

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)  
Кафедра Інформатики  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

### **РОЗРОБЛЕННЯ КЛАСИФІКАТОРІВ ЗОБРАЖЕНЬ ІЗ ВПРОВАДЖЕННЯМ ЗАСОБІВ ХЕШУВАННЯ ДАНИХ**

(тема)

Виконав:  
студент 4 курсу, групи ІТІНФ-18-2

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)

2022 р.

## Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2022 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Рибалці Михайлу Олексійовичу  
(прізвище, ім'я, по батькові)1. Тема роботи Розроблення класифікаторів зображень із впровадженням засобів хешування даних

затверджена наказом університету від 16 травня 2022 року № 541Ст

2. Термін подання студентом роботи до екзаменаційної комісії 23 травня 2022 р.

3. Вихідні дані до роботи алгоритми визначення ключових ознак зображень, теоретичні відомості про методи класифікації зображень, теоретичні відомості про методи знаходження дескрипторів ключових точок, тестові зображення, середовище розробки PyCharm 2018.

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

1. Огляд розпізнавання зображень за допомогою методу ключових точок.2. Аналіз методу ORB пошуку ключових точок на зображенні.3. Математична модель просторової обробки структурних описів.4. Дослідження застосування апарату хеш-функцій для прискорення класифікації.5. Порівняння якості роботи традиційного методу та методу класифікації з використанням хеш-функцій.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) візуалізація ключових точок, тестові зображення, візуалізація роботи хеш-функцій, дослідження результатів, висновки.

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
Консультант з дотримання діючих стандартів та норм	Доцент Белова Н.В.		

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	18.04.2022	
2	Аналіз завдання, підбір літератури	18.04.22-21.04.22	
3	Аналіз літератури з досліджуваної проблеми	22.04.22-25.04.22	
4	Аналіз технічних і програмних засобів	26.04.22-30.04.22	
5	Розробка методу	01.05.22-14.05.22	
6	Програмна реалізація	15.05.22-23.05.22	
7	Оформлення пояснювальної записки	24.05.22-26.05.22	
8	Перевірка на плагіат	27.05.22	
9	Рецензування	28.05.22	
10	Підготовка презентації та доповіді	29.05.22-30.05.22	
11	Занесення роботи в електронний архів	31.05.22	
12	Попередній захист кваліфікаційної роботи	31.05.22	

Дата видачі завдання 18 квітня 2022 р.

Студент \_\_\_\_\_

(підпис)

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

## РЕФЕРАТ/ ABSTRACT

Пояснювальна записка кваліфікаційної роботи: 52 с., 3 табл., 12 рис., 2 дод., 44 джерела.

РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, КЛЮЧОВА ТОЧКА, ДЕТЕКТОР ORB, ДЕСКРИПТОР, ХЕШ-ФУНКЦІЯ, ХЕШ-КОШИК, ХЕШ-СТРУКТУРА, ВІДСТАНЬ ХЕМІНГА, ШВИДКОДІЯ КЛАСИФІКАЦІЇ.

Об'єктом роботи є методи класифікації зображень.

Метою роботи є розроблення методу класифікації зображень у системах комп'ютерного зору. Опис зображення подається у вигляді множини дескрипторів ключових точок. У якості метода пошуку дескрипторів ключових точок використовувався детектор ORB.

Класифікація здійснюється на підставі обчислення ступеня релевантності описів вхідного зображення та еталонів. Для прискорення обчислень застосовано апарат хешування даних опису.

Досліджено вплив застосування хешування на результат класифікації, проведено порівняння розробленого методу за швидкістю та результативністю із традиційним методом без хешування.

Результати роботи доводять доцільність використання засобів хешування для прискорення процесу класифікації зображень.

IMAGE RECOGNITION, KEY POINT, ORB DETECTOR, DESCRIPTOR, HASH FUNCTION, HASH BASKET, HASH STRUCTURE, KHEMINGDING DISTANCE, CLASSIFICATION SPEED.

The object of the work are methods of image classification.

The aim of the work is to develop a method of classifying images in computer vision systems. The description of the image is presented in the form of a set of descriptors of key points. An ORB detector was used as a method to search for key point descriptors.

The classification is based on the calculation of the degree of relevance of the descriptions of the input image and standards. A description data hashing device has been used to speed up calculations.

The influence of hashing application on the classification result was investigated, the developed method is compared in terms of speed and efficiency with the traditional method without hashing.

The results prove the feasibility of using hashing tools to speed up the image classification process.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ	7
1 Розпізнавання зображень за множиною дескрипторів ключових точок	9
1.1 Формалізація методу класифікації	9
1.2 Застосування ORB-дескрипторів для класифікації	11
1.3 Застосування апарату хеш-функцій	15
1.4 Постановка задачі	21
2 Розроблення швидкісного методу класифікації	23
2.1 Двоетапна класифікація	24
2.2 Індексована структура даних для організації швидкого пошуку	27
2.3 Класифікація на основі хеш-структури	29
3 Результати комп'ютерного моделювання методу класифікації зображень	32
3.1 Огляд програмних засобів	32
3.2 Результати комп'ютерного моделювання	34
Висновки	42
Перелік джерел посилання	44
Додаток А Тестові зображення	49
Додаток Б Найважливіші частини коду програми	51

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

КТ – ключова точка

КЗ – комп'ютерний зір

ORB – Oriented FAST and rotated BRIEF

CRC – Cyclic Redundancy Check (циклічний надлишковий код)

Prec – Precision

Rec – Recall

## ВСТУП

У наш час у широкому спектрі напрямів науки і виробничих галузей часто застосовується оброблення даних за допомогою систем і програмних засобів штучного інтелекту. Одним з найбільш поширених сфер впровадження інтелектуальних засобів є аналіз та оброблення візуальної інформації у системах комп'ютерного зору [1 – 6].

Оброблення візуальної інформації супроводжується аналізом результативності за декількома критеріями, найбільш суттєвим з яких є обробка у реальному часі або з фіксованими часовими обмеженнями. Більш вимогливою у прикладних застосуваннях є аналіз візуальних даних у реальному часі, адже для неї необхідно виконувати всі дії за мінімальний час [4, 5, 7 – 9].

У даній роботі увага більшою мірою приділена програмним засобам розпізнавання зображень, адже саме ця сфера обробки візуальної інформації останнім часом користується особливою популярністю і, відповідно, отримала доволі відчутний поштовх до розвитку [5, 9 – 12]. Її затребуваність головним чином обумовлена створенням і розповсюдженням роботизованих систем, що потребують та аналізують візуальні дані про фізичне оточення для коректної роботи [5 – 8, 13 – 15].

Керуючись наведеними потребами, сучасна наука працює над створенням і вдосконаленням програмно-технічних систем, здатних отримувати інформацію з наданих візуальних даних. Для підвищення якості і швидкості обробки інформації розробляються нові, більш сучасні алгоритми, та вдосконалюється апаратна частина, що використовується для отримання і обробки візуальної інформації і, власне, розпізнавання образів.

Дана робота присвячена дослідженню та реалізації методів та програмних засобів з метою підвищення швидкодії систем розпізнавання зображень з використанням апарату детекторів ключових точок (КТ) шляхом впровадження механізму хеш-функцій [2, 4, 7 – 9].

Апарат ключових точок, що застосовується у системах комп'ютерного зору, ґрунтується на даних про образи візуальних об'єктів у вигляді множини дескрипторів ключових точок [1, 3]. Ключова точка у даному контексті – точка, що певним чином виділяється на фоні оточення. Дескриптор ключової точки – це, як правило, вектор розміром від 64 до 512 [15 – 17] бінарних компонентів, що певним чином характеризує ключову точку, зазвичай шляхом аналізу значень яскравості оточуючих точок. Побудова класифікатора заснована на визначенні ступеня релевантності типу множина-множина між описами розпізнаваного об'єкту та еталону та оптимізації релевантності на наявній множині еталонів.

Розпізнавання зображень з допомогою апарату ключових точок – доволі ефективний метод, що користується значною популярністю. Визначними перевагами даного методу є інваріантність отриманих ознак – дескрипторів до повороту, масштабу та зміщення, нечутливість до зміни ступеня освітленості зображення, можливість розпізнавати зображення за наданим фрагментом, низька чутливість до зміни кута бачення зображення [3, 9 – 11]. Водночас недоліками апарату ключових точок є відсутність інваріантності до повної групи афінних перетворень [14], чутливість до шуму.

