

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти перший (бакалаврський)

**МОДЕЛЮВАННЯ МЕТОДУ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ**  
**ІЗ ЗАСТОСУВАННЯМ КОЕФІЦІЄНТІВ**  
**ВАГОМОСТІ ДЛЯ ЕЛЕМЕНТІВ ОПИСУ**  
(тема)

Виконав:  
студент 4 курсу, групи ІТІНФ-19-2

Гончаров Д.М.  
(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)

2023 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)

« \_\_\_\_\_ » \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові \_\_\_\_\_ Гончарову Данилу Миколайовичу  
(прізвище, ім'я, по батькові)1. Тема роботи Моделювання методу класифікації зображень із застосуванням коефіцієнтів вагомості для елементів опису

затверджена наказом університету від 15 травня 2023 року № 474 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 29 травня 2023 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література з класифікації зображень, матеріали наукових конференцій, дані інтернет-мережі, бібліотека комп'ютерного зору з відкритим кодом OpenCV, середовище розробки Visual Studio Code, мова програмування C#, NuGet package Emgu.CV.

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

1. Аналіз методів опису зображень та їх оброблення.

2. Реалізація метричного алгоритму класифікації зображення за допомогою обчислення відстані в межах фіксованої бази еталонів.

3. Створення програмної реалізації класифікатора з використанням параметра вагомості.

4. Аналіз результативності розробленого алгоритму класифікації.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність вивчення та розвитку проблеми швидкості класифікації зображень, постановки задачі та застосування сучасних методів класифікації, постановка задачі, 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
Консультант з дотримання діючих стандартів та норм	Доцент Творошенко І.С.		

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	10.04.2023	
2	Аналіз завдання, підбір літератури	11.04.23-17.04.23	
3	Аналіз літератури з досліджуваної проблеми	18.04.23-20.04.23	
4	Аналіз способів класифікації	21.04.23-30.04.23	
5	Розробка методу оцінки за показником вагомості	03.05.23-15.05.23	
6	Програмна реалізація	16.05.23-23.05.23	
7	Оформлення пояснювальної записки	24.05.23-25.05.23	
8	Перевірка на плагіат	26.05.23	
9	Рецензування	27.05.23	
10	Підготовка презентації та доповіді	28.05.23-30.05.23	
11	Занесення роботи в електронний архів	31.05.23	
12	Попередній захист кваліфікаційної роботи	06.05.23	

Дата видачі завдання 10 квітня 2023 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

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

**РЕФЕРАТ/ABSTRACT**

Пояснювальна записка до кваліфікаційної роботи: 59 с., 2 табл., 13 рис., 47 джерел.

**РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, ДЕСКРИПТОР, КЛЮЧОВА ТОЧКА, ІНФОРМАТИВНІСТЬ, РЕЛЕВАНТНІСТЬ ОПИСІВ, ВІДСТАНЬ ХЕММІНГА.**

Об'єктом роботи є дослідження методів класифікації та кластеризації описів за допомогою відбору показників вагомості.

Метою цієї роботи є застосування та дослідження способів класифікації зображень, за рахунок обрахунку відстаней Хеммінга в множині дескрипторів ключових точках та за параметрами вагомості.

Виконано побудову програмної моделі класифікаційних ознак зображень, з використанням вагомості для відбору частини дескрипторів для прискорення процесу оцінки для економії ресурсів при класифікації.

На основі впровадження такого методу та попередньої оцінки показнику вагомості та зменшення загальної кількості необхідних операцій для проведення класифікації, при забезпеченні такого ж рівня результативності на навчальній вибірці даних.

**IMAGE RECOGNITION, DESCRIPTOR, KEY POINT, INFORMATIVENESS, RELEVANCE OF DESCRIPTIONS, HAMMING DISTANCE.**

The object of the work is the study of methods of classification and clustering of descriptions using the selection of weighting indicators.

The aim of this work is the application and research of methods of image classification, due to the calculation of Hamming distances in a set of descriptors of key points and by weight parameters.

The construction of a software model of image classification features was carried out, using weighting to select a part of descriptors to speed up the evaluation process to save resources during classification.

