаходимо оптимальне значення релевантності  $d_j$ ,  $j \in 1, \dots, n$ , і визначаємо номер  $v$  класу еталонного зображення  $v = \arg \operatorname{opt}_{j=1 \dots M} d_j(Z_j, O)$ .

### 1.3 Дескриптори ORB для опису зображень

У загальному випадку клас при класифікації визначається як нескінченна множина зображень з урахуванням його допустимих геометричних перетворень в межах поля зору комп'ютерної системи [1]. Представником класу для класифікатора є еталонне зображення, за яким і визначаємо множину дескрипторів КТ еталонного зображення [1 – 4].

Останнім часом набули прикладного застосування такі детектори КТ, такі як ORB (Oriented FAST and Rotated BRIEF) [10, 11], що обчислюють дескриптор КТ у вигляді бінарного вектора з розмірністю, кратною ступеню двійки. Бінарне представлення є вигідним завдяки тому, що дозволяє прискорити процес порівняння дескрипторів завдяки використанню бінарних операцій. Додатково бінарна арифметика надає можливість застосувати ефективний апарат обробки бінарних даних і розробити нові способи визначення подібності у просторі дескрипторів та при побудові правил класифікації зображень за множиною дескрипторів КТ.

Встановлення відповідності для пари дескрипторів ORB здійснюється за допомогою обчислення відстані Хемінга. По суті, необхідно попарно виконати логічну операцію XOR для кожного біта двох дескрипторів. Модульність побудови ORB дає можливість комбінувати побудову детектора КТ в поєднанні з довільним способом визначення його дескриптора і навпаки, оптимізуючи бажану продуктивність вирішуваних завдань [1, 10, 11].

Таким чином, основні переваги методу ORB полягають у забезпеченні високої швидкодії роботи за рахунок спрощення процесу оброблення і побудови бінарних дескрипторів КТ.

Метод ORB є одним із засобів, що найбільш часто використовуються для пошуку КТ та формування їх дескрипторів. На даний момент цей метод є одним із найбільш розвинених завдяки організованому у ньому комбінуванню переваг інших методів. У якості детектора у даному методі використовується метод FAST, а у якості дескриптора КТ – модель BRIEF [10, 11].

Подібно до детектора FAST метод ORB виконує аналіз 16 пікселів з оточення поточного пікселя  $p$ . Описаний процес проілюстровано на рисунку 1.2 [10].

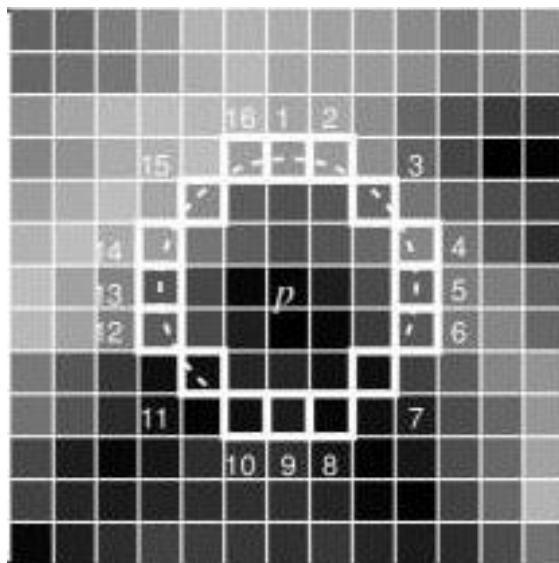


Рисунок 1.2 – Схема аналізу поточного пікселя  $p$  та його оточення

Якщо серед пронумерованих пікселів знаходяться бодай 9, для кожного з яких справедливе твердження:

$$l_p + \varepsilon < l_i, \quad (1.2)$$

або для кожного справедливе твердження

$$l_p - \varepsilon > l_i, \quad (1.3)$$

де  $l_p$  – яскравість поточного пікселя;

$l_i$  – яскравість пікселя оточення;

$\varepsilon$  – допустима гранична величина, точка вважається кандидатом у КТ.

Наступним кроком є прийняття рішення про те, є чи ні кандидат ключовою точкою. Для цього перевіряються 4 пікселі на рисунку 1.2 з номерами 1, 5, 9, 13. Якщо три з них відповідають твердженню (1.2) або твердженню (1.3), то точка вважається ключовою [1].

Метод пошуку ключових точок ORB має кращі швидкісні характеристики роботи завдяки використанню незначної кількості пікселів оточення поточного пікселя  $p$  при пошуку КТ [13 – 16].

Дескриптор ORB, відповідно до моделі BRIEF має довжину 256 бінарних компонентів (бітів). Формування дескриптора відбувається наступним чином: у межах прямокутної області – оточення поточного пікселя  $31 \times 31$  обчислюються середні значення яскравості пікселів в підобластях  $5 \times 5$ . Бінарні значення дескриптора розраховуються за формулою:

$$D_i = \begin{cases} 1, & I_x < I_y \\ 0, & I_x \geq I_y \end{cases} \quad i = \overline{0, n}, \quad (1.4)$$

де  $D$  – дескриптор;

$I$  – середня яскравість;

$x$  та  $y$  – під області  $5 \times 5$ ;

$n$  – довжина дескриптора (у даному випадку 256).

У даній роботі було використано готову реалізацію метода ORB з бібліотеки OpenCV. Бібліотека OpenCV призначена для обробки та розпізнавання зображень. Основними її перевагами є кросплатформеність, підтримка багатьох мов програмування та наявність багатьох реалізованих алгоритмів обробки зображень. Реалізація метода ORB з даної бібліотеки дозволяє ефективно знайти дескриптори ключових точок.

Використання метрики Хемінга у даній роботі зумовлене необхідністю порівняння великих обсягів бінарних даних [17 – 19]. Зручність метрики Хемінга для бінарних даних полягає в тому, що є можливість знехтувати величиною різниці елементів, врахувавши сам факт їх відмінності. Для розрахунку метрики Хемінга для двох векторів необхідно провести їх поелементне порівняння. У випадку виявлення відмінності між двома відповідними елементами векторів значення метрики Хемінга збільшується на одиницю. В іншому випадку значення метрики Хемінга не змінюється.

Класичний варіант класифікатора передбачає лінійний пошук [19]. Це фактично повний перебір множин вхідних дескрипторів КТ зображення та еталонних та їх попарне порівняння з метою підрахунку числа дескрипторів, для яких встановлено відповідність.

Даний процес вимагає значних часових ресурсів. Виникає необхідність удосконалення процесу шляхом прискорення роботи класифікатора. У даній роботі було проведено порівняння методів прискорення класифікації зображень за рахунок використання хеш-кошиків та групи методів оцінювання.

#### 1.4 Застосування моделей нейронних мереж

Нейронні мережі зараз успішно використовують для рішення багатьох задач розпізнавання образів, у тому числі і для класифікації зображень [20, 21]. В даний час найбільш часто в задачах класифікації застосовують класичні архітектури нейронних мереж, такі як багат шаровий перцептрон та згорткові нейронні мережі. Проте класичні нейронні мережі не завжди є ефективними у рамках задачі класифікації:

- зазвичай застосовується ансамбль нейронних мереж (найчастіше 2 або 3 нейронні мережі, навчені з різними початковими значеннями синаптичних коефіцієнтів і порядком надходження образів), що негативно позначається на обчислювальній складності розв'язання задачі і відповідно на часі виконання [21];

- як правило, класичні нейромережеві архітектури використовуються в сукупності з допоміжними методами виділення сюжетної частини зображення (сегментація за кольором, виділення контурів і т.д.), які вимагають якісної і копіткої попередньої обробки навчальних і робочих даних, що є досить трудомістким;

- нейромережеві архітектури є вкрай чутливими до впливу різних зовнішніх чинників (зміни умов зйомки, присутність індивідуальних особливостей на зображенні, зміна орієнтації) [21].

Крім того, на сьогодні існує ряд суттєвих недоліків застосування традиційних нейронних мереж до реальних завдань класифікації зображень. Більшість зображень мають велику розмірність, що впливає на збільшення кількості нейронів вхідного рівня нейронної мережі і, відповідно, розмір самої нейронної мережі значно зростає. Через зростання кількості нейронів вхідного рівня для адекватного навчання системи більша навчальна вибірка, що негативно впливає на часові характеристики процесу навчання, проте не має впливу на власне класифікацію.



Другим недоліком повнозв'язної архітектури є те, що порядок введення не має значення для навчання. Змінні у вхідний шар нейронної мережі подаються без урахування порядку, що часто не є прийнятним для зображень, у яких сусідні пікселі є сильно корельованими один з одним.

На відміну від традиційних архітектур, згорткові нейронні мережі є більш стійкими до більшості перетворень і враховують порядок подачі параметрів. Крім того, у згорткових нейронних мережах за рахунок зменшення розмірності вхідних даних покращуються часові характеристики класифікації. У згорткових нейронних мережах поєднано три архітектурних ідеї для забезпечення інваріантності до більшості афінних перетворень і просторових спотворень.

Для вирішення задачі класифікації часто обирається підхід з використанням нейронних мереж автоасоціативного типу на основі моделі геометричних перетворень. Кожен вхідний вектор у нейронних мережах даного типу розглядається як точка у  $N$  вимірному просторі, де  $N$  – кількість компонент вектора.

Перевагами даного метода у порівнянні з більшістю нейронних мереж, що базуються на концепції біологічних аналогій та ітеративного навчання, є швидке не ітеративне навчання. Кількість епох задається перед початком навчання. Крім того, асоціативний метод працює для навчальних вибірок будь-якого розміру.

Потужним методом розпізнавання зображень за допомогою нейронних мереж зараз є YOLO (You only live once) [22]. Найбільша перевага даної моделі полягає у високій швидкодії класифікації. Швидкодія класифікації досягається за рахунок лише однієї ітерації навчання моделі. У процесі навчання модель YOLO накладає на зображення сітку, розділяючи його на області. У кожній із областей модель намагається передбачити координати зони виявлення з оцінкою впевненості для цих полів і ймовірністю класів. Потім оцінка впевненості для кожної зони виявлення множиться на ймовірність класу, щоб отримати остаточну оцінку [22].

Під час класифікації YOLO розділяє вхідне зображення на сітку  $S \times S$ . У кожній із комірок сформованої сітки проводиться пошук одного об'єкта. На рисунку 1.3 [22] зображено приклад сітки у роботі класифікатора з використанням методу YOLO.