Зараз найбільш популярними методами формування дескрипторів ключових точок є SURF, BRISK, FAST, SIFT, BRIEF та ORB. Кожен з наведених методів має певні переваги та недоліки і застосовується в досить вузькій сфері.

У даній роботі проведено дослідження запропонованого методу класифікації зображень на основі множини дескрипторів ключових точок з метою збільшення швидкодії оброблення даних у системах КЗ. Моделювання та експерименти проведені із зображеннями з поворотом, з шумом та фрагментами зображень.

# 1 РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ЗА МНОЖИНОЮ ДЕСКРИПТОРІВ КЛЮЧОВИХ ТОЧОК

У роботах [9, 10] розвинуто наближені методи швидкого метричного пошуку за зразком на підставі створення індексованих структур даних на множині ієрархічних ознак зображень, для яких здійснюється сегментація значень функції яскравості. Спеціалізована індексована структура на підґрунті принципу кластеризації об'ємних масивів багатовимірних даних впроваджена також у розробках професійних систем пошуку [10], де задля покращення продуктивності пошуку компонентів візуальних даних застосована багатоетапна схема кластеризації з моделлю наближеного пошуку релевантних елементів всередині кластеру. Створення та застосування кластерних структур даних як сучасний підхід інтелектуального аналізу дає можливість краще адаптуватися до змісту еталонної інформації, що застосовано при визначенні релевантності візуальних об'єктів у технологіях «мішок слів» [1, 11, 16]. Алгоритмічні та програмно-апаратні засоби швидкого пошуку у об'ємних сховищах даних (наприклад, у вебзастосунках) успішно розвиваються і знаходять впровадження у сучасних прикладних пошукових системах [5, 10, 12, 14].

## 1.1 Формалізація методу класифікації

Дотримуючись традиційного підходу [3, 14, 18] розглянемо математичну постановку задачі класифікації візуального об'єкта на зображенні, представленого описом у просторі дескрипторів КТ. Маємо два стрижневих простори даних: простір сигналів, в якому формуються розпізнавані образи – зображення об'єктів, і простір ознак, де реалізується прийняття рішення про клас образів. Класи об'єктів задані у просторі сигналів, а визначаються і

розрізняються у просторі ознак. На рисунку 1.1 наведено приклад зображення і візуальне відображення кілочками координат КТ, сформованих детектором.



Рисунок 1.1 – Зображення та його множина координат КТ

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  $Z = \{z \mid z \in R^n\}$  – у якості опису образу. Опис – це скінченна множина числових векторів. Кожному  $\omega \in \Omega$  можна поставити у відповідність певний опис  $Z$ .

Для створення повноцінного класифікатора необхідно мати правило, яке множині  $Z$  ознак образу ставить у відповідність номер класу образу. Процес класифікації – це ланцюжок перетворень даних із простору образів у множину класів, має вид

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

Головною задачею процесу (1.1) є побудова вирішального правила  $g$ , завдяки якому стає можливим знайти 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  $\Omega$ .

У традиційній постановці задачі класифікації зображень, де описи представлені сукупностями дескрипторів КТ, зображення зазвичай не враховують, що образи у просторі  $\Omega$ , як і їх описи у просторі  $Z$ , можуть мати непустий перетин.

Базовий метод класифікації полягає в наступному [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 
$$v = \arg \operatorname{opt}_{j=1..M} d_j(Z_j, O)$$
.

## 1.2 Застосування ORB-дескрипторів для класифікації

У загальному випадку клас зображення при класифікації визначається як нескінченна множина зображень з урахуванням його допустимих геометричних перетворень в межах поля зору комп'ютерної системи [3, 4, 19 – 22]. Представником класу для класифікатора є еталонне зображення, за яким і визначаємо множину дескрипторів КТ еталонного зображення [3 – 6].

Останнім часом набули прикладного застосування такі детектори КТ, як ORB [1, 6, 19, 21, 23], що обчислюють дескриптор КТ у вигляді бінарного вектора з розмірністю, кратною ступеню двійки. Бінарне представлення значно прискорює процес зіставлення дескрипторів за рахунок можливості застосування двійкових операцій і відповідно спрощує апаратну реалізацію системи розпізнавання. Крім того, бінарна арифметика дає потенцію застосувати ефективний апарат оброблення двійкових даних і синтезувати нові підходи для визначення подібності у просторі дескрипторів при побудові правил класифікації.

Встановлення відповідності для дескрипторів ORB здійснюється за рахунок обчислення відстані Хемінга, тобто підрахунком числа бітів, що відрізняються у двох бінарних векторах. Відповідні дії зводяться до побітової логічної операції XOR, обчислення якої проводиться більш ефективно ніж класичні математичні операції. Модульність побудови ORB дає можливість комбінувати побудову детектора КТ в поєднанні з довільним способом визначення його дескриптора і навпаки, оптимізуючи бажану продуктивність вирішуваних завдань [3, 23, 24].

Таким чином, основні переваги методу пошуку дескрипторів КТ ORB полягають у забезпеченні високої продуктивності за рахунок спрощення процесу знаходження і аналізу КТ, оброблення і побудови дескрипторів бінарного типу.

Метод ORB (Oriented FAST and Rotated BRIEF) є одним з найбільш часто використовуваних засобів для пошуку КТ та для визначення і обробки їх дескрипторів. На даний момент цей метод є одним із найбільш досконалих завдяки організованому у ньому комбінуванню переваг інших методів та усуненню їхніх недоліків. У якості детектора у даному методі використовується метод FAST, а у якості дескриптора КТ – модель BRIEF [1, 6].

Детектор КТ на базі метода пошуку КТ FAST аналізує 16 пікселів, що оточують піксель  $p$ , що розглядається на поточній ітерації як показано на рисунку 1.2.

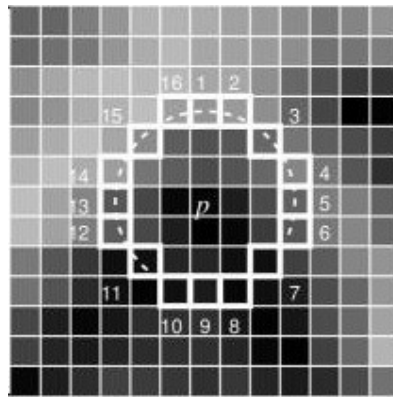


Рисунок 1.2 – Схема аналізу поточного пікселя  $p$  та його оточення

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].

Завдяки незначній кількості пікселів, що перевіряються на етапі прийняття рішення, метод має кращі швидкісні характеристики, ніж більшість інших методів пошуку КТ.

Дескриптор на підставі методу BRIEF має довжину 256 бінарних компонентів (бітів). Отримується 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:

$$D_i = \{ 1, I_x < I_y, 0, I_x \geq I_y, i = \underline{0, n} \}, \quad (1.4)$$

де  $D$  – дескриптор;

$I$  – середня яскравість;

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

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

Метод ORB забезпечує високу прикладну швидкість за рахунок незначного числа пікселів, що аналізуються, і водночас є ефективним за рахунок дієвого алгоритму розрахунку дескрипторів.

У даній роботі було використано готову реалізацію метода ORB з бібліотеки OpenCv [25 – 28]. Бібліотека з відкритим кодом 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.

Дескриптори ключових точок, отримані у результаті навчання, необхідно зберегти до бази даних для подальшого порівняння з дескрипторами, отриманими у процесі розпізнавання з вхідних зображень.

Для попарного порівняння бінарних дескрипторів доцільно використовувати метрику Хемінга. 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.

Метрика Хемінга є найбільш зручною для роботи з бінарними даними, адже є необхідність порівнювати елементи векторів на відповідність, не

зважаючи на величину різниці між елементами вектора. Візуальне представлення принципу обчислення метрики Хемінга зображено на рисунку 1.3. Як можна побачити на рисунку 1.3 різниця при посимвольному порівнянні рядків суміжних вершин дорівнює один символ, відповідно, метрика Хемінга для рядків двох сусідніх вершин дорівнює 1.