Based on the implementation of such a method and preliminary assessment of the weighting indicator and reduction of the total number of necessary operations for classification, while ensuring the same level of effectiveness on the training sample of data.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	6
Вступ.....	7
1 Розпізнавання зображень за множиною ключових точок .....	9
1.1 Поняття і застосування ключових точок.....	9
1.2 Аналіз детекторів для формування дескрипторів ключових точок.....	11
1.3 Класифікація зображень за множиною дескрипторів.....	22
1.4 Постановка задачі .....	27
2 Моделі для класифікації зображень з використанням вагомості.....	28
2.1 Коефіцієнти вагомості для елементів опису.....	28
2.2 Вагомість інтегрованих ознак.....	32
2.3 Статистичне та блокове подання описів .....	35
3 Результати комп'ютерного моделювання .....	41
3.1 Вибір фреймворку та мови програмування.....	41
3.2 Аналіз особливостей програмної моделі.....	47
3.3 Аналіз результатів експериментів.....	50
Висновки .....	54
Перелік джерел посилання .....	55

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

КТ – ключові точки

AKAZE – Accelerated-KAZE

GDI – Graphics Device Interface

GUI – Graphics User Interface

CNN – Convolutional Neural Networks (згорткові нейронні мережі)

ROI – Region of Interest (область інтересу)

BRIEF – Binary Robust Independent Elementary Features (бінарний дескриптор інтенсивності та особливих точок)

BRISK – Binary Robust Invariant Scalable (бінарний надійний інваріант масштабований)

SURF – Speeded Up Robust (прискорені надійні функції)

SIFT – Scale-Invariant Feature Transform (масштабно-інваріантний перетворювач ознак)

PCA – Principal Component Analysis (метод головних компонент)

ROC – Receiver Operating Characteristic (характеристика роботи приймача)

АКК – Adaptive Quality Coefficient (адаптивний коефіцієнт якості)

## ВСТУП

Сучасні напрями науки, техніки та виробництва зосереджені на розробці систем, в яких інформація представлена у вигляді зображень. Однією з найскладніших проблем є аналіз та розпізнавання зображень [1-4]. Цифрова обробка зображень має широкі та різноманітні застосування. Існує багато суміжних областей, таких як аналіз зображень та комп'ютерний зір, але їх відокремлення від цифрової обробки зображень є досить складним. Процеси цифрової обробки зображень можна розділити на низького, середнього та високого рівнів [5-9].

Низький рівень включає прості операції, такі як зменшення шуму, підвищення контрастності або поліпшення різкості зображень. Вхідним та вихідним даними для цього рівня є зображення. Високий рівень включає аналіз зображень, що передбачає розуміння та інтерпретацію набору розпізнаних об'єктів.

Обробка зображень на середньому рівні охоплює такі завдання, як сегментація – виділення на зображенні об'єктів, опис об'єктів і стиснення їх в зручну для комп'ютерної обробки форму, а також класифікація (розпізнавання) окремих об'єктів. Для процесів середнього рівня характерна наявність зображень тільки на вході, на вихід само роблять ознаки і атрибути, які добувають із цих зображень, такі як кордони областей, лінії контурів, відмінні ознаки конкретних об'єктів, тощо. Отже, основне завдання розпізнавання образів – одержання семантичного опису зображених об'єктів. Мета розпізнавання може бути різною: як виділення окремих елементів на зображенні, так і класифікація зображення в цілому [1].

Класифікація зображень застосовується при автоматизації, ідентифікації видів, процесах управління (наприклад промисловими роботами), системах розпізнавання тексту, виявленні подій, задачах взаємодії між людиною та комп'ютером, моделюванні, медицині, навігації, організації

інформації. Тому на сьогодні комп'ютерний зір, а також пов'язані з ним технології програмної та апаратної обробки, є одним з перспективніших напрямків у світі комп'ютерних технологій. Постає питання створення ефективних алгоритмів, спрямованих на вирішення завдань класифікації, які працюють максимально точно, з мінімумом помилок та з високою швидкістю.

Автоматизація аналізу та обробки зображень, а також класифікація візуальних об'єктів для отримання статистичних даних є важливим завданням при створенні систем штучного інтелекту для сучасних високотехнологічних виробництв. Основними ускладнюючими факторами є: зовнішнє середовище у вигляді фону і перешкод, а також геометричні перетворення, які приводять до зміни об'єктів та відповідно заважають алгоритмам дати правильні відповіді. Адже такий вплив може призвести до викривлення фрагментів аналізованого об'єкта або сформувати помилкові об'єкти з близькими характеристиками [2].

Вивчення моделювання методу класифікації зображень із застосуванням коефіцієнтів вагомості для елементів опису може бути корисним з кількох причин. Це може допомогти покращити точність класифікації зображень, що є важливим у багатьох областях, включаючи медицину, технології безпеки та автоматичне розпізнавання образів. Також вивчення даної теми може знайти своє застосування у вивченні штучного інтелекту, що є актуальними напрямками розвитку сучасних технологій.

У даній роботі запропоновано та програмно змодельовано метод структурної класифікації зображень на основі їх структурних описів з використанням параметрів вагомості та інформативності, отримано висновки по результатам його прикладного використання. Застосування коефіцієнтів вагомості для елементів опису зображень дозволило покращити точність класифікації та знизити вплив зовнішніх факторів на результати аналізу. Цей метод може бути ефективно використаний для автоматизованого аналізу зображень у високотехнологічних виробництвах.