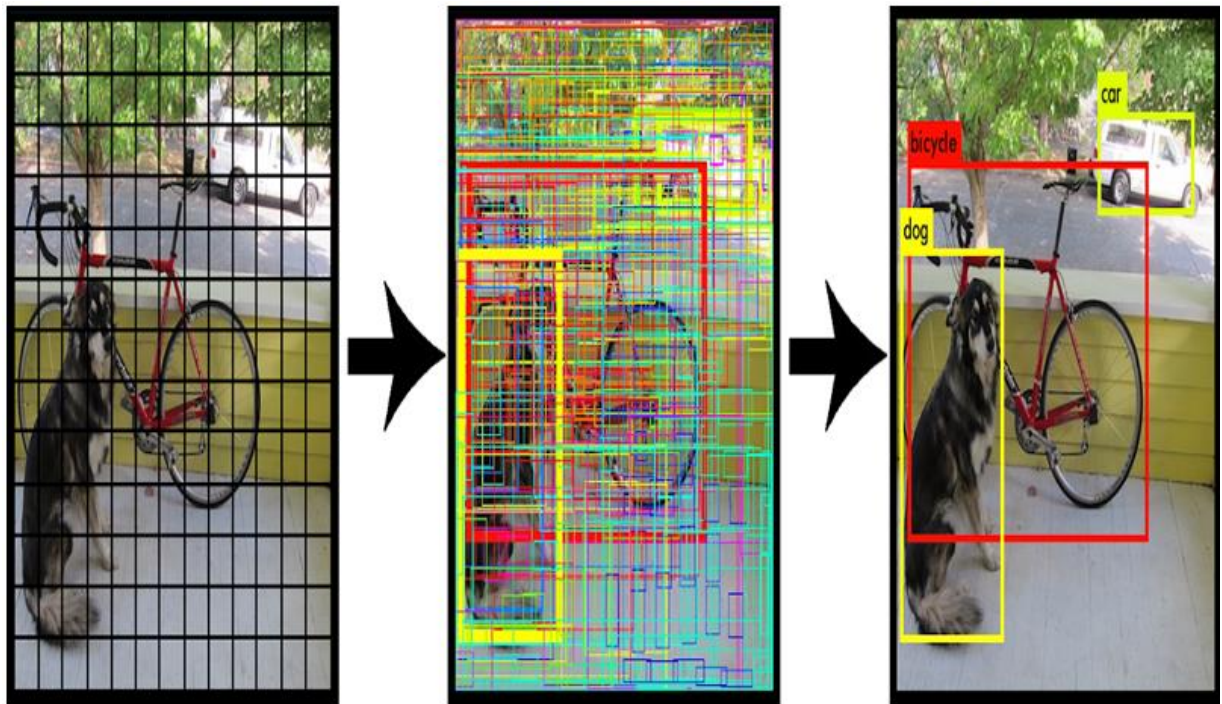


Рисунок 1.3 – Приклад сітки YOLO

Однак функціонування методу з одним об'єктом обмежує близькість виявлених об'єктів. Тому YOLO має деякі обмеження щодо наближеності об'єктів між собою.

Для кожної комірки сітки метод YOLO здійснює такі дії:

- передбачає наявність певної кількості граничних полів, у кожного поля своя шкала достовірності;
- виявляє лише один об'єкт незалежно від кількості вікон;
- прогнозує ймовірності умовного класу (одна на клас для оцінювання ймовірності для класу об'єктів) [22 – 24].

Традиційна нейромережа YOLO має 24 згорткових та 2 повнозв'язних шари. Деякі шари згортки альтернативно використовують шари відновлення  $1 \times 1$  для зменшення глибини карт.

Нейромережа YOLO прогнозує результат у кількох обмежувальних контурах у рамках однієї комірки сітки. Для обчислення втрат необхідно, щоб об'єкт було знайдено в одному з контурів. Для класифікації необхідно обрати контур з найбільшою вірогідністю правильної класифікації. Даний спосіб дозволяє спеціалізувати кожен рамковий прогноз. Результати прогнозування покращуються, коли розміри сторін контуру співвідносяться більш точно.

У якості функції оцінки YOLO використовує помилку між прогнозами в квадраті та основною істиною для обчислення втрат [23]. Функція втрат складається з таких:

- класифікаційні втрати;
- втрата локалізації (помилки між передбачуваним полем границі та основним істиною);
- втрата впевненості (об'єктивність коробки).

Як і більшість алгоритмів класифікації зображень, YOLO має чимало вдосконалених версій, наприклад, Fast (Tiny) YOLO, що відрізняється більшою швидкістю класифікації, проте поступається традиційній версії алгоритму у точності. Fast (Tiny) YOLO має 9 згорткових шарів і використовує більш дрібні карти функцій.

Не дивлячись на те, що YOLO здається сьогодні найкращою технікою для вирішення задачі класифікації зображень, для нього характерні певні недоліки. Оскільки кожна сітка може ідентифікувати лише один елемент, для YOLO виявлення дрібних елементів, що знаходяться на зображенні поряд з ним, є складною задачею. Особливо цей недолік проявляється, коли є необхідність класифікувати велику масу об'єктів вкупі, що призводить до звуження сфери застосування алгоритму. У порівнянні з більшістю інших

нейромереж для класифікації зображень мережа YOLO має дещо меншу точність класифікації.

Завдяки своїй швидкодії та не найбільш високій точності серед решти алгоритмів найбільш доцільно застосовувати YOLO у системах класифікації та розпізнавання зображень в режимі реального часу. Найбільш поширеним прикладом можливого застосування даного алгоритму є класифікація кадрів у відеопотоці.

### 1.5 Постановка задачі дослідження

Класифікація зображень за описом у формі множини дескрипторів ключових точок із використанням засобів для оцінювання відстаней у просторі даних – задача, актуальність якої є сьогодні високою.

Об'єктом дослідження є методи класифікації зображень із використанням засобів для оцінювання відстаней від точки до множини.

Метою є дослідження ефективності та оцінювання швидкодії традиційних та модифікованих структурних методів класифікації зображень. Опис зображення подається у формі множини дескрипторів ключових точок ORB. Треба застосувати і дослідити новітні модифікації методів, що значно прискорюють класифікацію шляхом впровадження процедури оцінювання при встановленні належності дескриптора до класу.

Для досягнення мети необхідно вирішити такі завдання:

- розробити та здійснити програмне моделювання методів класифікації із використанням моделей для оцінювання відстані у векторному просторі даних;
- експериментально проаналізувати та порівняти ефективність класифікації та швидкодію роботи з методами традиційного лінійного пошуку та використанням хеш-функцій;
- зробити висновки про результативність роботи розроблених методів.

## 2 РОЗРОБЛЕННЯ МЕТОДІВ КЛАСИФІКАЦІЇ З ВИКОРИСТАННЯМ ОЦІНЮВАННЯ

У загальному випадку клас візуального об'єкта визначається як нескінченна множина зображень об'єкту з урахуванням його допустимих геометричних перетворень: зсувів, поворотів, масштабування в межах поля зору комп'ютерної системи [1]. Представником класу для класифікатора є еталонне зображення, за яким і визначаємо образ (множину ознак) еталона. Набір обраних користувачем еталонних образів створює скінченну базу зразків. У рамках даних бази еталонів здійснюється класифікація об'єктів, що полягає у віднесенні вхідного зображення до одного із еталонних класів або у відмові від класифікації у разі недостатнього для прийняття рішення рівня релевантності з еталонами [1 – 4].

### 2.1 Реалізація методу класифікації

Візьмемо до розгляду простір  $B^n$  багатовимірних бінарних векторів розмірністю  $n$ . Для опису кожного з еталонів створимо опис у вигляді  $Z \subset B^n$  – один з векторів, які утворюють опис об'єкта. Даний опис є одним з елементів простору множин дескрипторів КТ  $E_i = \{e_v(i)\}_{v=1}^s$ ,  $s = \text{card } E_i$  – число дескрипторів у множині [1].

Для класифікації необхідно створити певну базу  $E$  описів еталонних зображень розмірністю:  $N : E = \{E_1, E_2, \dots, E_N\}$ . Кожний еталонний опис  $E_i$  представляє собою певний клас, що складається із сукупності бінарних дескрипторів КТ. Для розуміння процесу віднесення до певного класу певного опису  $Z \subset B^n$ ,  $Z = \{z_w\}_{w=1}^s$  розглянемо його в рамках процесу класифікації.

Задача класифікації за допомогою класифікатора  $K$  зводиться до встановлення відображення  $K : Z \rightarrow [1, 2, \dots, N]$  на основі вхідних даних. В контексті даної задачі класифікація  $K$  представляє собою процес, що складається з двох етапів  $K = K_2 K_1$ , де перший етап  $K_1 : B^n \rightarrow [1, 2, \dots, N]$  – це визначення класу  $d_w$  для кожного дескриптора  $z_w \in Z$ , другий етап  $K_2 : D \rightarrow [1, 2, \dots, N]$  – формування відповіді класифікатора про клас вхідного об'єкта  $Z$  на основі отриманих локальних рішень із множини  $D = \{d_w\}_{w=1}^s$ . Такий принцип оброблення узагальнює множину рішень однорідних локальних класифікаторів [2, 14]. Фактично, тут  $K_1$  можна розглядати як багатозначну характеристичну функцію для визначення еталонного класу окремого дескриптора із опису розпізнаваного об'єкта.

Етапи  $K_1$ ,  $K_2$  можна побудувати і в інших варіантах, наприклад, шляхом створення на етапі  $K_1$  деякого ймовірнісного розподілу за класами еталонів, включаючи процедури логічного оброблення таких розподілів [3].

Конструктивно реалізація  $K$  спирається на апріорні дані наявної бази  $E$ , так як належність усіх  $e_v(i)$  до відповідного образу  $E_i$  всередині бази уже відома на початку класифікації.

При проведенні етапу класифікації  $K_1$  методом лінійного пошуку (повного перебору) необхідно почергово проаналізувати всі елементи множини  $E$  за формулою

$$d_w = \arg \min_{i, v} \rho(z_w, e_v(i)), \quad (2.1)$$

де  $d_w$  – номер еталону  $E_i$ , до якого буде віднесено дескриптор  $z_w$  об'єкту;

$\rho(z_w, e_v(i))$  – метрика у векторному просторі.

Задля результативної класифікації необхідно проводити логічну обробку значення отриманого за (2.1) мінімуму. Одним з варіантів логічної обробки є впровадження значення порогу. Тобто, вважаємо, що значення  $d_w$

можливо визначити лише в тому випадку, якщо мінімальна відстань менша ніж встановлений поріг  $\delta_\rho$ :  $\rho_{\min} \leq \delta_\rho$ . У протилежному випадку клас  $d_w$  визначити неможливо.

Загалом необхідну кількість  $Q$  обчислених значень метрики в (2.1) у повній базі еталонів лінійним пошуком можна оцінити значенням параметру  $Q = N \cdot s^2$ , вважаючи обсяги описів еталонів та об'єкта рівноцінними. Для векторів простору  $B^n$  в (2.1) може бути застосована проста в обчислювальному сенсі метрика Хемінга, що підраховує число не співпадаючих бітів для векторів.

На етапі класифікації  $K_2$ , що для кожного  $z_w \in Z$  визначений клас  $d_w$  за правилом (2.1), враховуємо при сумі числа  $r_i$  голосів елементів, віднесених до  $i$ -го класу

$$r_i = \begin{cases} r_i + 1, d_w = i, \\ r_i, d_w \neq i, \end{cases} \quad (2.2)$$

а клас  $i_0$  образу  $Z$  об'єкта визначаємо за максимумом серед числа голосів:

$$i_0 = \arg \max_{i=1, \dots, N} r_i. \quad (2.3)$$

Значення  $\{r_i\}_{i=1}^N$  відображають, скільки голосів було віднесено до кожного класу за елементами із  $Z$ . Вирази (2.2), (2.3) конкретизують етап  $K_2$ , суть якого полягає у обробці голосів для компонентів опису  $Z$ . Розглянута двоетапна процедура класифікації базується на основоположному принципі інтелектуального аналізу даних, що полягає у підрахунку числа позитивних рішень (підтримка, рейтинг частоти зустрічей) на аналізованій множині даних [14 – 16].

Розглянуті варіанти побудови класифікатора природно можна трактувати в рамках теорії ансамблевих моделей, де за рахунок створення та агрегування відгуків компонентних класифікаторів (локальних рішень) синтезується «сильний» класифікатор із гарантовано вищою результативністю прийняття рішень. Це відповідає моделі бустінгу [14, 16].