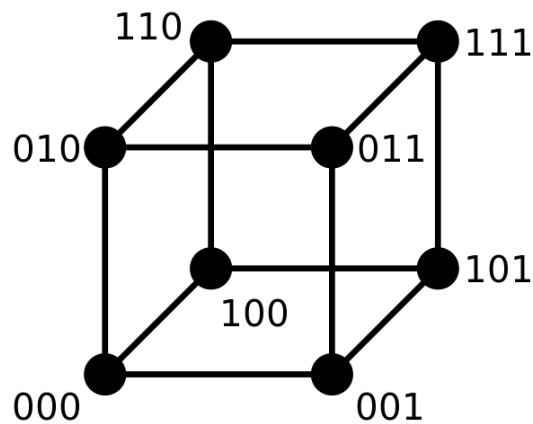


Рисунок 1.3 – Візуалізація принципу обчислення метрики Хемінга на прикладі куба

У класичному варіанті класифікатора необхідно проводити порівняння дескрипторів ключових точок вхідного зображення методом повного перебору за принципом множина-множина [26 – 30], що потребує значних часових ресурсів. Виникає необхідність прискорити роботу класифікатора. У даній роботі було здійснено прискорення класифікації за рахунок впровадження апарату хеш-функцій.

### 1.3 Застосування апарату хеш-функцій

Хешування – перетворення вхідної послідовності даних довільної довжини в вихідну послідовність даних заданої довжини [2, 4, 20]. Перетворення такого типу також називаються хеш-функціями або функціями згортки, а їх результати називають хешем або хеш-кодом [5, 8, 18].

Хеш-функції, що лежать в основі хеш-таблиць, часто застосовуються у алгоритмах науки про дані, де хеш-таблиці часто набувають незвичайної форми. У рамках даній роботи є необхідність розглянути основні принципи роботи з хеш-функціями.

Перш за все, хеш-функція  $h$  приймає вхідну послідовність для хешування як аргумент і повертає номер хеш-кошика – це ціле число, у діапазон від 0 до  $B - 1$ , де  $B$  – кількість осередків [29 – 33]. Тип ключа хешування може бути (зазвичай ціле число, рідше символна послідовність). На інтуїтивному рівні хеш-функція переводить до випадкового вигляду вхідну послідовність за певним законом [33 – 36]. Точніше, якщо вхідна послідовність випадково вибирається з розумної сукупності можливих послідовностей, то  $h$  помістить у кожний із  $B$  хеш-кошиків приблизно однакову кількість вхідних послідовностей. Реалізувати вказане правило було б неможливо, якщо, наприклад, розмір сукупності вхідних послідовностей хешування був менший за  $B$  [4, 15, 36, 37].

Хешування часто застосовується у засобах порівняння даних. Наприклад, якщо у двох структур даних хеш-коди різні, масиви гарантовано розрізняються. Якщо однакові – структури даних, швидше за все, співпадають. У загальному випадку однозначної відповідності між вихідними даними і хеш-кодом немає в силу того, що кількість значень хеш-функцій суттєво менше, ніж варіантів вхідної послідовності. Якщо існує кілька структур даних з різним вмістом, яким відповідають однакові хеш-коди, природно виникають так звані колізії [4]. При виникненні колізії необхідно безпосередньо порівнювати структури даних. Імовірність виникнення колізій відіграє важливу роль в оцінюванні якості хеш-функцій і залежить як від хеш-функції, так і від значень даних.

Розроблено багато алгоритмів хешування з різними характеристиками. Вибір тієї або іншої хеш-функції визначається специфікою розв’язуваної задачі. Найпростішими прикладами хеш-функцій можуть служити контрольна сума або CRC [2].

Хешування відносять до точних методів трансформації даних. Природно зрозуміло, що похибка класифікації із впровадженням хешування може виникнути із-за того, що всередині корзини згрупованих даних можуть зустрітися рівноцінні за значенням дескриптори різних класів.

На рисунку 1.4 зображено загальну концепцію застосування хеш-функцій.



Рисунок 1.4 – Візуалізація роботи хеш-функції

Не всі хеш-функції передбачають використання криптографії – технології забезпечення секретності інформації, а лише ті, які спеціально призначені для цього, так звані криптографічні хеш-функції. Завдяки їх роботі блокчейни та інші розподілені системи здатні досягти високого рівня цілісності даних та безпеки. Як звичайні, так і криптографічні хеш-функції є детермінованими. Це означає те, що доки вхідні дані не змінюються, алгоритм хешування завжди видаватиме той самий результат (також відомий як дайджест або хеш).

Алгоритми хешування розроблені таким чином, що їх функція працює в односторонньому порядку, тобто дані не можуть бути повернені у зворотному порядку без вкладення великої кількості часу та ресурсів для здійснення обчислень. Іншими словами, досить легко отримати результат маючи вхідні дані, але надзвичайно важко здійснити процес у зворотному напрямку. Чим

складніше знайти вхідне значення, тим безпечнішим вважається алгоритм хешування [4, 8].

Виходячи з того, що вхідна послідовність даних доволі часто може бути нечисловою, виникає потреба певним чином обробляти її. Зазвичай використовуються прості правила, що дозволяють перетворити найбільш уживані типи у цілі числа. 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. Далі конкатенуємо коди всіх символів та розглядаємо результат як одне ціле число. Складаємо все, що вийшло таким чином та проводимо хешування за обраним алгоритмом, як зі звичайним цілим числом.

Подібного роду дії можна провести з будь-яким типом даних, тому, що будь-які дані у комп'ютері по суті є послідовністю бінарного коду, який легко уявити у числовому вигляді.

Крім цього, результат роботи хеш-функції є постійним для одних і тих же вхідних даних, а навіть незначна зміна вхідних даних може призвести до отримання зовсім іншого значення хеша. Перетворення даних таким чином відбувається не за рахунок роботи всіх хеш-функцій, але більшість найбільш відомих алгоритмів працюють саме таким чином. Це одночасно і перевага, і недолік хешування. Рисунок 1.5 демонструє кардинальну зміну значення хеш-функції для схожих вхідних даних.

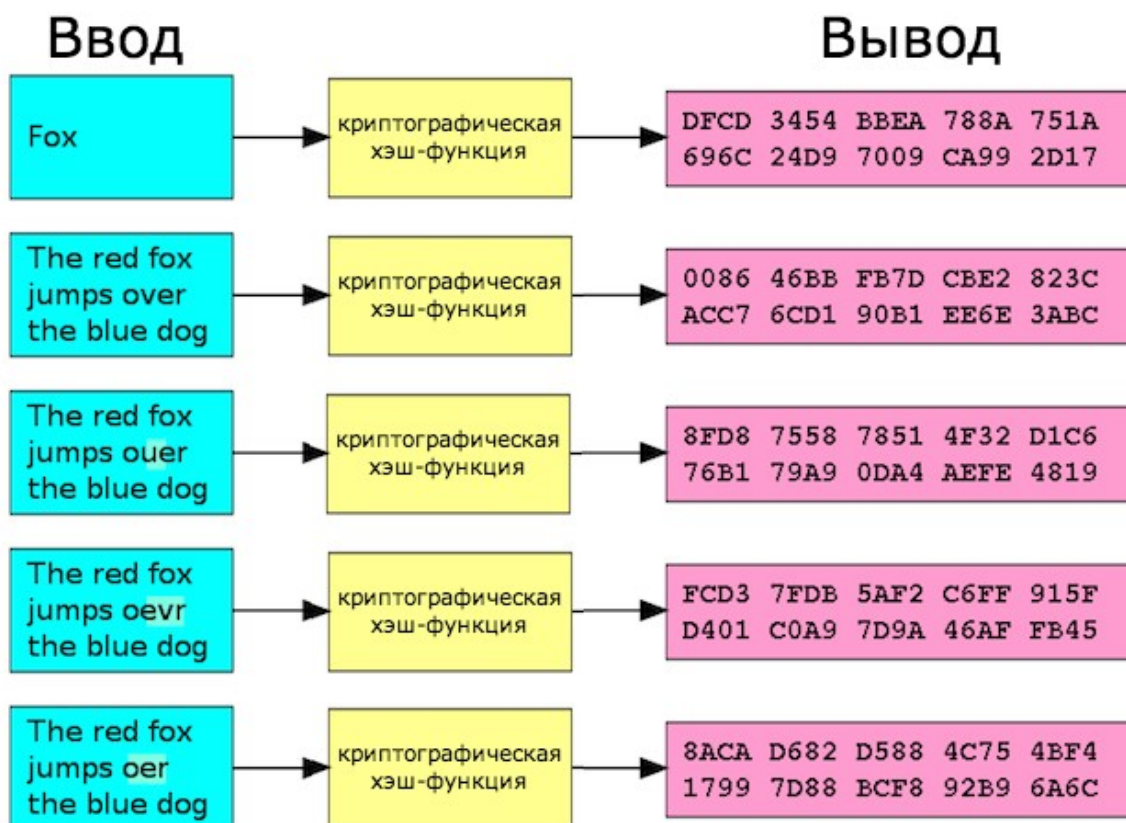


Рисунок 1.5 – Принцип перетворення вхідного тексту довільної довжини до тексту фіксованої довжини за допомогою хеш-функції

Хеш-функції мають широкий спектр варіантів використання, включаючи пошук у базах даних, аналіз великих файлів та управління даними. У свою чергу криптографічні хеш-функції широко використовуються у застосунках, пов'язаних з інформаційною безпекою для аутентифікації повідомлень та цифрової дактилоскопії.