# 1 РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ЗА МНОЖИНОЮ КЛЮЧОВИХ ТОЧОК

## 1.1 Поняття і застосування ключових точок

Виникає складна задача вилучення з рисунку осмисленої та структурованої інформації. Для її вирішення можуть бути використані технології, що базуються на аналізі особливостей зображення в окремих точках [3-5]. Шляхом аналізу характеристик кінцевої кількості характерних ознак (КТ) можна вирішити складні практичні завдання.

Перевагою системи характерних ознак для розпізнавання зображень є спрощення процесу формування ознак, зменшення обсягу ознак, можливість універсального структурного аналізу об'єктів та стійкість до фонових спотворень та помилок. Ключова точка (КТ), або точкова особливість зображення, повинна мати деякі особливості, такі як відмінність, інваріантність, стабільність та інтерпретацію. Крім того, КТ повинна бути локальною, достатньо чисельною, точною та ефективною. Ці властивості допомагають забезпечити надійне розпізнавання зображень та забезпечують ефективність роботи системи.

Утворення КТ передбачає виділення фрагменту зображення від фону. Процес створення множини дескрипторів КТ складається з двох етапів. Перший етап полягає в знаходженні ключових точок зображення (детектор). Детектором називається метод, що виокремлює точки або локальні області, які мають особливості, що відрізняють їх від сусідніх точок. Другий етап передбачає безпосереднє формування дескриптора. Дескриптором називається функція, що описує КТ за її околицею. Його можна вважати альтернативним представленням КТ, для якого можна визначити міру схожості. Так, наприклад, для опису точки  $c$  з координатами  $c = (x, y)$  використовується просторова околиця  $O(c)$ ,  $c \in O(c)$ , де  $O(c)$  – результат

застосування оператора з побудови множини точок, координати яких знаходяться в безпосередній близькості, тобто  $O(c) = [\{c_i = (x_i, y_i)\}, \forall c_i : \rho(c_i, c) \leq \nu]$ , де параметр задає розмір околиці, а  $\rho$  – деяку відстань між точками на площині [3, 6].

Точкова особливість на зображенні  $B(x, y)$ ,  $(x, y) \in D$ ,  $D \subset R \times R$  – це точка  $c^*$ , в околиці  $O(c^*)$ ,  $O(c^*) \subset D$  якої функція  $B(x, y)$  за деякими властивостями відрізняється від множини своїх значень з більшою за розміром околиці  $O_1(c^*)$ ,  $O(c^*) \subset O_1(c^*)$  [3, 6]. КТ формується з множини точок з околиці  $O(c^*)$ .

Продемонструємо поняття дескрипторів на прикладі: двовимірну функцію зображення  $B(x, y)$  представимо у вигляді скінченної множини фрагментів  $\{b_k(x, y)\}$  або як кінцеву множину  $\Lambda = \{\lambda^k\}$  КТ. При цьому з кожним  $\lambda^k$  пов'язано елемент  $c^k = (x^k, y^k)$  множини координат  $C = \{c^k\}$ , який визначає просторові властивості опису. В результаті візуальний об'єкт представляється у вигляді сукупності дескрипторів (рис. 1.1), що включають атрибути  $\lambda^k$  та координати  $c^k$ .

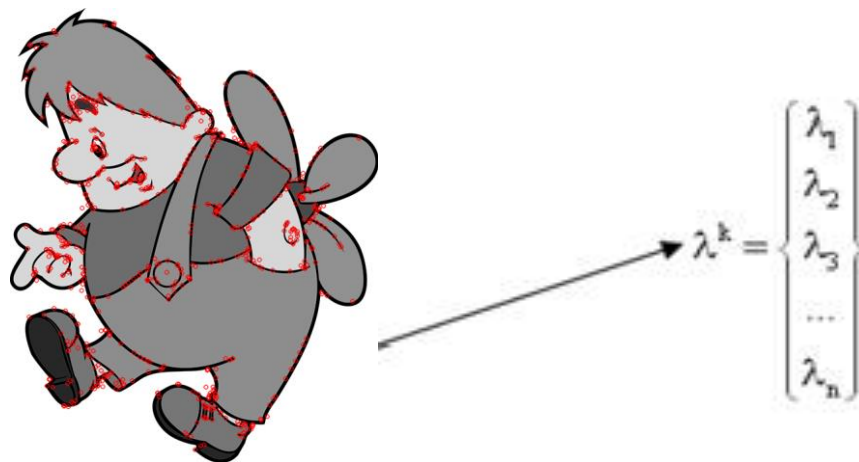


Рисунок 1.1 – Побудова опису у вигляді множини характерних ознак

Зазвичай формування множини КТ зображення відображає кривизну функції яскравості та рівень відмінності в порівнянні з сусідніми

фрагментами. Різноманітні алгоритми пошуку КТ можуть бути спрямовані на виявлення кутів, перепадів, кордонів, контрастних точок, ліній та іншого, залежно від використовуваного математичного апарату [7-9].