Основним недоліком описаного методу є низькі часові характеристики. Дана проблема виникає через те, що необхідно здійснити повний перебір всіх еталонних даних. Для вирішення даної проблеми у двоетапній процедурі класифікації  $K$  замість повного перебору на етапі  $K_2$  будемо виконувати оцінку відстаней від поточного дескриптора КТ  $z_w$  до кожного з еталонів  $d_i$ . Завдяки описаним змінам очікується зменшення обсягу обчислень  $Q$  у декілька сотень разів.

## 2.2 Застосування апарату хеш-функцій

Метод класифікації з використанням апарату хешування розподіляє знайдені дескриптори КТ за визначеним числом хеш-кошиків. Хеш-функції перетворюють вхідну множину дескрипторів у множину хеш-кошиків, кількість яких задається перед початком роботи [2, 4, 25].

Під час навчання хеш-функція приймає вхідну послідовність дескрипторів КТ як аргумент і повертає номер хеш-кошика – ціле число у діапазоні від 0 до кількості хеш-кошиків. Під час класифікації необхідно для кожного із дескрипторів КТ вхідного зображення спочатку провести процедуру хешування і знайти відповідний хеш-кошик, а далі за допомогою лінійного пошуку всередині встановленого кошику знайти релевантний йому дескриптор.

Описаний метод дозволяє прискорити роботу класифікатора у десятки разів, проте, можливе зниження ефективності класифікації під час роботи з неідеальними зображеннями.



Хешування – перетворення вхідної послідовності даних довільної довжини у вихідну послідовність даних заданої довжини [2, 4]. Перетворення такого типу також називаються хеш-функціями або функціями згортки, а їх результат називають хешем або хеш-кодом [5, 8, 26, 27].

Хешування часто застосовується для порівняння у процедурах пошуку даних. Наприклад, якщо у двох наборів хеш-коди різні, дані гарантовано розрізняються. Якщо ж однакові – імовірність збігу даних висока і їх подальший аналіз має сенс. Проте однозначної відповідності між вихідними даними і хеш-кодом немає через те, що кількість значень хеш-функцій значно менше, ніж варіантів вхідної послідовності. Якщо існує кілька структур даних з різним вмістом, яким відповідають однакові хеш-коди, виникають так звані колізії [8]. При виникненні колізії необхідно на пряму порівнювати структури даних. Імовірність виникнення колізій відіграє важливу роль в оцінці якості хеш-функцій.

На даний час створено велику кількість алгоритмів хешування з різними характеристиками. Найпростішими їх прикладами є контрольна сума або CRC [2] – алгоритм обчислення контрольної суми, що використовується для перевірки цілісності даних. CRC є алгоритмом завадостійкого кодування, що використовує циклічний код для хешування. Для контролю вірності даних використовується контрольна сума.

Хешування є точним методом трансформації даних. Природно зрозуміло, що похибка класифікації із впровадженням хешування може виникнути через те, що всередині хеш-кошика згрупованих даних можуть зустрітися рівноцінні за значенням дескриптори різних класів.

Не у всіх видах хеш-функцій передбачено використання криптографії – забезпечення шифрування даних. Дана технологія використовується лише у певному класі хеш-функцій, що, відповідно, називаються криптографічними. Завдяки даному класу хеш-функцій працюють блокчейн технології та інші види розподілених систем, що забезпечують безпеку та цілісність даних.

Для хеш-функцій властива детермінованість, тобто кінцевий результат (хеш) не змінюється для одних і тих же даних. Відображення хеш-функцій не є оберненими, тобто неможливо, або дуже складно отримати вхідні дані, маючи результат хешування. Чим складніше знайти вхідне значення, тим безпечнішим вважається алгоритм хешування [8, 28 – 30].

Враховуючи те, що дані нечислового типу використовуються доволі часто, виникає необхідність їх коректної обробки. Найчастіше має місце застосування простих правил для перетворення вхідних даних нечислового типу до числового. Наприклад, якщо ключ хешування – символічний рядок, то можна представити кожен його символ у вигляді числового коду [4]. Потім перед математичними операціями числові коди символів складаються. На наступному етапі проводимо конкатенацію всіх наявних кодів символів та виконуємо процедуру хешування за обраним алгоритмом, як з числовими даними.

Враховуючи те, що дані у комп'ютерних системах представлені у вигляді бінарного коду, описані вище дії можна виконати з будь-якими даними.

Спектр застосування хеш-функцій є доволі широким, від пошуку у базах даних до забезпечення безпеки і цілісності даних за рахунок криптографічних хеш-функцій.

Проте, найбільш ефективно хешування для роботи з великими об'ємами даних. Наприклад, можна пропустити великий файл або набір даних через хеш-функцію, а потім використати результат для швидкої перевірки точності та цілісності даних. Дана можливість зумовлена тим, що хеш-функції є детермінованими: вхід завжди буде приводити до спрощеного стиснутого виходу (хеш) [30 – 33]. Завдяки цьому відпадає необхідність у зберіганні великих обсягів даних для забезпечення контролю їх цілісності, що є основою блокчейн технології.

Іншою сферою застосування хеш-функцій є прискорення засобів пошуку даних [4, 12, 15, 33 – 35]. Для того, щоб дані відповідали описаним

характеристикам, перед зберіганням (транспортуванням) проводиться процедура хешування.

Найбільш популярними алгоритмами хешування є CRC32 – алгоритм знаходження контрольної суми [36 – 40], MD6 – побудований на основі дерева Меркла – деревоподібної структури даних, створеної для хешування та алгоритми лінійки SHA (основні SHA256, SHA384, SHA512, SHA224 та SHA384) – алгоритм на основі структури Меркла-Дамгора – метода побудови хеш-функцій, що передбачає розбиття вхідних повідомлень довільної довжини на блоки фіксованої довжини [40 – 42].

За типом процедури обробки даних виділяють декілька глобальних типів хеш-функцій, найбільш популярними з яких є хеш-функції на основі множення (2.4) та на основі ділення (2.5).

Хеш-функції на основі множення мають більш високу швидкодію. Цей підхід перетворює арифметичну прогресію у арифметичну прогресію хеш-значень.

$$h(K) = [M * (C * K)], \quad (2.4)$$

де  $K$  – вхідна послідовність;

$C$  – константа від 0 до 1, що відповідає за кількість хеш-кошиків;

$M$  – константа, підібрана таким чином, щоб не виходити за межі хеш-таблиці.

Хеш-функції на основі ділення є більш простими в реалізації, проте мають гіршу швидкодію:

$$h(K) = K \bmod M, \quad (2.5)$$

де  $K$  – вхідна послідовність;

$M$  – константа, підібрана таким чином, щоб не виходити за межі хеш-таблиці.

У якості хеш-функції для прискорення роботи класифікатора та порівняння з іншими методами прискорення було обрано один із найпростіших у обчисленні хешів – кількість одиничних бітів у бінарному дескрипторі

$$c_i^l = \begin{cases} c_i^l + 1, z_{ij} = 1 \\ c_i^l, z_{ij} = 0 \end{cases} \quad i = \overline{0, N}, j = \overline{0, l}, \quad (2.6)$$

де  $c_i^l$  – поточна кількість одиничних бітів дескриптора КТ;

$z_{ij}$  – поточний біт дескриптора КТ;

$N$  – кількість дескрипторів КТ еталону;

$l$  – довжина дескриптора КТ.

Після підрахунку кількості одиничних бітів за (2.6) дескриптори розподіляються за хеш-кошиками. Кількість хеш-кошиків є параметром, придатним для регулювання, тобто дана хеш-функція надає можливість розподілити дескриптори за змінним діапазоном кількості одиничних бітів, що припадає на один хеш-кошик.

Хешування дозволяє знизити розмірність даних завдяки побудові деякої індексованої структури даних для множин еталонних дескрипторів перед навчанням системи.

За рахунок використання апарату хеш-функцій можливо суттєво скоротити час класифікації зображень (у десятки разів). Проте відбувається збільшення необхідного об'єму пам'яті [5 – 8].

### 2.3 Метод оцінювання відстані до класу

Метод оцінювання дає можливість визначити відстань між множинами дескрипторів ключових точок еталонного та вхідного зображень за декількома обраними дескрипторами ключових точок із усієї множини [12].

Для правильної роботи методу оцінювання важливим є вибір дескриптора ключової точки у складі опису еталону. Наприклад, можна обрати дескриптор випадкової ключової точки із множини ключових точок еталону, або середньої точки класу. Проте, найбільш логічним було б обрати дескриптор точки-медоїда, тобто точки, сумарна відстань Хемінга від дескриптора якої до решти дескрипторів є мінімальною [12, 43].

Загалом даний метод дає суттєвий вигреш у часі (у сотні разів) за рахунок того, що необхідно провести лише один цикл порівняння дескрипторів ключових точок вхідного зображення з медоїдом еталону замість повного перебору двох множин дескрипторів за принципом «кожний з кожним». Проте точність класифікації може різко знизитися за рахунок можливого перетину класів. Для підвищення точності можна використовувати ще одну точку з множини дескрипторів кожного еталону.

Група методів оцінювання використовує певну множину точок кожного еталону як основу для порівняння даного класу з дескрипторами вхідного зображення.

Загальним чином методи даної групи можна описати так:

Крок 1. Для порівняння дескрипторів з вибраною точкою обчислюється відстань Хемінга  $\rho(z, E_i)$  від кожного дескриптора  $z \in Z$  вхідного зображення до кожного з еталонів [12, 44]. На підставі мінімізації отриманої відстані визначається клас еталону, до якого найбільш ймовірно відноситься кожний із дескрипторів вхідного зображення.

Крок 2. Підрахунок акумульованого числа голосів для наборів дескрипторів, віднесених до кожного із еталонів  $E_i$ .

Крок 3. На підставі Кроку 2 здійснюється оптимізація числа голосів на множині класів  $i \in \{1, 2, \dots, N\}$  і визначається клас аналізованого об'єкту [43, 44].

У групі методів оцінок ключовим фактором збільшення швидкодії роботи класифікатора є заміна лінійного пошуку з необхідністю повного перебору на одиночну операцію порівняння поточного дескриптора  $z \in Z$  з

обраною точкою кожного із  $N$  класів шляхом обчислення відстані Хемінга  $\rho$  від елемента  $z \in Z$  об'єкта до класу еталону  $E_i$

$$\rho(z, E_i) = \min_{v=1, \dots, s} \rho(z, e_v(i)). \quad (2.7)$$

Для наочного представлення правила(2.7) використаємо зображення трикутника на площині [43, 44]. Нехай  $a, b, c$  – довжини сторін (рис. 2.1). Тоді за правилом сторони трикутника маємо:

$$c - b \leq a \leq c + b. \quad (2.8)$$

Виникає необхідність переробити формулу (2.8) для застосування у якості виразу для оцінювання для дескрипторів КТ. Нехай  $e^*(i) \in E_i$  – деяка фіксована точка множини  $E_i$ , тоді  $d(i) \in E_i^*$  – точка множини  $E_i^* = E_i / e^*(i)$ . Виходячи з цього відстань  $a_i = \rho(z, d(i))$  від точки  $z$  об'єкта до точки  $d(i)$  множини можна оцінити через обчислену відстань  $b_i = \rho(z, e^*(i))$  та відстань, визначену на попередньому етапі  $c_i = \rho(d(i), e^*(i))$  наступним способом (рис. 2.1):

$$\rho(d(i), e^*(i)) - \rho(z, e^*(i)) \leq \rho(z, d(i)) \leq \rho(d(i), e^*(i)) + \rho(z, e^*(i)). \quad (2.9)$$

Введемо позначення:  $c_{i, \min} = \min_{d(i) \in E_i^*} \rho(d(i), e^*(i))$ ,  $c_{i, \max} = \max_{d(i) \in E_i^*} \rho(d(i), e^*(i))$ .