Хешування демонструє весь свій потенціал під час роботи з величезною кількістю інформації. Наприклад, можна пропустити великий файл або набір даних через хеш-функцію, а потім використати результат для швидкої перевірки точності та цілісності даних. Це можливо завдяки детермінованій природі хеш-функцій: вхід завжди буде приводити до спрощеного стиснутого виходу (хеш). Такий метод усуває необхідність зберігати та запам'ятовувати великі обсяги даних. Хешування є особливо корисним для технології блокчейн.

Основною сферою застосування хеш-функцій є забезпечення цілісності та безпеки інформації при її зберіганні та транспортуванні. Додатковою сферою є прискорення засобів пошуку даних [4, 8, 12, 15]. Для забезпечення згаданих характеристик даних перед збереженням або транспортуванням обчислюється значення хеш-функції за певним алгоритмом і зберігаються або транспортується разом з даними. Після зчитування або отримання даних значення хеш-функції обчислюється повторно і порівнюється з початковим значенням хеш-функції.

Найбільш популярними алгоритмами хешування є CRC32 – алгоритм знаходження контрольної суми [35 – 38], 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 [39 – 41].

Також можна виділити глобальні типи хеш-функцій, кожен з яких лягає в основу певного алгоритму. Найбільш розповсюдженими є хеш-функції на основі множення (1.5) та на основі ділення (1.6).

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

$$h(K) = [M * (C * K)], \quad (1.5)$$

де  $K$  – вхідна послідовність;

$C$  – константа в діапазоні 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:

$$h(K) = K \bmod M, \quad (1.6)$$

де  $K$  – вхідна послідовність;

$M$  – певна константа, що визначає кількість можливих хеш-кошиків.

У якості хеш-функції для прискорення роботи класифікатора було обрано кількість одиничних бітів у дескрипторі. Після підрахунку кількості одиничних бітів дескриптори розподіляються по хеш-кошикам. Кількість хеш-кошиків є регульованим параметром, тобто дана хеш-функція надає можливість розподілити дескриптори за змінним діапазоном кількості одиничних бітів, що припадає на один хеш-кошик.

Впровадження засобів хешування даних здатне суттєво скоротити обсяг часових затрат у процесі практичної реалізації процедур пошуку та класифікації. При хешуванні дані бази еталонів попередньо розкладають на групи подібних за деякою ознакою елементів, що значно прискорює процес оброблення за рахунок відповідного зростання вимог до об'ємів пам'яті пристрою [5 – 8].

Хешування реалізує перспективну ідею зниження розмірності шляхом попередньої побудови для еталонної множини деякої індексованої структури даних, що загалом сприяє спрощенню оброблення та знижує обчислювальні затрати на класифікацію.

#### 1.4 Постановка задачі

Класифікація візуальних об'єктів за множиною дескрипторів КТ є актуальною задачею у сфері розпізнавання зображень. Поставлено задачу розроблення методів порівняння з еталоном для вхідних розпізнаваних зображень та отримання прийнятної швидкодії методу за рахунок застосування хеш-функцій.

Об'єктом роботи є методи класифікації зображень.

Метою роботи є розроблення методу класифікації зображень у системах комп'ютерного зору. Опис зображення подається у вигляді множини дескрипторів ключових точок. У якості метода пошуку дескрипторів ключових точок використовувався детектор ORB.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз методів пошуку КТ та обрати найбільш придатний для даної роботи;
- розробити алгоритм зіставлення з еталоном для множин зображень;
- реалізувати алгоритм хешування для прискорення процесу класифікації;
- експериментально перевірити результативність методу класифікації із впровадженням хешування.

## 2 РОЗРОБЛЕННЯ ШВИДКІСНОГО МЕТОДУ КЛАСИФІКАЦІЇ

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 [3]. 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 [3 – 6, 39].

При несуттєвому рівні впливу фону та завад на аналізованому зображенні отримав застосування класифікаційний підхід, де на підставі значень комплектної множини дескрипторів опису формується система інтегрованих характеристик [3]. Зрозуміло, що впровадження інтегрованої системи ознак суттєво скорочує обчислювальні витрати на класифікацію за рахунок принципу узагальнення інформації. Але більш універсальним у аспекті завадостійкості вважається по-елементний аналіз компонентів опису, що на основі формування статистичних характеристик цілісного образу дає можливість у процесі класифікації частково знизити вплив перешкод [4].

Для способу по-елементного аналізу традиційний підхід структурної класифікації зводиться до пошуку для кожного дескриптора із опису розпізнаваного об'єкта деякої підмножини релевантних йому у комплектній множині дескрипторів бази еталонів та підрахунку числа дескрипторів об'єкту, віднесених відповідно до кожного із класів. Зважаючи на те, що число дескрипторів у описі кожного еталону досягає 300...500, а класів може бути десятки і навіть сотні, такий підхід потребує досить громіздких обчислень [3, 5, 7].

Зрозуміло, що для систем комп'ютерного зору ключовим критерієм при впровадженні засобів швидкого пошуку є все-таки гарантоване забезпечення достатніх показників результативності класифікації.

## 2.1 Двоетапна класифікація

Розглянемо простір  $B^n$  багатовимірних бінарних векторів розмірністю  $n$ , у якому будемо конструювати образи  $E_i \subseteq B^n$  розпізнаваного об'єкту і еталонів. Зафіксуємо деяку мультимножину векторів як опис еталону у  $E_i = \{e_v(i)\}_{v=1}^s$  просторі множин дескрипторів КТ,  $s = \text{card } E_i$  – число дескрипторів у множині [3]. Окремі ознаки – це вектори, скінченна множина яких створює опис об'єкту.

Класифікація передбачає наявність деякої бази  $E$  описів еталонних зображень розмірністю:  $N : E = \{E_1, E_2, \dots, E_N\}$ . Кожний еталонний опис  $E_i$  репрезентує окремий клас, представлений скінченною множиною дескрипторів КТ.

Тепер розглянемо довільний опис  $Z \subset B^n$ ,  $Z = \{z_w\}_{w=1}^s$  розпізнаваного об'єкту. Поставимо задачу побудови класифікатора  $K$  як відображення  $K : Z \rightarrow [1, 2, \dots, N]$  на основі попереднього конструювання деякої індексованої структури на множині.

Класифікацію  $K$  представимо у вигляді двоетапного процесу  $K = K_2 K_1$ , де на першому етапі  $K_1 : B^n \rightarrow [1, 2, \dots, N]$  здійснюємо визначення класу  $d_w$  для кожного дескриптора  $z_w \in Z$ , а на другому етапі  $K_2 : D \rightarrow [1, 2, \dots, N]$  із множини  $D = \{d_w\}_{w=1}^s$  отриманих локальних рішень першого етапу формуємо результуючий висновок про клас об'єкту  $Z$ . Такий принцип оброблення узагальнює множину рішень однорідних локальних класифікаторів [2, 14].

Фактично тут можна розглядати як багатозначну характеристичну функцію для визначення еталонного класу щодо 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 [9].

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.

Якщо класифікацію  $K_1$  здійснювати традиційно методом лінійного пошуку (повний перебір) шляхом послідовного аналізу кожного елемента набору  $E$ , то застосовують конкурентне правило

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

де  $d_w$  – номер еталону  $E_i$  до якого буде віднесено дескриптор  $z_w$  об'єкту  $d_w \in \{1, \dots, N\}$ ;

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

Важливим задля результативної класифікації є впровадження логічного аналізу значення мінімуму, що досягнуто при здійсненні (2.1). Значення  $d_w$  визначається тільки тоді, якщо отриманий мінімум відстані не перевищує заданий поріг  $\delta_\rho$ :  $\rho_{\min} \leq \delta_\rho$ . У протилежному випадку клас  $d_w$  не визначається, тобто дескриптор вважається хибним.