Оцінку отримуємо за рахунок максимізації лівої частини нерівності (2.9) або мінімізації правої частини нерівності (2.9). Результируюча оцінка описана наступною формулою:

$$c_{i, \max} - b_i \leq \rho(z, E_i) \leq c_{i, \min} + b_i. \quad (2.10)$$

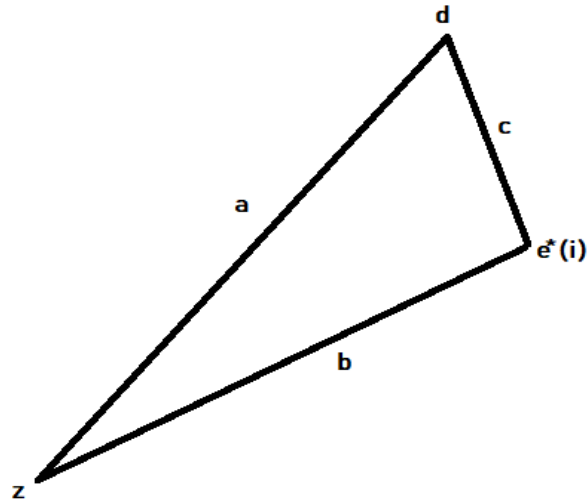


Рисунок 2.1 – Візуалізація нерівності трикутника

Сформована оцінка (2.10) параметрично залежить як від складу множини  $E_i$ , так і від обраної точки  $e^*(i) \in E_i$ . Оцінка за (2.10) – це деяке числове значення  $\rho_i$  для відстані (2.10) від поточного дескриптора до  $i$ -го класу.

$c_{i,\min}$ ,  $c_{i,\max}$  для (2.10) розраховуються під час навчання системи, тобто їх обчислення не впливає на швидкість класифікації. Для того, щоб знайти поточну оцінку  $\rho_i$ , необхідно обчислити відстань  $b_i$  від поточного дескриптора КТ до точки  $e^*(i) \in E_i$ .

Існують варіанти для оцінки відстані від поточного дескриптора КТ до еталонів за правою частиною нерівності (2.10)

$$k = \min_{i=1,\dots,N} (c_{i,\min} + b_i). \quad (2.11)$$

Другий спосіб оцінки відстані – за лівою частиною нерівності (2.10)

$$k = \max_{i=1,\dots,N} (c_{i,\max} - b_i). \quad (2.12)$$

Зазвичай (2.12) застосовують у випадку виникнення неоднозначності при класифікації за (2.11). У випадку, коли обидві нерівності не надають однозначної класифікації, відносимо поточний дескриптор до будь якого з визначених класів [12, 43, 44]. За рахунок цього отримуємо точність кращу, ніж при оцінці відстаней за однією точкою (медоїдом) без суттєвого зниження швидкодії.

Використання оцінки виду (2.9) з її моделями використання (2.10) – (2.12) дає можливість суттєво скоротити час класифікації за рахунок уникнення лінійного пошуку. Для визначення обраної точки множини можна використати, наприклад, інформацію із матриці внутрішніх відстаней, що містить усі попарні відстані між елементами множини [43 – 45].

Оцінювання і результат його застосування у вигляді (2.10) можна узагальнити наступними методами:

- використання замість однієї ключової точки  $e^*(i) \in E_i$  для множини  $E_i$  двох і більше точок множини;
- використання для однієї точки  $e^*(i) \in E_i$  двох і більше найближчих до неї точок множини  $E_i$ ;
- застосування додаткових логічних правил для підвищення достовірності оцінки при віднесенні дескриптора об'єкту до еталону. Такі правила можуть базуватися на інших параметрах множини, таких як діаметр, найбільш віддалена точка та ін. [12].

Запропоновані підходи дозволяють покращити результат класичного методу оцінки за одним дескриптором КТ з множини дескрипторів. Проте, застосування описаних підходів ускладнює класифікацію та призводить до незначного погіршення часових характеристик класифікатора через необхідність врахування  $c_{i,\min}$  та  $c_{i,\max}$ .

Звернемо увагу на те, що дескриптори можуть бути схожими і, у деяких випадках, навіть еквівалентними між собою як у рамках однієї множини дескрипторів КТ класу, так і для різних множин дескрипторів КТ



класів. Дану проблему може вирішити використання матриці відстаней. Дескриптори КТ, відстань між якими менше за певний встановлений поріг, вносяться в матрицю відстаней з деякими коефіцієнтами, що загалом сприяє пришвидшенню класифікації. Аналіз на близькість можна здійснити на етапі класифікації при пошуку медоїда, обчисленні  $c_{i,\min}$  та  $c_{i,\max}$ .

Для більш повного розуміння даного методу класифікації зображень доцільно розглянути розрахунковий приклад. У якості прикладу візьмемо 3 множини дескрипторів по 5 дескрипторів кожна (табл. 2.1) і проведемо описані дії для методу оцінювання за (2.11).

Таблиця 2.1. Вхідні дані для 3-х еталонів

Дескриптори і медоїд еталону E1					Дескриптори і медоїд еталону E2					Дескриптори і медоїд еталону E3				
$d_1(1)$	$d_2(1)$	$d_3(1)$	$e^*(1)$	$d_5(1)$	$d_1(2)$	$d_2(2)$	$d_3(2)$	$d_4(2)$	$e^*(2)$	$d_1(3)$	$e^*(3)$	$d_3(3)$	$d_4(3)$	$d_5(3)$
1	1	1	1	1	0	0	0	0	0	0	0	1	0	0
0	0	0	0	1	0	1	0	1	1	0	1	1	1	1
1	1	0	0	1	0	0	1	1	0	0	0	0	0	1
1	0	1	0	0	1	1	1	0	1	1	1	1	0	1
0	0	1	1	1	1	1	0	1	1	0	0	0	0	0
1	1	0	1	1	1	1	0	1	1	0	1	1	1	0
1	1	0	1	1	1	0	1	0	1	1	0	1	0	0
1	0	0	1	1	0	0	0	0	0	1	1	0	1	0

Після виконання всіх етапів класифікації за методом оцінювання отримуємо наступні кількості дескрипторів, віднесені до кожної з множин еталонів (табл. 2.2).

Таблиця 2.2 – Результати голосування

	Номер еталону		
	1	2	3
1	4	1	0
2	0	5	0
3	0	1	4

Слід зазначити, що рішення стосовно першого дескриптора з першого еталону було неоднозначним (оцінки за (2.11) для першого і третього еталонів були однаковими) тому було застосовано (2.12). Отже, на даному розрахунковому прикладі метод оцінювання показав працездатність, що надалі необхідно підтвердити експериментальним шляхом на реальних зображеннях.

Завдяки використанню у методах оцінювання лише однієї точки (або двох, як у методі з максимумом та мінімумом) маємо отримати значний вигреш у часі (приблизно у 600 разів пропорційно обсягу описів у 500 дескрипторів). У результаті застосування методів з групи оцінок можна досягти вигрешу в часі, що значно перевищує вигреш за рахунок хешування (приблизно у 60 разів).

У даній роботі у якості точки для порівняння у кожному еталоні обрано медоїд кожної множини дескрипторів ключових точок еталонів. Основна перевага використання медоїда у тому, що він є точкою найбільш оптимально розташованою по відношенню до інших і, відповідно, надає можливість найбільш точно визначити відстань від дескриптора до еталону. З медоїдом методи з використанням максимальної та мінімальної відстаней до точок даного еталону дають найбільш оптимальні результати.

#### 2.4 Оцінка результативності класифікації

У задачах класифікації зображень для оцінки ефективності розроблених моделей і порівняння різних методів класифікації використовуються певні критерії. Найбільш поширеними критеріями є Accuracy, Precision та Recall [13, 46].

Для розуміння наведених метрик необхідно ввести поняття матриці помилок (confusion matrix) (табл. 2.3).

Таблиця 2.3 – Матриця помилок

	$Y = 1$	$Y = 0$
$X = 1$	True Positive ( $TP$ )	False Positive ( $FP$ )
$X = 0$	False Negative ( $FN$ )	True Negative ( $TN$ )

У даній матриці  $X$  – відповідь методу класифікації для обраного вхідного зображення,  $Y$  – справжня мітка даного класу зображення. Тобто помилки класифікації діляться на два типи: False Negative ( $FN$ ) і False Positive ( $FP$ ).

Найбільш простою метрикою є Ассурасу (2.13), проте через неспроможність працювати з класами, що містять різну кількість об'єктів, на практиці вона рідко застосовується.

$$acc = \frac{TP + TN}{TP + TN + FP + FN}. \quad (2.13)$$

Метриками, що використовуються більш часто, є Precision (2.14) та Recall (2.12).

$$prec = \frac{TP}{TP + FP}, \quad (2.14)$$

$$rec = \frac{TP}{TP + FN}. \quad (2.15)$$

Precision (2.14) показує, яку частину об'єктів, віднесену класифікатором до поточного, дійсно належить до цього класу. Саме з допомогою критерію Precision не відбувається віднесення всіх зображень до одного класу, адже це спричинить ріст False Positive ( $FP$ ) випадків.

Recall (2.15) – визначає частину зображень поточного класу з усіх зображень поточного класу, класифікованих методом класифікації. Критерій якості Recall показує здатність метода визначати конкретний клас взагалі.

Максимальне значення Precision та Recall – 1, що вказує на максимальну результативність класифікатора.

Критерії якості класифікації Precision та Recall не залежать від розмірів класів, тому їх використання можливе на практиці навіть на незбалансованих наборах даних. Проте користуватись двома критеріями незручно. Одним із способів зведення Precision та Recall в один зручний критерій якості роботи класифікатора є  $F$ -міра (2.16), що представляє собою середнє гармонічне Precision та Recall [46]

$$F_{\beta} = \frac{(1 + \beta^2) * prec * rec}{\beta^2 * prec + rec}, \quad (2.16)$$

де  $\beta$  – вага точності у метриці, при  $\beta = 1$  це середнє гармонічне.

$F$ -міра досягає максимуму, якщо Precision та Recall будуть рівні 1, та наближатися до 0, якщо один з аргументів наближається до 0.

### 3 РЕЗУЛЬТАТИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ ТА ДОСЛІДЖЕННЯ МЕТОДІВ

#### 3.1 Обґрунтування вибору програмних засобів

Для порівняння методів класифікації необхідно провести моделювання кожного з методів використовуючи певні програмні засоби. Під програмними засобами у даному контексті мається на увазі сукупність програм та систем оброблення інформації та програмних документів, необхідних для експлуатації цих програм.

Середовищем розробки було обрано PyCharm 2023 із використанням бібліотек OpenCV, Numpy, Pyodbc та мови програмування Python. У якості СУБД було обрано MS SQL Server.

Інтегроване середовище розробки для мови програмування Python PyCharm 2023 [47] було створене компанією JetBrains у 2010 році на основі середовища IntelliJ IDEA. PyCharm працює під операційними системами Windows, Mac OS X і Linux. Перевагами PyCharm є простота роботи з ним, наявність інструментів автодоповнення коду, зневадження та авто пошуку. Крім того PyCharm надає засоби для аналізу коду, графічний зневаджувач, інструмент для запуску юніт-тестів і підтримує веброзробку на Django.

Мова програмування Python [48] є багатоцільовою мовою програмування, що має можливість працювати на багатьох платформах. Дана мова розроблена Гвідо ван Россумом в 1990 році. Завдяки лаконізму мови Python є можливість створювати програми, що значно переважають за компактністю аналоги, написані із використанням інших мов, при цьому несуттєво поступаючись за швидкістю роботи. Програми на Python можна запускати на комп'ютерах, що працюють під різними операційними системами без будь-яких змін вихідного коду та налаштувань, за рахунок кросплатформенності даної мови.

Однією з найбільших переваг Python є набір стандартних бібліотек, що встановлюється разом з мовою програмування і містить готові інструменти для роботи з операційною системою, вебсторінками, базами даних, різними форматами даних, для побудови графічного інтерфейсу програм та інших.

У випадку, якщо необхідна бібліотека не була встановлена під час встановлення мови, її можна легко додати, використовуючи утиліту `pip` для мови програмування Python.

Проте мова програмування Python має і ряд суттєвих недоліків. Найсуттєвіший з них – невисока швидкість виконання коду при запуску. Описаний недолік зумовлений тим, що Python не є компільованою мовою програмування. Код, написаний на Python, спочатку компілюється у внутрішній байт-код, який потім виконується інтерпретатором Python. Зазвичай програми, розроблені з допомогою даної мови програмування, виконуються повільніше ніж розроблені за допомогою мов більш низького рівня, таких як C, C++, або Java.

Втім, розвиток апаратної частини сучасних комп'ютерів надає достатню потужність для того, щоб виконувати програми, розроблені з допомогою мови програмування Python, без помітного для людини зниження швидкості роботи.

Слід зазначити, що популярність Python також зумовлена швидкістю написання програм з її допомогою, що знижує витрати часу розробників. Таким чином зменшення часу розробки компенсує збільшення часу подальшої роботи програмного продукту.

Мову програмування Python можна використати у широкому спектрі задач різного характеру розміру і складності. Виходячи з цього, програми, розроблені за допомогою мови програмування Python, можуть бути як невеликими скриптами, так і складними системами. Проте найбільш широкого застосування мова програмування Python набула у науковій сфері. Завдяки своїй простоті і широкому функціоналу її часто використовують для

проведення різного роду наукових досліджень, розробки штучного інтелекту, систем розпізнавання зображень та комп'ютерного зору.

Мову програмування Python було обрано для даної роботи через те, що вона дозволяє писати компактний код, що добре читається, має широкий набір бібліотек, що спрощують розробку, і виконує програми за прийнятний час.

Окрім того, код, написаний мовою програмування Python, легко розширюється модулями, написаними на C або C++. Такі модулі можуть використовуватися для виконання частин програми, що створюють інтенсивне навантаження на процесор. Одним з таких модулів є бібліотека для роботи із візуальними даними OpenCV.

OpenCV – бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. Реалізована дана бібліотека була у 2006-му році з допомогою мов програмування C та C++. OpenCV дозволяє ефективно та швидко використовувати алгоритми по роботі із візуальними даними при розробці програмного забезпечення з допомогою мов програмування Python, Java, Ruby, Matlab, Lua та ін. Бібліотеку OpenCV можна вільно використовувати в академічних та комерційних цілях.

Популярність бібліотеки OpenCV у сфері класифікації та розпізнавання зображень зумовлена насамперед тим, що вона реалізує більшу частину методів роботи із зображеннями, особливо у сфері розпізнавання зображень. Інша причина її популярності полягає у тому, що OpenCV можна використовувати в інтеграції з більшою частиною мов програмування.

Проте реалізація більшості алгоритмів роботи із зображеннями є водночас і мінусом тому, що це є збитковістю. Якщо необхідно використати якийсь один алгоритм, необхідно встановлювати бібліотеку повністю, що деколи може сприяти встановленню менш досконалих, проте легших бібліотек.

Дана бібліотека була обрана для розробки завдяки наявності реалізованого методу визначення КТ, простоті встановлення та використання.

Бібліотека NumPy – бібліотека з відкритим вихідним кодом для мови програмування Python, що надає функції для виконання складних математичних операцій, у тому числі з багатовимірними даними. Створена 2005 року Тревісом Оліфантом та його командою розробників на основі бібліотек Numeric та NumArray, що є досі працюючими, проте вже застарілими бібліотеками для мови програмування Python.

Розробка даної бібліотеки вирішила більшість недоліків бібліотек Numeric та NumArray таких як повільна обробка великих масивів даних.

На початку просування NumPy був частиною бібліотеки SciPy. Для того, щоб надати можливість іншим проектам використовувати дану бібліотеку її код було включено в окремий пакет.

Вихідний код бібліотеки NumPy є у відкритому доступі та має велику кількість документації, що сприяє частому використанню даної бібліотеки. Крім того NumPy підтримує роботу з багатовимірними масивами у тому числі з матрицями, та надає можливість використовувати готові складні математичні функції.

Необхідність використання NumPy зумовлена тим, що математичні алгоритми, розроблені на інтерпретованих мовах програмування однією з яких є Python, мають набагато більший час виконання на відміну від реалізованих з допомогою компільованих мов програмування. У бібліотеці NumPy наявні реалізації багатьох обчислювальних алгоритмів, оптимізованих для роботи з багатовимірними масивами даних. У результаті виконання математичних алгоритмів значно пришвидшується.

У даній роботі було використано бібліотеку NumPy через те, що вона надає можливість швидко і точно обробляти великі масиви даних, наприклад для автозаповнення зображення.



У якості бібліотеки для роботи з базою даних у даній роботі було обрано Pyodbc. Дана бібліотека надає можливість швидкого з'єднання з базою даних незалежно від її виду. Також дана бібліотека вирізняється простотою встановлення і роботи з нею, що і було ключовими факторами для її використання для даного дослідження.

Microsoft SQL Server – система управління реляційними базами даних (РСУБД), розроблена корпорацією Microsoft 1989 року. Проте, вдосконалення і випуск нових версій триває і досі. Для даного дослідження було використано Microsoft SQL Server 2019.

Перевагами даної СУБД є простота роботи з даними, перевіреність часом, та широкий спектр застосування – від персональних до баз даних масштабу світових корпорацій. Крім того дана СУБД дозволяє швидко та ефективно тестувати та виконувати запити до бази даних та створювати діаграми баз даних.

Для забезпечення популярності Microsoft SQL Server важливим є і те, що дана СУБД легко інтегрується практично з будь-якою мовою програмування, що значно розширює спектр її застосування.

Важливим моментом є те, що Microsoft SQL Server дозволяє легко переносити бази даних з одного пристрою на інший, що в наш час є вкрай важливим.

Проте Microsoft SQL Server має суттєвий недолік – вона не є кросплатформенною, а працює лише на ОС Windows.

Проте враховуючи всі її переваги, саме Microsoft SQL Server було обрано у якості СУБД для даного дослідження.

### 3.2 Особливості здійснення комп'ютерного моделювання

У рамках кваліфікаційної роботи розроблено та порівняно результативність наступних методів класифікації зображень: традиційний

метод класифікації за множиною дескрипторів КТ, метод класифікації за допомогою хешування, метод класифікації за медоїдами та метод оцінювання.

Вхідними зображеннями для даної роботи було обрано зображення брендів автомобілів (рис. 3.1). Розмір зображень масштабовано до  $500 \times 500$  пікселів. Для визначення дескрипторів КТ застосовано детектор ключових точок ORB з дескриптором розмірністю  $n=256$ . Кількість дескрипторів у описі кожного із еталонів складає  $s=500$ .



Рисунок 3.1 – Еталонні зображення брендів Ламборгіні, Феррарі, Масераті, Бентлі, БМВ

На основі кожного із вхідних зображень сформовано окремий клас. Для кожного класу сформовано множину дескрипторів КТ на основі метода ORB (Oriented FAST and rotated BRIEF) по 500 256-бітних дескрипторів кожна [13, 17 – 19]. Ключові точки зображень та їх дескриптори знайдено за допомогою метода з бібліотеки OpenCV `detectAndCompute()`. Для загального уявлення про положення ключових точок на кожному зображенні було виконано візуалізацію КТ за допомогою метода `drawKeypoints()` бібліотеки

OpenCV. На рисунку 3.2 КТ для кожного із еталонних зображень позначено синім кругом. Слід зазначити, що пошук КТ та їх дескрипторів слід виконувати на зображеннях в градаціях сірого, тому перед пошуком КТ було виконано перетворення кожного із зображень в зображення в градаціях сірого з допомогою метода `cvtColor()` бібліотеки OpenCV.



Рисунок 3.2 – Приклади виділених координат ключових точок дескрипторів ORB (Ламборгіні, Феррарі, Масераті, Бентлі, БМВ)

Повний процес знаходження та візуалізації множини КТ зображення приведено на лістингу 3.1.

Лістинг 3.1 Знаходження та візуалізації множини КТ зображення:

```
def greyColorSetting(self, image):
    greyImage = 0
    try:
        greyImage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    except Exception:
        greyImage = image
```

```
return greyImage
```

```
def keyPointsSearching(self, image):
```

```
    greyImage = self.greyColorSetting(image)
```

```
    orb = cv2.ORB_create(nfeatures=500)
```

```
    keypoints, descriptors = orb.detectAndCompute(greyImage, None)
```

```
    finalImage = cv2.drawKeypoints(image, keypoints, greyImage,  
color=(255, 0, 255), flags=0)
```

```
    return keypoints, descriptors, finalImage
```

У результаті виконання коду з лістингу 1 для кожного із вхідних зображень отримано 2500 КТ, 2500 десяткових дескрипторів КТ та 5 зображень із відміченими КТ на них. Для зручності подальших розрахунків формат байтового виду для дескрипторів КТ було переведено у бінарний вигляд.

Для метода класифікації за допомогою хешування необхідно було розраховувати значення хеш-функції кожного з дескрипторів усіх еталонів. У якості хеш-функції обрано кількість одиничних бітів дескриптора КТ. Виходячи з [13] для досягнення оптимальних результатів за часом сформуємо максимальну кількість можливих хеш-кошиків – 257. Тобто, кожній з можливих кількостей одиничних бітів дескриптора від 0 до 256 включно отримує по одному хеш-кошику.

Проте практичний досвід вказує на те, що більшість дескрипторів потрапляють у хеш-кошики, що відповідають кількостям одиничних бітів, що ближче до медіанної можливої їх кількості, тобто кількості, що відповідає середньому значенню довжини дескриптора. У випадку з дескриптором ORB дана кількість становить 128 бітів.

Даний ефект виникає через те, що метод пошуку ключових точок ORB формує дескриптори КТ за описом околів КТ. Відповідно до цього правила отримання дескриптора, який би складався більшою мірою з нульових або

одиночних бітів є малоімовірним. Отримання дескриптора виключно з одних нульових або одиночних бітів є практично неможливим.

Для перевірки було виконано запит до бази даних для перевірки розподілу кількостей дескрипторів за хеш-кошиками. У результаті виявлено, що більшість дескрипторів збережено у хеш-кошиках, що відповідають кількостям одиночних бітів від 105 до 138 (табл. 3.1). Хеш-кошки з кількостями одиночних бітів, ближчих до мінімального і максимального можливих значень, у даному випадку від 0 до 69 і від 172 до 257 відповідно зазвичай залишаються порожніми. Таким чином отримуємо розподіл дескрипторів КТ за хеш-кошиками близький до нормального.