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.

$$\rho(z_w, e_v(i)) = \sum_{k=0}^N |z_{wk} - e_{vk}(i)|. \quad (2.2)$$

Етап класифікації  $K_2$  зводиться до того, що на кожному кроці аналізу опису  $Z$  по правилу (2.1) за значенням  $d_w$  інкрементується число  $r_i$  голосів елементів, віднесених до  $i$ -го класу

$$r_i = \begin{cases} r_i + 1, & d_w = i, \\ r_i, & d_w \neq i, \end{cases} \quad (2.3)$$

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

$$i_0 = \arg \max_{i=1, \dots, N} r_i. \quad (2.4)$$

Значення  $\{r_i\}_{i=1}^N$  відображають гістограму класів за числом голосів елементів із  $Z$ . Вирази (2.2), (2.3) конкретизують етап  $K_2$ , що полягає у обробленні 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 зустрічальності) на аналізованій множині даних [14 – 17].

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].

Ключовою обчислювальною проблемою при побудові класифікатора для об'ємних баз даних є реалізація пошуку, що фактично є традиційним підходом ближнього сусіда. Зосередимося на її вирішенні шляхом застосування у двоетапній процедурі класифікації спеціалізованої структури даних, що спрямована на зменшення обсягу обчислень  $Q$ .

## 2.2 Індексована структура даних для організації швидкого пошуку

На етапі попередньої обробки створимо у еталонній множині  $E$  образів спеціалізовану структуру задля забезпечення швидкісної результативної класифікації.

Здійснимо розбиття  $T$  на множині  $E$  дескрипторів бази зображень, отримуємо множину із  $M$  непересічних груп  $T_k(E)$ :

$$E = T(E) = \cup_{k=1}^M T_k(E), T_k(E) \cap T_j(E) = \emptyset. \quad (2.5)$$

Будемо мати на увазі два основних та найбільш поширених способи розбиття: хешування та кластеризацію [3, 15, 41]. Загалом впровадження хешування пов'язують із точними засобами оброблення, а кластеризацію – із наближеними способами самонавчання для вирішення прикладних задач. Трансформація (2.4) здійснює попередню класифікацію і повністю зберігає усю сукупність аналізованих даних, яка тепер розподіляється між групами  $T_k(E)$ . У результаті перетворення (2.4) кожний дескриптор  $e_v \in E$  бази еталонів отримує параметр  $k$  номеру групи (хеш-кошика).

Зважаючи на уже існуюче  $E = \bigcup_{i=1}^N E_i$  розбиття даних на окремі еталонні образи, визначимо величину

$$t_{i,k} = \text{card}\{e_v \mid e_v \in E_i \ \& \ e_v \in T_k\} \quad (2.6)$$

як число дескрипторів  $i$ -го класу, що потрапили до кошика  $T_k$ .

На підставі  $t_{i,k}$  маємо вагову кількісну характеристику  $b_i(T_k)$  для кожного кластеру

$$b_i(T_k) = \frac{t_{i,k}}{\text{card}T_k} \quad (2.7)$$

Вираз (2.6) визначає статистичний розподіл елементів кожного сегменту даних за класами еталонів у вигляді вагових коефіцієнтів класів, причому  $\sum_{i=1}^N b_i = 1$ . Розподіл  $b$  є спільною характеристикою усіх елементів сегменту даних, отриману за результатами аналізу (навчання) для бази  $E$ . На етапі навчання ефективним для підвищення результативності класифікації може бути запровадження логічного оброблення вектору  $b$  задля спрощення аналізу та підсилення впливу вагових коефіцієнтів  $b_i$  для найбільш значущих класів у хеш-кошику [15 – 17].

Конкретно для хешування ознакою окремого хеш-кошику (кластеру) даних виступає значення хеш-функції.

Ще одним важливим параметром, що впливає на швидкодію пошуку, є число кошиків  $M$ . Чим менше число кошиків, тим вища швидкодія переходу до кошику. Чим більше число кошиків, тим менше об'єм даних для аналізу всередині кошиків. Граничними ситуаціями є один кошик ( $M=1$ ) та повна відсутність хешування ( $M=N \cdot s$ ), що відповідають лінійному пошуку. За думкою дослідників можна формулювати задачу оптимізації числа  $M$  кошиків, де критерієм виступає число  $Q$  обчислень метрики для компонентів опису [9, 10].

Якщо в одному кошику в середньому число елементів є  $\frac{N \cdot s}{M}$ , то кількість обчислень (2.1) всередині індексної структури з використанням значень хеш-коду пропорційна  $Q_1 = s \cdot M + \frac{N \cdot s}{M}$ , що значно менше, ніж  $Q_2 = N \cdot s^2$  для традиційного лінійного пошуку. Для конкретних значень  $s = 500$ ,  $N = 10$ ,  $M = 10$ , вираш  $\beta = \frac{Q_1}{Q_2}$  складає приблизно 450 разів і зростає зі збільшенням обсягів  $N, s$  даних.

Задля подальшого прискорення процедури оброблення можна застосувати додаткове упорядкування отриманої структури даних, впровадивши, наприклад, сортування даних всередині кошиків  $T_k$  за деяким ключем  $g(e_v)$ .

Метод оброблення даних з використанням хешування, який фактично здійснює попередню класифікацію на окремі кошики та має свої особливості. Число кошиків для цього методу, як правило, фіксоване і пов'язане із діапазоном значень хеш-ключа  $u(e_v)$  для хешування, яке задає користувач та вважають заданим.

Ключ – це, як правило, ціле число, що приймає множину фіксованих значень  $u(e_v) \in U_0$ ,  $U_0 = \{u_k\}_{k=1}^L$  – множина значень хеш-функції. Для векторів із  $B^n$  таким ключем може бути, наприклад, число одиничних бітів у складі дескриптору, число вибраних комбінацій бітів довільної довжини та ін. Взагалі ознаку  $u(e_v)$  для хешування можна корисно адаптувати до значень описів наявної бази еталонів.

Із впровадженням хешування розбиваємо множину даних  $E$  на непересічні групи  $U_k(E)$  (ланки, кошики, полиці) числом  $L$  з рівнозначним значенням хеш-ключа  $u(e_v) = u_k$ .

$$E = U(E) = \cup_{k=1}^L U_k(E), \quad U_k(E) \cap U_j(E) = \emptyset, \quad U_k(E) = \{e_v \in E \mid u(e_v) = u_k\}. \quad (2.8)$$

### 2.3 Класифікація на основі хеш-структури

Застосовуємо детектор КТ до зображення розпізнаваного об'єкту, формуємо опис  $Z = \{z_w\}_{w=1}^s$  об'єкту як множину дескрипторів. Побудуємо класифікатор з використанням швидкісного пошуку у базі  $E$  на підставі створеної структури [41 – 43].

Спосіб із впровадженням хешування включає такі дії.

Етап 1. Для дескриптора  $z_w$  вхідного образу обчислюємо величину хеш-ключа  $u_w = u(z_w)$ ,  $u_w \in U_0$  як значення хеш-функції для аргумент  $z_w$ .

Етап 2. Переходимо до хеш-кошика зі значенням  $u_w$ , а далі реалізуємо один із способів пошуку в межах кошика: лінійний, найближчих сусідів, за розподілами даних чи інші. Лінійний пошук та визначення найближчих сусідів в межах ланки реалізують точну процедуру. Використання оцінок розподілів за класами для окремих ланок хеш-кошика сприяє прискоренню обчислень і відповідає наближеному оціночному способу на підставі навчання.

Етап 3. За результатом пошуку на етапі 2 для елемента  $z_w$  визначаємо клас  $d_w$ .

Етап 4. На підставі аналізу усієї множини  $Z$  дескрипторів об'єкту накопичуємо лінійку  $\{r_i\}_{i=1}^N$  значень голосів за кожний з еталонних класів.

Етап 5. За виразом (2.3) класифікуємо об'єкт до класу, що набрав найбільшу кількість голосів.

Як бачимо, запропонований продуктивний за швидкістю спосіб класифікації на підґрунті впровадження хешування може реалізувати як точний, так і наближений види пошуку. Параметрами є число груп (хеш-

кошиків), побудованих для компонентів даних в індексованій структурі. Цим параметром можна керувати, виходячи із наповненості кошиків для конкретних даних.