Таблиця 3.1 – Кількісний розподіл дескрипторів бази за хеш-кошиками

Значення хеш-функції	Число дескрипторів
0 – 69	0
70 – 104	166
105 – 138	1500
139 – 171	834
172 – 257	0

Тому для економії пам'яті було вирішено знаходити мінімальне і максимальне значення серед кількостей одиночних бітів дескрипторів, а хеш-кошки з більшими та меншими кількостями не створювати. Основна логіка хешування наведена на лістингу 3.2.

Лістинг 3.2 Хешування дескрипторів:

```
def hash(self):
    self.descriptorsToStrings()
    self.onesCounting()
    min, max = self.minMaxSearching()
    self.hashesCount = max - min + 1
```

```

self.bordersSearching( min, max)
return self.allStringDescriptors, self.oneCounts, self.borders

```

Для методів класифікації за медоїдами та методу оцінок було знайдено медоїди для кожної з множин дескрипторів КТ, відстані, що відділяють найвіддаленіший дескриптор КТ від відповідного медоїда еталону та відстані, що відділяють найближчий дескриптор КТ від відповідного медоїда еталону. Програмний код для пошуку описаних даних наведено на лістингу 3.3.

Лістинг 3.3 Пошук медоїда та відстаней до найближчої та найвіддаленішої точок еталону:

```

def findMedoid(self, descriptors):
    imedoid = -1
    minsumhemmdist = 2147483647
    allmindist = 0
    allmaxdist = 0
    for i in range(len(descriptors)):
        hd = 0
        mindist = 2147483647
        maxdist = 0
        for j in range(len(descriptors)):
            if i != j:
                dist = self.hemmingDistanceCounting(descriptors[i], descriptors[j])
                hd += dist
                if dist < mindist:
                    mindist = dist
                elif dist > maxdist:
                    maxdist = dist
        if hd < minsumhemmdist:

```

```

minsumhemmdist = hd
imedoid = i
allmindist = mindist
allmaxdist = maxdist
return imedoid, allmindist, allmaxdist

```

Після того, як всі дані, отримані під час навчання системи за всіма необхідними методами, були отримані, необхідно записати їх у базу даних з відповідною структурою (рис. 3.3) за допомогою бібліотеки pyodbc.

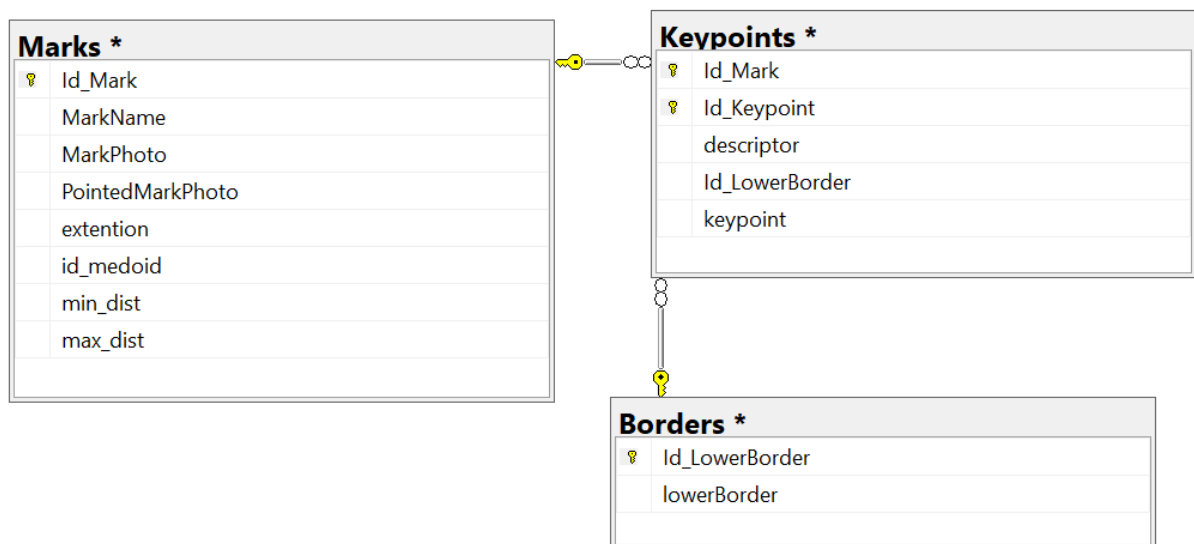


Рисунок 3.3 – Структура бази даних для зберігання даних класифікатора

Після завершення навчання системи необхідно провести її тестування, для того, щоб пересвідчитися у ефективності функціонування кожного з методів класифікації. Для цього реалізуємо частину для тестування кожного з описаних методів. Слід зазначити, що на відміну від функціонального стилю програмування, під час розробки програми для навчання та під час розробки програми для класифікації використано класовий стиль, через те, що він надає можливість більш гнучкої роботи з

різними методами класифікації, що для даної кваліфікаційної роботи є суттєвою перевагою.

Традиційний метод класифікації реалізовано за рахунок лінійного пошуку серед дескрипторів КТ еталонних зображень, найбільш близьких за метрикою Хемінга до кожного з дескрипторів КТ вхідного зображення, запису класів дескрипторів КТ еталонних зображень у список голосів, підрахунку голосів за кожен клас і пошук класу з найбільшою кількістю голосів [13, 17 – 19, 40]. Отриманий клас і є результатом роботи традиційного метода класифікації. Для покращення сформовано відсортований масив значень `mostPossibleDescriptorClasses` для того, щоб бачити, скільки голосів віддано кожному з класів.

Лістинг 3.4 Класифікація зображень традиційним методом:

```
def mostPossibleDescriptorsClassesSearching(self,res):
    self.mostPossibleDescriptorClasses = [0] * len(self.descriptors)
    counter = 0
    for i in range(0,len(self.stringDescriptors)):
        hemmingDistance = self.maxDistance
        for j in range(0,len(res)):
            hD = self.hemmingDistanceCounting(res[j][self.idDescriptor],
self.stringDescriptors[i])
            jd = 0
            if hD < hemmingDistance and hD < self.criticalDistance:
                hemmingDistance = hD
                self.mostPossibleDescriptorClasses[i] = res[j][self.idIdmark]
                jd = j
        if hemmingDistance == self.maxDistance:
            counter += 1

def orderPossibilities(self,carMarks):
```



```

for k in range(0,len(self.possibilities) - 1):
    for i in range(0, len(self.possibilities) - k - 1):
        if self.possibilities[i] < self.possibilities[i + 1]:
            self.possibilities[i],self.possibilities[i + 1] = self.possibilities[i +
1],self.possibilities[i]
            self.classes[i],self.classes[i + 1] = self.classes[i +
1],self.classes[i]
            carMark = carMarks[i]
            carMarks[i] = carMarks[i + 1]
            carMarks[i + 1] = carMark

def mostPossibleClassSearching(self,carMarks,res):
    self.mostPossibleDescriptorsClassesSearching( res)
    for i in range(0,len(carMarks)):
        self.classes.append(carMarks[i][0])
        counter = 0
        for j in range(0,len(self.mostPossibleDescriptorClasses)):
            if self.mostPossibleDescriptorClasses[j] == carMarks[i][0]:
                counter += 1
        self.possibilities.append( counter )
    self.orderPossibilities(carMarks)
    if self.possibilities[0] >= self.criticalMarkIds:
        self.isRecognized = True

```

Метод класифікації з використанням хешування замість лінійного пошуку здійснює пошук серед сформованих під час навчання хеш-кошиків за кількостями одиничних бітів кожного з дескрипторів вхідного зображення, а потім лінійний пошук у рамках кожного з хеш-кошиків. Реалізацію класифікації з використанням хеш-функцій представлено на лістингу 3.5.

Лістинг 3.5 Класифікація з використанням хеш-функцій:

```

self.mostPossibleDescriptorClasses = [0] * len(self.descriptors)
    self.bordersPositionsSearching( res,borders )
    counter = 0
    for i in range(0, len(self.stringDescriptors)):
        self.oneCounts.append(self.onesCounting(i))
        startId = self.idBorderSearching(self.oneCounts[i], borders)
        endId = startId + 1
        startIdBorder = self.bordersPositions[startId]
        endIdBorder = self.bordersPositions[endId]
        hemmingDistance = self.maxDistance
        for j in range(startIdBorder, endIdBorder):
            hD = self.hemmingDistanceCounting(res[j][self.idDescriptor],
self.stringDescriptors[i])
            if hD < hemmingDistance and hD < self.criticalDistance:
                hemmingDistance = hD
                self.mostPossibleDescriptorClasses[i] = res[j][self.idIdmark]
        if hemmingDistance == self.maxDistance:
            counter += 1

```

Навчання класифікатора та класифікацію за значенням медоїдів класів та оцінок можна описати наступною схемою (рис. 3.4).

Суть методів класифікації за медоїдами та оцінок у тому, що кожен з дескрипторів КТ вхідного зображення необхідно порівняти лише з кожним з медоїдів еталонів. Різниця полягає в тому, що у методі класифікації за медоїдами необхідно просто знайти мінімальну відстань між поточним дескриптором КТ вхідного зображення і медоїдами множини дескрипторів КТ еталонного зображення.



Рисунок 3.4 – Схема роботи методів класифікації за медоїдами та оцінок

У методі оцінок для покращення якості класифікації додатково враховується відстань до найближчої точки кожного з еталонних зображень. Програмний код для даного процесу наведено на лістингу 3.6. За рахунок додавання відстані до найближчої точки метод оцінювання намагається врахувати можливість близькості дескрипторів КТ одного еталону до медоїда іншого еталону.

Лістинг 3.6 Метод оцінок:

```

def mostPossibleDescriptorsClassesSearchingWithCMin(self, carMarks,
res):
    self.mostPossibleDescriptorClasses = [0] * len(self.descriptors)
    for i in range(0, len(self.stringDescriptors)):
        hemmingDistance = 1000000
        for j in range(0, len(carMarks)):
  
```

```

        hD = self.hemmingDistanceCounting(res[j] * 500 +
carMarks[j][self.idIMedoid]][self.idDescriptor], self.stringDescriptors[i]) \
        + carMarks[j][self.idMinDistance]
        print(j, hD, end=" ")
        if hD < hemmingDistance:
            hemmingDistance = hD
            self.mostPossibleDescriptorClasses[i] =
carMarks[j][self.idMark]
            print("min= ", hemmingDistance, "mostPossibleClass= ",
self.mostPossibleDescriptorClasses[i])

```

Для того, щоб провести класифікацію, спочатку необхідно навчити систему. Для цього потрібно виконати наступну послідовність дій:

Крок 1. Запустити програму для навчання системи класифікаторів.

Крок 2. У файловому діалозі, що з'являється обрати файл зображення, який необхідно включити у множину еталонних зображень.

Крок 3. У консолі ввести назву класу.

Крок 5. Дочекатися завершення навчання моделі.

Крок 6. Буде показано 2 вікна, в першому вихідне зображення, в другому зображення із позначеними КТ. Для закриття вікон натиснути будь яку клавішу.

Крок 7. Програма завершила виконання. Для додання ще одного зображення виконати Крок 1. В противному випадку можна переходити до тестування системи.

Тестування системи передбачає виконання наступних дій:

Крок 1. Запустити програму для тестування системи класифікаторів.

Крок 2. У файловому діалозі, що з'являється обрати файл зображення, яке необхідно класифікувати. Важливо обирати серед тих, зображень, які були використані при навчанні, через те, що система, поки що, може працювати лише з ними.

Крок 3. Відбувається класифікація зображення за методом медоїдів.

Крок 4. Буде показано 4 вікна, в першому вихідне зображення, в другому вихідне зображення із позначеними КТ, в третьому зображення еталону, до якого було віднесено вихідне зображення і в четвертому зображення еталону, до якого було віднесено вихідне зображення з позначеними КТ. Для закриття вікон натиснути будь яку клавішу. У консоль буде виведено назву метода, за допомогою якого проводилась класифікація, результати класифікації час класифікації та клас, до якого було віднесено вихідне зображення (рис. 3.5).

Крок 5. Повторення Крок 1 та Крок 2 для методу оцінювання.

Крок 6. Повторення Крок 1 та Крок 2 для методу класифікації з хешуванням.

Крок 7. Повторення Крок 1 та Крок 2 для класичного методу класифікації. Слід зауважити, що класифікація за допомогою класичного методу займає набагато більше часу ніж класифікація будь яким іншим методом.

```
Метод класифікації за медоїдами  
Час розпізнавання: 0.1  
Результати голосування  
Lamborghini 222 BMW 86 Bentley 75 Maserati 67 Ferrari 50  
Розпізнана марка автомобіля Lamborghini
```

Рисунок 3.5 – Приклад результату для методу класифікації за медоїдами

### 3.3 Результати дослідження та їх аналіз

Для того, щоб порівняти розроблені методи і впевнитися у ефективності кожного з них та визначити їхні часові характеристики, було проведено тестування програмної моделі. У якості тестів було вирішено виконати запуск системи класифікаторів за описаною вище інструкцією для

тестування, подаючи на вхід кожне з еталонних зображень без будь-яких видів шумів чи інших змін для того, щоб отримати максимально точну початкову оцінку.

Важливим під час тестування є правильний підрахунок голосів. Для того, щоб порахувати голоси кожного класу, необхідно у списку визначених класів дескрипторів КТ підрахувати кількість дескрипторів КТ, віднесених до поточного класу. Процес підрахунку голосів можна описати з допомогою формули

$$S_i = \begin{cases} S_i + 1, d_j = i \\ S_i, d_j \neq i \end{cases} \quad i = \overline{1, n}, j = \overline{1, m}, \quad (3.1)$$

де  $S_i$  – поточна сума голосів наданих  $i$ -му класу;

$d_j$  – голос відданий за  $j$ -й дескриптор;

$n$  – кількість класів;

$m$  – кількість дескрипторів КТ, всіх еталонів.

Одним з основних параметрів, від яких залежить результативність класифікації, є поріг для значення мінімальної відстані при віднесенні дескриптора до класу на першому етапі класифікації. Для дескрипторів, сформованих за допомогою метода ORB, значеннями порогу можуть бути від 0 до 256.

У процесі експериментів було виявлено, що при зменшенні значення порогу відсікається більша частина «чужих» дескрипторів КТ (інших еталонів), але при цьому немає можливості правильно класифікувати і значну частину «своїх». Зі зростанням порогу навпаки відбувається збільшення числа «своїх» з одночасним збільшенням числа «чужих» дескрипторів КТ.

Описана проблема не має єдиного вірного рішення і може бути вирішена лише експериментальним шляхом [13, 40]. Причому для різних методів оптимальні значення порогу можуть бути різними. Наприклад, у даному дослідженні для метода класифікації з використанням хешування та

класичного методу класифікації найкращі результати досягнуто при значенні порогу у 100 бітів. Проте, навіть таке доволі значне значення порогу відсіює більшість дескрипторів для методу класифікації за медоїдами та методу оцінювання.

Причина такої поведінки класифікатора за цими двома методами, полягає у тому, що у геометричному представленні між класами можливий перетин і дескриптори КТ різних еталонів можуть змішуватися, що призводить до суттєвої результуючої мінімальної відстані.

Провівши експерименти з різними значеннями порогу за мінімальною відстанню між дескрипторами, було вирішено не враховувати значення порогу для методів для методу класифікації за медоїдами та методу оцінювання. Завдяки даному рішенню всі дескриптори, які було виявлено найближчими за цими двома методами, враховуються на другому етапі класифікації, що покращує якість класифікації.

Розглянемо ситуацію, коли на вхід подається одне з еталонних зображень без будь-яких перетворень чи зашумлення. У такому разі класичний метод класифікації здійснює повний перебір і відносить всі дескриптори КТ вхідного зображення до правильного класу. Такі ж результати отримано для наших даних і у методі класифікації з використанням хешування. Це можна пояснити тим, що всередині кошика теж здійснювався лінійний пошук даних.

Після проведення експериментів результати підрахунку голосів класів для аналізованих методів було зведено у таблицю 3.2. Виходячи з отриманих результатів, методи класифікації за медоїдами та оцінок показують дещо гірші результати, ніж класичний метод та метод класифікації з хешуванням на рівні віднесення дескрипторів КТ вхідного зображення до найбільш схожого еталону за відповідним методом.

Таблиця 3.2 – Результати голосування для досліджуваних методів

Істинний клас	Метод	Номер еталону				
		1	2	3	4	5
1	За медоїдами	222	50	67	75	86
	Оцінювання	327	14	15	81	63
2	За медоїдами	102	158	98	110	32
	Оцінювання	137	147	73	79	64
3	За медоїдами	117	74	142	136	31
	Оцінювання	83	20	195	160	42
4	За медоїдами	57	74	126	170	73
	Оцінювання	126	15	75	218	66
5	За медоїдами	89	61	84	78	188
	Оцінювання	151	24	51	100	174

Описаний результат зумовлений тим, що множини дескрипторів КТ еталонів, у графічному представленні, є такими, що взаємно перетинаються і, відповідно, деякі дескриптори (імовірно навіть більша частина) можуть опинитися ближче до медоїда іншого класу. Відповідно, віднесені вони будуть до цього класу. Метод оцінок дещо нівелює можливість виникнення такої ситуації за рахунок додавання  $c_{i,\min}$  до відстані від поточного дескриптора КТ до медоїда еталону. Проте в силу того, що у якості найбільш відповідного еталону обирається  $k = \min_{i=1,\dots,N} (c_{i,\min} + b_i)$  ситуація, коли дескриптор відноситься до невірного класу через геометричне розташування все одно можлива.

Проте зниження якості класифікації спостерігається для усіх методів, коли на вхід подаються не ідеальні еталонні зображення, а певним чином видозмінені. Наприклад, може бути накладено гаусівський шум або обрізано частину зображення [40, 49, 50].



Окрім якості класифікації (у даному випадку кількості дескрипторів КТ віднесених до вірного класу), важливим критерієм при виборі метода класифікації є швидкодія класифікації. Важливість даного критерію полягає у тому, що зазвичай необхідно отримати не лише якомога точніший результат, але і за найменший час. У даній роботі час вимірювався стандартним способом: за допомогою вбудованої Python функції засікався час початку та кінця роботи кожного з методів, а потім від часу кінця роботи методу віднімався час початку роботи методу. Отриманий час роботи у мілісекундах було переведено в секунди для зручності подальшої роботи з ним.

Проводити розрахунок критеріїв якості Precision та Recall для описаних методів немає сенсу, адже всі дескриптори КТ було віднесено до вірних класів. Тобто Precision та Recall для даних методів дорівнюють 1.

У таблиці 3.3 наведено середній час роботи кожного з досліджуваних методів класифікації. Для даної роботи було обрано саме середній час роботи кожного з методів тому, що різниця між часом роботи для різних вхідних еталонних зображень не є суттєвою.

Таблиця 3.3 – Час роботи для методів

Метод	Час (с)
Оцінювання	0,1
За медоїдами	0,1
Хешування	1,0
Лінійний пошук	62,2

Враховуючи об'єми даних, що використовуються для навчання та тестування, для сфери класифікації зображень більш важливим є аналіз

глобальних результатів класифікації, а не кількостей дескрипторів КТ вхідного зображення, віднесених до кожного з еталонів.

Виходячи з того, що метод класифікації за медоїдами та метод оцінок показали практично однакові результати за часом та доволі схожі результати за якістю класифікації, а також загальну схожість їх концепції, надалі доцільно розглядіти їх у парі.

Виходячи з таблиці 3.3 методи, обрані для прискорення класифікації виправдали очікування. Метод класифікації з використанням хешування дозволив скоротити час класифікації у середньому у 62 рази, як і очіувалось. Метод класифікації за медоїдами та метод оцінювання дозволили прискорити процес класифікації у середньому у 620 разів, що також було обґрунтовано в теоретичній частині дослідження.

Проте, покращення часу класифікації викликало погіршення її якості. Якщо класичний метод класифікував вірно всі дескриптори КТ, то метод класифікації за медоїдами і метод оцінювання мають деякі вади.

Для вибору найбільш ефективного методу класифікації зображень важливими є як якість, так і час класифікації. Слід пам'ятати, що для різних задач класифікації різні методи можуть бути більш підходящими. Наприклад, якщо необхідно провести швидкісну класифікацію, не надто дбаючи про точність, можна обрати метод класифікації за медоїдами або метод оцінювання. Час класифікації даних методів значно менший, проте за результативністю класифікації вони дещо поступаються іншим методам. Застосування даних методів можливе у сфері класифікації відео, наприклад в охоронних чи пошукових системах.

За необхідності точної класифікації необхідно використовувати класичний метод. Проте за швидкістю класифікації даний метод значно поступається решті розглянутих в даній роботі методів класифікації. Даний метод можна використати у системі класифікації певного роду об'єктів, що не працює у режимі реального часу. Крім того даний метод є найбільш стійким до різних видів шуму [17, 40, 49].

Метод класифікації з використанням хешування є середнім між класичним і методами класифікації за медоїдами та оцінювання за обома характеристиками. Відповідно, і використаний він може бути як у системі класифікації зображень у режимі реального часу, так і в звичайній системі класифікації. Проте, слід зазначити, що швидкодія роботи даного метода все ж ближча до систем класифікації зображень у режимі реального часу.

Якість класифікації кожного з методів суттєво знижується у випадку, якщо необхідно класифікувати зображення з афінними перетвореннями, з певним видом шуму, або частину зображення. Найбільш стійким до завад є класичний метод класифікації, що надає йому суттєву перевагу при виборі методу. Метод класифікації за медоїдами та метод оцінювання в теорії є найменш стійкими до всіх видів перетворень, через те, що навіть в ідеальній ситуації не відносять всі дескриптори до правильних класів. Цей фактор також слід враховувати під час вибору метода класифікації.

Різниця між методом класифікації за медоїдами і методом оцінювання полягає в тому, що метод за медоїдом є дещо менш стійким до перетину множин дескрипторів у графічному уявленні, тому коли невідомо як у просторі розміщено дескриптори краще використовувати метод оцінювання. Щодо часових показників, то суттєвої різниці між цими двома методами немає і даний фактор можна виключити з тих, які буде враховано при остаточному виборі.

Проте метод класифікації за медоїдами є простішим у реалізації і більш інтуїтивно зрозумілим. Крім того, для нього неважливим є положення одного з дескрипторів КТ, що використовуються у методі оцінювання, таких, як найвіддаленіший від медоїда та найближчий до нього. Тому вибір між цими двома методами є доволі складним, і однозначного рішення прийняти неможливо.

Слід зазначити, що суттєвий вплив на час класифікації за методом з використанням хешування має кількість хеш-кошиків. Виходячи з експериментів проведених у [13] експериментальний час класифікації для

розглянутих описів лінійно зростає зі зменшенням числа кошиків при застосуванні хешування.

Отримана залежність потребує додаткових досліджень, проте є правдивою для більшості зображень.