Розподіли за класами еталонів для створених груп даних теж несуть важливу інформацію для класифікації, на підставі чого можна видалити із аналізу окремі кошики (якщо класи всередині них розподілені суто рівномірно), або за значеннями розподілу приймати однозначне класифікаційне рішення (якщо спостерігаються суттєві переваги для окремих класів).

За результатами попереднього оброблення еталонної інформації (навчальна, контрольна вибірки) можна оцінити результативність застосування класифікатора. Точність (*Precision*) класифікації можна оцінити відношенням загального числа *TP* дескрипторів бази еталонів, для яких правильно визначено клас, до загального їх числа у відповідності до моделі [2, 14, 44]

$$prec = \frac{TP}{TP + FP}, \quad (2.9)$$

де *TP* – число дескрипторів з правильно визначеним поточним класом;

*FP* – число дескрипторів помилково віднесених до поточного класу.

Найвища точність відповідає значенню  $prec = 1$ .

Іншим дієвим критерієм якості є повнота (*Recall*), який по суті є відношенням числа *TP* дескрипторів до всіх дескрипторів, що дійсно відносяться до поточного класу.

$$rec = \frac{TP}{TP + FN}, \quad (2.10)$$

де *TP* – число дескрипторів з правильно визначеним поточним класом;

*FN* – помилково не віднесених до поточного класу.

Найвища повнота відповідає значенню  $rec = 1$ .

### **3 РЕЗУЛЬТАТИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ МЕТОДУ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ**

#### **3.1 Огляд програмних засобів**

Моделювання виконано у середовищі PyCharm 2018 із використанням засобів бібліотеки OpenCV та мови програмування Python.

Python – багатоцільова та багатоплатформенна мова програмування, яка дозволяє писати код, що добре читається. Розроблена в 1990 році. Відносний лаконізм мови Python дозволяє створити програму, яка буде набагато коротше свого аналога, написаного на іншій мові. Програми на Python можна запускати на комп'ютерах під різними операційними системами без будь-яких змін вихідного коду та налаштувань.

Ще однією перевагою 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 ін.

Один з найсуттєвіших недоліків Python – невисока швидкість виконання коду. Даний недолік зумовлений тим, що Python не є компільованою мовою. Код, написаний на Python, спочатку компілюється у внутрішній байт-код, який потім виконується інтерпретатором Python. У більшості випадків при використанні Python програми виходять повільнішими ніж аналогічні, написані на мовах більш низького рівня, таких як C, C++, або Java.

Втім, сучасні комп'ютери мають таку обчислювальну потужність, що для більшості застосунків швидкість розробки важливіша швидкості виконання, а програми на Python зазвичай пишуться набагато швидше.

Окрім того, Python легко розширюється модулями, написаними на C або C++. Такі модулі можуть використовуватися для виконання частин програми, що створюють інтенсивне навантаження на процесор.

Мова програмування Python може використовуватися у широкому спектрі задач різного розміру і складності. Виходячи з цього, програми, написані з допомогою мови програмування Python, можуть бути як невеликими скриптами, так і складними системами. Найбільш часто мову програмування Python використовують для проведення наукових досліджень, створення штучного інтелекту, систем розпізнавання зображень та комп'ютерного зору.

PyCharm – інтегроване середовище розробки для мови програмування Python. Надає засоби для аналізу коду, графічний зневаджувач, інструмент для запуску юніт-тестів і підтримує веброзробку на Djangoю Середовище розробки PyCharm розроблене компанією JetBrains у 2010 році на основі середовища розробки IntelliJ IDEA.

PyCharm працює під операційними системами Windows, Mac OS X і Linux.

OpenCV – бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. Реалізована у 2006 році з допомогою мов програмування C та C++, може бути застосована 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.

Бібліотека OpenCV набула широкого застосування у сфері розпізнавання зображень, насамперед, через те, що її можна використовувати в інтеграції з більшою частиною мов програмування.

Дана бібліотека була обрана для розробки, завдяки наявності реалізованого необхідного метода пошуку КТ, простоті встановлення та використання.

### 3.2 Результати комп'ютерного моделювання

Для визначення дескрипторів КТ застосовано детектор ключових точок ORB з дескриптором розмірністю  $n=256$ .

Розроблені моделі класифікатора вжито на прикладі зображень порід собак (німецька вівчарка, хаскі, сербернар), розмір зображень масштабовано до  $500 \times 500$  пікселів. Ілюстрація класів еталонних зображень та сформовані координати КТ показані на рисунку 3.1. Кількість дескрипторів у описі кожного із еталонів складає  $s=500$ .

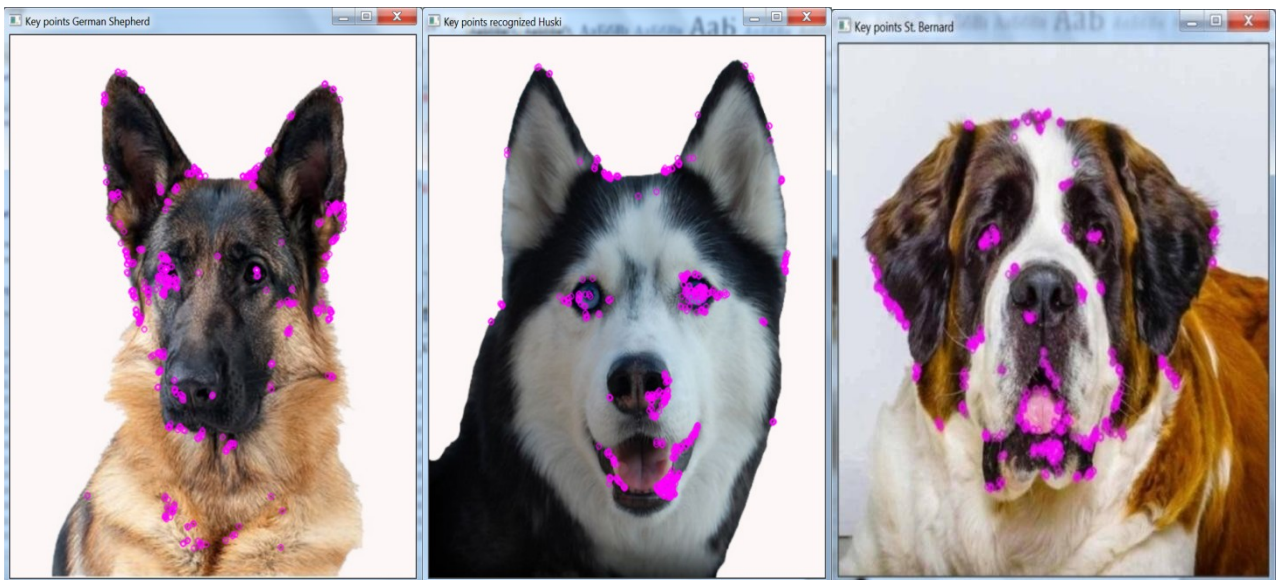


Рисунок 3.1 – Еталонні зображення порід собак з координатами КТ

Розроблені моделі класифікатора вжито на прикладі зображень порід собак (німецька вівчарка, хаскі, сербернар), розмір зображень масштабовано до  $500 \times 500$  пікселів. Ілюстрація класів еталонних зображень та сформовані координати КТ показані на рисунку 3.1. Кількість дескрипторів у описі кожного із еталонів складає  $s=500$ .

Як бачимо, візуально ці зображення за зовнішніми ознаками досить близькі. Якщо у якості хеш-функції  $u(z_w)$  взяти число одиничних бітів у векторі розмірністю 256, то кількість вибраних кошиків змінюється у межах від 1 до 257, тобто найбільша множина  $U_0$  містить значення  $0, 1, 2, \dots, 256$ . Всередині кошиків у експерименті для спрощення застосовано традиційний

лінійний пошук. Зважаючи на бінарний вид аналізованих даних, для зіставлення векторів застосовано метрику Хемінга.