Таким чином, при необхідності прийняття найбільш швидкісних рішень треба використовувати можливість зменшити кількість хеш-кошиків до мінімуму, що могло б посприяти більш зваженому вибору методу класифікації з меншими втратами якості класифікації за рахунок меншої різниці в часі між методом класифікації зображень за медоїдами або оцінювання та методом класифікації зображень з хешуванням.

Ситуація, коли значна кількість дескрипторів різних класів зосереджується у спільному хеш-кошику є однією з найбільш несприятливих для обсягів часу, необхідних для класифікації. Проте, чим менший розмір кожного з хеш-кошиків, тим менша вірогідність такої ситуації. Для вибраної хеш-функції стосовно досліджуваних описів найкращим вибором виглядає число кошиків 257.

Час класифікації спадає зі збільшенням числа кошиків через те, що зменшуються кількості дескрипторів всередині кошиків, а після знаходження кошика обчислюється відстань Хемінга уже для меншої кількості дескрипторів.

Пошук кошика за значенням хеш-функції відбувається за значно менший проміжок часу у порівнянні з обчисленням метрики для множини дескрипторів всередині кошиків через те, що замість обчислення метрики для масиву даних (фактично робота з двовимірним масивом дескрипторів), необхідно провести пошук числа в одновимірному масиві.

Повертаючись, до задачі порівняння методів класифікації, можна сказати, що кожен з методів класифікації має суттєві переваги та недоліки, що характерно для всіх груп методів обробки зображень [50 – 53]. Проте, найбільш збалансованим є метод з використанням хешування. Він не має таких великих втрат якості як метод класифікації за медоїдами чи метод

оцінок та виконує класифікацію, за час, що не є для людини суттєво помітнішим за час класифікації цих методів. Крім того метод класифікації за медоїдами має середні показники при роботі з видозміненими еталонними зображеннями.

## ВИСНОВКИ

У рамках кваліфікаційної роботи було досліджено методи класифікації зображень за множиною дескрипторів КТ: традиційний метод, з хешуванням, за медоїдами, метод оцінювання. Проведено порівняння роботи цих методів за ефективністю та часом класифікації.

Методи класифікації за медоїдами та оцінок значно скорочують час класифікації. Проте дані методи значно знижують якість класифікації.

Використання методу класифікації з хешуванням є компромісним варіантом між класичним методом та методом за медоїдами і методом оцінок за рахунок його стійкості до різного роду завад і середній швидкості роботи.

Практична значущість роботи полягає у побудові та оцінці швидкісних моделей класифікації у просторі дескрипторів КТ, підтвердженні працездатності запропонованих модифікацій на прикладах зображень, створенні програмних застосунків для впровадження розроблених методів класифікації у системах комп'ютерного зору. Розроблені методи можуть бути застосовані у сферах пошуку зображень у базі даних, охоронній, військовій та у обробці відео.

У перспективі можливим є додаткові дослідження із застосуванням інших точок (окрім найвіддаленішої від медоїда та найближчої до нього) для методу оцінювання для покращення результатів класифікації. Іншою перспективною сферою є дослідження методів із використанням видозмінених зображень. Додатковою перспективною сферою може бути поєднання порівнюваних методів класифікації у ансамбль.

Результати роботи апробовано у вигляді тез доповідей [18, 49].

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Гороховатський, В.О., Гадецька, С.В. (2020) Статистичне оброблення та аналіз даних у структурних методах класифікації зображень (монографія), Харків, ФОП Панов А.Н., 128 с.
2. P. Flach. (2012) Machine learning. The Art and Science of Algorithms that Make Sense of Data. New York, NY, USA: Cambridge University Press.
3. Daradkeh, Y., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., Al-Dhaifallah, M. (2021) Methods of Classification of Images on the Basis of the Values of Statistical Distributions for the Composition of Structural Description Components, *IEEE Access*, 9, pp. 92964-92973.
4. Bai Q., Li S., Yang J., Song Q., Li Z., Zhang X., (2020) Object Detection Recognition and Robot Grasping Based on Machine Learning: A Survey, *IEEE Access*, vol. 8, pp. 181855–181879.
5. Гороховатський, В. О., & Творошенко, І. С. (2022). Аналіз багатовимірних даних за описом у формі множини компонент.
6. 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. С. 93–98.
7. Gorokhovatskiy V.A., Gorokhovatskiy A.V., Peredrii Ye.O. (2018) Hashing of Structural Descriptions at Building of the Class Image Descriptor, Computing of Relevance and Classification of the Visual Objects. pp. 1159–1168.
8. Gorokhovatskyi O., Gorokhovatskyi V., Peredrii O. (2018) Analysis of Application of Cluster Descriptions in Space of Characteristic Image Features. p. 52.
9. 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.

10. ORB feature detector and binary descriptor. URL:[https://scikit-image.org/docs/dev/auto\\_examples/features\\_detection/plot\\_orb](https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_orb) (дата звернення 11.10.2023).
11. A Review of Keypoints' Detection and Feature Description in Image Registration. URL:<https://www.hindawi.com/journals/sp/2021/8509164> (дата звернення 04.11.2023).
12. Chávez, E., & Navarro, G. (2005). A compact space decomposition for effective metric indexing. *Pattern Recognition Letters*, 26(9), 1363-1376.
13. Рибалка М.О. Розроблення класифікаторів зображень із впровадженням засобів хешування даних: кваліфікаційна робота першого (бакалаврського) рівня вищої освіти: 122 Комп'ютерні науки. Харків, 2022. 48 с.
14. Zhang X, Yu F., Karaman S., Chang S.. (2017) Learning Discriminative and Transformation Covariant Local Feature Detectors. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6818-6826.
15. Manning, C.D., Raghavan, P., Schütze, H. (2008) *Introduction to Information Retrieval*, Cambridge, University Press, 528 p.
16. Celik, C., Sakir, H., (2017) Content based image retrieval with sparse representations and local feature descriptors: A comparative study, *Pattern Recognition*, 68, pp. 1-13.
17. Гороховатський, В.О., Власенко, Н.В., Рибалка, М.О. (2021) Застосування засобів хешування даних для прискорення класифікаційних рішень у структурних методах розпізнавання зображень. *Сучасні інформаційні системи*, т. 5, №2, с. 13–20.
18. Гороховатський, В., Власенко, Н., Рибалка, М. (2021) Засоби хешування даних у структурних методах розпізнавання зображень. *Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту (ISDMCI'2021): матеріали Міжн. наук. конф. (24-28 травня 2021 р., с. Залізний Порт)*. Херсон: ФОП Вишемирський В.С., с. 45–47.



19. Гороховатський, В. О., Передрій, О. О., Творошенко, І. С., & Марков, Т. Є. (2023). Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень.
20. Forsyth, D. A., & Ponce, J. (2003). A modern approach. *Computer vision: a modern approach*, 17, 21-48.
21. YOLO: Real-Time Object Detection. URL: <https://pjreddie.com/darknet/yolov2> (дата звернення 24.10.2023).
22. A Comprehensive Guide to Types of Neural Networks. URL: <https://www.digitalvidya.com/blog/types-of-neural-networks/> (дата звернення 24.10.2023).
23. YOLOv3: An Incremental Improvement. URL: <https://pjreddie.com/media/files/papers/YOLOv3.pdf> (дата звернення 24.10.2023).
24. Вступ до YOLO: виявлення об'єктів у реальному часі. URL: <https://hashdork.com/uk/Yolo/> (дата звернення 25.10.2023).
25. Хешування. URL: <https://znaimo.com.ua/> (дата звернення 02.11.2023).
26. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description. *Advances in Electrical and Electronic Engineering*, 21(1), 19-27.
27. Gorokhovatsky, V.A., (2014), *Structural analysis and intellectual data processing in computer vision*, SMIT, Kharkiv, p 316.
28. Путятін Є.П., Гороховатський В.О., Матат О.О. (2006) *Методи та алгоритми комп'ютерного зору: навч. посібник*.
29. Гороховатський В.О., Гадецька С.В., Стяглик Н.І. (2019) Вивчення статистичних властивостей моделі блочного подання для множини дескрипторів ключових точок зображень. *Радіоелектроніка, інформатика, управління*, №2, с. 100–107.

30. Gorokhovatsky V.A. Putyatin Y. P. (2009) Image Likelihood Measures of the Basis of the Set of Conformities. *Telecommunications and Radio Engineering*, 68 (9), p. 763–778.
31. Gorokhovatskyi, V., Rusakova, N., Tvoroshenko, I. (2020) The application of image analysis methods and predicate logic in applied problems of magnetic monitoring. *Telecommunications and Radio Engineering*, 79 (20), pp. 1801-1811.
32. 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.
33. M. A. Ahmad, V. Gorokhovatskyi, I. Tvoroshenko, N. Vlasenko, S. K. Mustafa (2021) The Research of Image Classification Methods Based on the Introducing Cluster Representation Parameters for the Structural Description, *International Journal of Engineering Trends and Technology*, 69(10), pp. 186-192.
34. Gorokhovatskyi V., Putyatin Y., Gorokhovatskyi O, Peredrii O. (2018) Quantization of the Space of Structural Image Features as a Way to Increase Recognition Performance. *The Second IEEE International Conference on DataStream Mining & Processing 21-25 August 2018, Lviv, Ukraine*. pp. 464 – 467.
35. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Al-Dhaifallah M., (2022) Classification of Images Based on a System of Hierarchical Features, *Computers, Materials & Continua*, 72(1), pp. 1785-1797.
36. 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.
37. 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.

38. 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.

39. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40-48,

40. Рибалка, М. (2022) Класифікація зображень з адитивним гаусівським шумом з допомогою метода ключових точок. *Радіоелектроніка і молодь у XXI столітті: матеріали Міжнародного молодіжного форуму (20-22 квітня р., м. Харків)*. Харків: ХНУРЕ.

41. Tvoroshenko, I.S., Gorokhovatsky, V.O. (2019) Modification of the branch and bound method to determine the extremes of membership functions in fuzzy intelligent systems. *Telecommunications and Radio Engineering*, 78 (20), pp. 1857-1868.

42. Scherer R. (2018) *Computer Vision Methods for Fast Image Classification and retrieval*, Cześćochowa, Springer, p 137.

43. 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.

44. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S»*. Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine, pp. 25-27.

45. Gorokhovatskyi V., Gadetska S., Ponomarenko R. (2020) Recognition of Visual Objects Based on Statistical Distributions for Blocks of Structural Description of Image. *Lecture Notes in Computational Intelligence and Decision Making*. Proceedings of the XV International Scientific Conference “Intellectual

Systems of Decision Making and Problems of Computational Intelligence” (ISDMCI’2019), Ukraine, May 21–25, 2019, pp. 501-512.

46. Top Performance Metrics in Machine Learning: A Comprehensive Guide. URL: <https://www.v7labs.com/blog/performance-metrics-in-machine-learning> (дата звернення 01.11.2023).

47. PyCharm. URL: <https://www.jetbrains.com/pycharm/> (дата звернення 25.10.2023).

48. Python. URL: <https://www.python.org/doc/> (дата звернення 25.10.2023).

49. Рибалка, М. (2023) Класифікація зображень за частковим поданням. Радіоелектроніка і молодь у ХХІ столітті: матеріали Міжн. молодіжного форуму (10-12 травня р., м. Харків). Харків: ХНУРЕ.

50. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Tools for fast metric data search in structural methods for image classification, IEEE Access, 10, pp. 124738-124746.

51. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, Сучасні інформаційні системи, 6(3), С. 5-12.

52. 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.

53. 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.