З метою зіставлення характеристик швидкодії програмно змодельований метод лінійного пошуку на усій множині еталонних даних (1500 дескрипторів) без застосування хешування. Для еталонних зображень із застосуванням хешування на 256 кошиків та без хешування отримано максимально точний результат: усі 500 дескрипторів віднесені до свого еталону. При цьому час оброблення без хешування склав приблизно 40 с, а із хешуванням – лише 0,6 с. Як бачимо, вираш у часі обчислень для запропонованого підходу (при заданому числі 3 еталонів та  $S=500$ ) складає більше 65 разів у порівнянні із традиційним. При цьому ясно, що вираш у часі обчислень збільшується при зростанні числа еталонів та дескрипторів у описах. Аналогічні точні результати класифікації еталонів отримано і для іншої кількості кошиків.

Вивчення отриманої множини дескрипторів за визначеною хеш-функцією для зображень рисунку 3.1 показало, що в межах її значень від 0 до 92 та від 217 до 255 взагалі немає жодного дескриптора. Розподіл дескрипторів для досліджуваної бази еталонів на 3 кошики, рівноцінні за діапазоном значень хеш-функції, показано у таблиці 3.1.

Таблиця 3.1 – Кількісний розподіл дескрипторів бази

Значення хеш-функції	Число дескрипторів
93 – 133	375
134 – 174	1004
175 – 216	121

Експериментальний результат таблиці 3.1 відкриває перспективу подальшого скорочення об'єму обчислень, так як допускає виключення із аналізу суттєвої не використовуваної частини від загального числа кошиків. У той же час ця можливість сприяє забезпеченню завадостійкості роботи методу з хешуванням.

Важливим для забезпечення результативної класифікації представляється вплив параметра порогу  $\delta_\rho$  для значення мінімальної відстані при прийнятті рішення про клас дескриптора об'єкта за правилом (2.1). Діапазон значень для вибраної метрики складає  $\delta_\rho \in [0, \dots, 256]$ .

Моделювання для навчальної вибірки показало, що зі зменшенням порогу відсікається більша частина «чужих» дескрипторів (інших еталонів), але при цьому одночасно губиться і значна частина «своїх». Зі зростанням порогу  $\delta_\rho$  спостерігається збільшення числа «своїх» з одночасним збільшенням числа «чужих». Як бачимо, виникає конкурентна ситуація, розв'язання якої може бути досягнуто шляхом оптимізації значення порога конкретно для наявної бази еталонів, а критерієм виступає точність (8) класифікації, результативним фактом при цьому є переважаюче число правильно класифікованих дескрипторів для свого класу. Оптимізація універсально може бути реалізована експериментальним шляхом для прикладної бази зображень. У даному дослідженні найкращі результати досягнуто при значенні  $\delta_\rho = 100$ .

Найбільш критично вибір порога впливає на результат класифікації при геометричних перетвореннях вхідного зображення. Експерименти з дією перетворення повороту (найбільшим чином викривляє зображення, рисунок 3.2) для вхідних еталонних зображень показали, що при низьких порогах втрачається статистично значиме число правильно класифікованих дескрипторів, що потребує збільшувати значення порогу. У той же час навіть при невеликому значенні порогу  $\delta_\rho = 25$  вхідні зображення еталонів (навчальна вибірка) класифікуються вірно, незважаючи на суттєве зниження числа виявлених «своїх» даних (від 330 до 129 для різних еталонів).

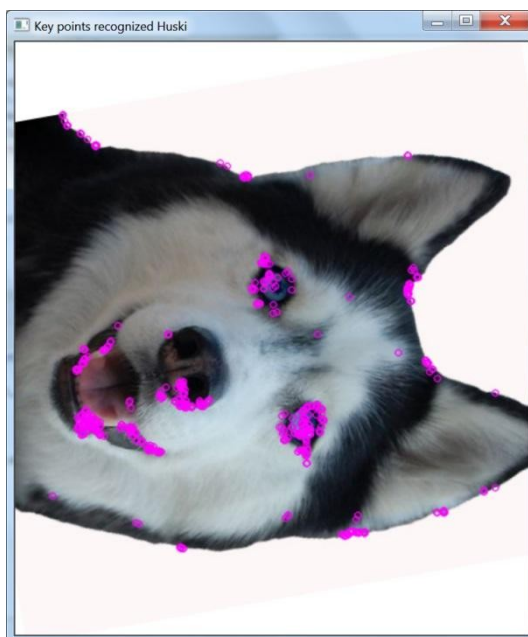


Рисунок 3.2 – Зображення з координатами КТ при повороті

Дослідимо тепер детальніше залежність часу класифікації (в секундах) від числа кошиків. Експериментальні дані поміщені в таблиці 3.2, число кошиків 1 відповідає варіанту без хешування.

Таблиця 3.2 – Залежність часу класифікації від числа кошиків

Число хеш-кошків	257	128	64	32	16	8	1
Час класифікації	0,6	1,3	2,5	4,9	9,5	18,4	40

Як бачимо, експериментальний час класифікації для розглянутих описів лінійно зростає зі зменшенням числа кошиків при застосуванні хешування. Таким чином, при необхідності прийняття найбільш швидкісних рішень треба вибирати максимально можливе число кошиків. Отримана залежність властива виключно для аналізованих еталонних зображень. У загальному плані необхідно додатково вивчати цю залежність для досліджуваної бази даних. Найбільш несприятливі для обсягів часу є ситуації, коли, наприклад, значна кількість дескрипторів різних класів зосереджуються у спільному хеш-кошику.

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 значно швидше у порівнянні з обчисленням метрики для множини дескрипторів всередині кошиків. Задля оцінювання ступеня відмінностей між описами еталонних даних обчислено манхетенську відстань для числа кошиків  $L = 257$ . Ці відстані складають 234, 196, 274, що підкреслює досить значиму близькість еталонів у просторі ознак, так як максимум такої відстані дорівнює 1000 за числом дескрипторів для пари описів. Найменша відстань досягнута у порівнянні 1 та 3-го еталонів представлених на рисунку 3.1.

Для навчальної вибірки, коли на вхід подаються еталони, значення параметра точності  $prec=1$  як для традиційного, так і для пропонованого методу з використанням хешування. Але в реальній ситуації, наприклад, при перетвореннях повороту, величина  $prec$  коливається у межах до 0,47 з хешуванням та 0,53 без хешування з правильним визначенням класу.

Незважаючи на деяке зниження показника точності, у проведених експериментах класифікація здійснювалася вірно, тобто усі вхідні зображення з перетвореннями правильно отримали свій номер класу. Ці результати підтверджують ефективність розробленого методу.

Важливою проблемою у розпізнаванні зображень є опрацювання певним чином змінених зображень. У даній роботі було проведено експерименти з класифікацією зображень під дією адитивного гаусівського шуму (рис. 3.3) та частин (половинок) еталонних зображень (рис. 3.4).

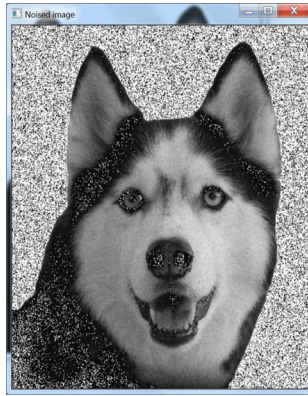


Рисунок 3.3 – Зображення з адитивним гаусівським шумом

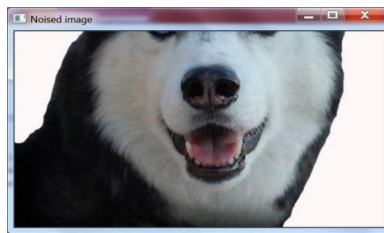


Рисунок 3.4 – Нижня половина еталонного зображення

Результати розрахунку середніх значень критеріїв якості *Precision* та *Recall* для трьох класів для частин (половинок) еталонних зображень та зображень з адитивним гаусівським шумом, згенерованим за нормальним законом при  $\sigma = 10$  з хешуванням і без хешування наведено в таблиці 3.3.

Таблиця 3.3 – Середні значення критеріїв якості

Експеримент	Хешування	Precision	Recall
Половини еталонних зображень	Є	0,933	0,917
Половини еталонних зображень	Немає	1	1
Адитивний гаусівський шум	Є	0,724	0,6
Адитивний гаусівський шум	Немає	0,889	0,867

Аналізуючи дані з таблиці 3.3 бачимо, середні значення критеріїв якості погіршуються доволі суттєво відносно до значень при класифікації без хешування. З іншого боку виграш у часі, що надає хешування, доволі суттєвий – час класифікації скорочується приблизно у 50 разів, як було видно з попередніх експериментів. Для більш детального аналізу доцільності застосування засобів хешування даних треба досліджувати результати з хешуванням для наведених експериментів окремо.

Виходячи з таблиці 3.3, середні значення *Precision* та *Recall* зазнають незначного зменшення при класифікації частин зображень. У даній ситуації використання апарату хеш-функцій можна вважати виправданим, так як значення критеріїв близькі до одиниці (більше 0,8).

Якщо брати до уваги класифікацію зображень в умовах дії адитивного гаусівського шуму, то спостерігаються деякі втрати у показниках якості класифікації, так як значення критерію *Recall* зменшуються до 0,6. Рішення з приводу застосування апарату хеш-функцій доцільно приймати, виходячи з пріоритетних показників часу чи якості класифікації. Можлива побудова комбінованих показників ефективності.

Якщо є необхідність швидко класифікувати зображення і є можливість дозволити системі допускати неточності у класифікації, то можна обрати метод з використанням хешування.

У протилежному випадку перевага має бути надана традиційному методу. Іншим рішенням є зменшення кількості хеш-кошиків при проведенні хешування, тобто знаходження певного роду компромісу між двома характеристиками. Будь-яке з наведених рішень дозволить покращити показники критеріїв якості класифікації за рахунок зменшення швидкодії класифікації.

Було прийнято рішення, що у даному випадку швидкодія є більш важливою, ніж висока точність класифікації. Тому для даної роботи було обрано метод з хешуванням.

На даний час при прийнятті подібних рішень перевага частіше надається швидкодії оброблення через необхідність саме швидкого реагування на інформацію, що надходить. Тому можна вважати вибір класифікатора з хешуванням повністю виправданим.

Таким чином, із введенням хешування і зростанням числа кошиків точність незначно знижується, але практично не залежить від числа кошиків. Основним недоліком впровадження хешування традиційно вважається зростання необхідних обсягів пам'яті комп'ютера.

## ВИСНОВКИ

Розроблені і досліджені у кваліфікаційній роботі методи класифікації базуються на принципі «порівняння з еталоном» і можуть бути універсально застосовані до довільних векторів даних бітового виду. Ці методи базуються на двох різновидах класів – для цілісного опису (об’єкт та еталони) та для окремих компонентів опису.

При використанні методу з лінійним пошуком показники якості класифікації *Precision* та *Recall* дорівнюють одиниці у випадку класифікації еталонних зображень і при класифікації зображення з геометричним перетворенням.

При застосуванні хеш-структур ключовим моментом є вибір результативної та адаптованої до даних хеш-функції, яка здійснює часткову класифікацію, не змінюючи дані.

Використання хеш-структури у класифікаторах значно скорочує час, необхідний для пошуку найбільш схожих дескрипторів еталонного зображення та вхідного зображення. Проте такий підхід дещо знижує показники критеріїв якості *Precision* та *Recall* при класифікації зображень з перетворенням за рахунок того, що два потенційно найбільш схожих дескриптора можуть опинитися у різних хеш-кошиках.

Під час реалізації запропонованих підходів можливі варіанти оброблення з використанням наявної структурованості еталонних даних, тобто за окремими описами еталонів. Це дає можливість подальшого скорочення обсягів обчислень шляхом введення логічного аналізу за результатом послідовного обчислення релевантності для компонентів бази еталонів.

Основний результат проведеного моделювання складає розроблення продуктивного методу класифікації зображень на основі впровадження швидкісного пошуку із використанням хеш-структур, що зменшує обсяг обчислень в десятки разів.

Практична значущість роботи полягає у побудові моделей класифікації у структурованому просторі даних, підтвердженні працездатності запропонованих модифікацій на прикладах зображень, створенні програмних застосунків для впровадження розроблених методів класифікації у системах комп'ютерного зору.

Перспективи розробки можуть бути пов'язані із введенням логічного оброблення розподілів даних всередині хеш-кошиків, вивченням завадостійкості розроблених методів, оцінюванні їх прикладної результативності стосовно об'ємних колекцій зображень а також аналіз фрагментів зображень для детекції еталонів на інших зображеннях.

Окремим напрямком для подальших досліджень є навчання нейронних мереж на основі апарату дескрипторів ключових точок зображень, дослідження у цій сфері могли б дати більш якісну класифікацію за менший час завдяки гнучкості підходу.

Результати роботи апробовано у вигляді 3 тез доповідей під час Міжнародного молодіжного форуму «радіоелектроніка і молодь у XXI столітті» [17, 39], Міжнародної наукової конференції «інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту» [16], статті «Застосування засобів хешування даних для прискорення класифікаційних рішень у структурних методах розпізнавання зображень» [15] та конкурсі «Всеукраїнський конкурс студентських наукових робіт 2022 року зі спеціальності «Комп'ютерні науки»».

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Детекторы и дескрипторы особых точек FAST, BRIEF, ORB. URL: <https://habr.com/ru/post/414459/> (дата звернення: 20.04.2022).
2. Хешування. URL:<https://znaimo.com.ua/> (дата звернення: 20.04.2022).
3. 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.
4. Лесковец Ю., Раджараман А., Ульман Д. (2016) Анализ больших наборов данных, Москва, ДМК Пресс, 2016. 498 с.
5. P. Flach. (2012) Machine learning. The Art and Science of Algorithms that Make Sense of Data. New York, NY, USA: Cambridge University Press.
6. 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: 21.04.2022).
7. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V. (2014) Neural codes for image retrieval. Conference Paper. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8689 LNCS(PART 1), p. 584-599.
8. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ. 124 с.
9. 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.
10. 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.

11. 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.
12. Manning, C.D., Raghavan, P., Schütze, H. (2008) Introduction to Information Retrieval, Cambridge, University Press, 528 p.
13. 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.
14. Путятин Е.П., Аверин С.И. (1990) Обработка изображений в робототехнике. Москва: Машиностроение, 320 с.
15. 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.
16. 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.
17. Рибалка, М. (2021) Метод класифікації зображень з використанням принципу ближнього сусіда. Радіоелектроніка і молодь у XXI столітті: матеріали Міжнародного молодіжного форуму (20-22 квітня 2021 р., м. Харків). Харків: ХНУРЕ, с. 8-9.
18. Гороховатський, В.О., Творошенко, І.С. (2021) Методи інтелектуального аналізу та оброблення даних: навч. посібник. Харків: ХНУРЕ.
19. 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.

20. 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.
21. Gorokhovatskiy O., Gorokhovatskiy V., Peredrii O. (2018) Analysis of Application of Cluster Descriptions in Space of Characteristic Image Features. p. 52.
22. Gorokhovatskiy 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.
23. Гороховатский В.А., Путятин Е.П., Столяров В.С. (2017) Исследование результативности структурных методов классификации изображений с применением кластерной модели данных. Радиоэлектроника, информатика, управление, №3 (42). С. 78–85.
24. Gorokhovatsky, V.A., (2014), Structural analysis and intellectual data processing in computer vision, SMIT, Kharkiv, p 316.
25. Гороховатский В.А., Передрий Е.О. (2009) Корреляционные методы распознавания изображений путем голосования систем фрагментов. Радиоелектроніка. Інформатика. Управління, №1(20), с.74–81.
26. Путятін Є.П., Гороховатський В.О., Матат О.О. (2006) Методи та алгоритми комп'ютерного зору: навч. посібник.
27. Гороховатський В.О., Гадецька С.В., Стяглик Н.І. (2019) Вивчення статистичних властивостей моделі блочного подання для множини дескрипторів ключових точок зображень. Радиоелектроніка, інформатика, управління, №2, с. 100–107.
28. 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.
29. Гороховатский В.А. (2003) Распознавание изображений в условиях неполной информации, Харків: ХНУРЭ, 112с.

30. 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.
31. 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.
32. 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.
33. 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.
34. 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.
35. 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.
36. 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.

37. 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.
38. 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,
39. 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 (подано до друку).
40. Хеш-алгоритми. URL: <https://habr.com/ru/post/93226/> (дата звернення 19.04.2022).
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. В.А. Гороховатский, Ю.Н. Ересько, Е.П. Путятин, В.И. Стрельченко. (1990) Локализация объектов на изображениях визуальных сцен. *Авто-метрия*. №6. –С.3–7.
43. Scherer R. (2018) Computer Vision Methods for Fast Image Classification and retrieval, Częstochowa, *Springer*, p 137.
44. 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.