Перехід до множини ключових точок зображення дозволяє суттєво скоротити обсяг інформації, зберігаючи при цьому високу достовірність та стійкість до перешкод. Це досягається за рахунок виділення та обробки нової інформації, яка міститься в структурних елементах і зв'язках. Зазвичай, в більшості застосувань, ХО представляють собою числовий вектор з фіксованою розмірністю та речовими значеннями.  $\lambda^k = (\lambda_1, \dots, \lambda_n)^k$ ,  $\lambda_i \in R^1, i = \overline{1, n}$ .

## 1.2 Аналіз детекторів для формування дескрипторів ключових точок

В рамках роботи проаналізуємо різні алгоритми пошуку ключових точок та формування дескрипторів зображення, кожен з яких має свої переваги та недоліки. Особлива увага буде приділена вивченню структури даних для множини дескрипторів, яка має значення для ефективності систем розпізнавання. Всі алгоритми можна розділити на дві основні групи залежно від способу представлення дескриптора: дескриптори з рухомою точкою (найбільш популярні алгоритми: SURF, SIFT [10], CSLBP та ін.) та бінарні дескриптори (найбільш популярні алгоритми: BRIEF [11], BRISK [12], AKAZE [13], ORB [14] та ін.).

Проведемо аналіз методів, які використовують дескриптори з рухомою точкою, зокрема SIFT (Scale Invariant Feature Transform). Цей метод будує простір змінного масштабу, в якому обчислюються функції Лапласіана Гаусіана з різним параметром згладжування. Точки, які є локальними екстремумами різниці Гаусіанів, вважаються ключовими. Далі з множини можливих ключових точок видаляються ті, що потрапили на кордон об'єктів або мають малий контраст, і обчислюються їх орієнтації. Для цього

будується зважена гістограма градієнтів в околиці, і обирається напрямок, що відповідає максимальній компоненті гістограми. Точці присвоюються всі напрямки, яким відповідають значення компонент гістограми, великих за заданим порогом. Загалом, метод SIFT є інваріантним щодо зсувів, обертань та масштабування, що забезпечує збереження локальних екстремумів.

**SURF: Speeded Up Robust Features.** Матриця Гессе використовується в алгоритмі SURF для знаходження КТ. Гессіан відображає екстремуми матриці Гессе, що відбувається в точках максимальної зміни градієнта яскравості. Після знаходження координат КТ метод SURF формує їх дескриптори – набір з 64 (або 128) чисел для кожної КТ, що відображають флуктуації градієнта навколо ключової точки. Цей алгоритм є інваріантним до повороту зображення.

**CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching.** Алгоритм CenSurE виконує визначення КТ у три кроки. Спочатку обчислюються відгуки на спрощену дворівневу функцію Лапласіана Гаусіана та малі відкидаються, що дозволяє визначити межі на зображенні. На другому кроці обчислюються локальні екстремуми, і за допомогою метрики Харріса визначаються екстремуми з високим відгуком. Для обчислення дескрипторів застосовується алгоритм SURF, який використовує зважені Гаусіаном вейвлети Хаара для обчислення відгуків в горизонтальному і вертикальному напрямках для областей [15].

Дескриптори SURF представляють собою вектор з 128 (або 64) чисел для кожної ключової точки, і порівняння точок здійснюється шляхом пошуку найближчої точки (або двох), причому чим ближче точка, тим краще результат. Таким чином, на зображенні з 1000 ключових точок, необхідно обробити 128 000 чисел з плаваючою комою. Однак виявлення ключових точок є складною та об'ємною операцією, що не дозволяє ефективно використовувати цей алгоритм на мобільних пристроях.

Алгоритми, що використовують бінарні дані, створюють дескриптори у вигляді бінарних векторів, розмір яких кратний ступеню двійки. Така бітова

природа дескрипторів дозволяє представляти та аналізувати їх як послідовності елементів, наприклад байтів, з відомим діапазоном значень. Бінарне подання сприяє швидкому порівнянню дескрипторів за допомогою двійкових операцій та спрощує апаратну реалізацію систем розпізнавання. Крім того, використання бінарної арифметики дає можливість застосувати ефективний апарат оброблення бінарних даних та створити нові підходи до визначення подібності дескрипторів при побудові правил класифікації.

Для знаходження кутових точок проводиться послідовний аналіз околиць пікселів радіусом 16 пікселів для кожного пікселя  $p$  (рис. 1.2) з використанням алгоритму Брезенхема. Конкретна точка  $p$  вважається кутовою, якщо в окрузі навколо неї є суміжні пікселі, інтенсивність яких вища, і тому вона є підозрілою на кутову  $I_p + t$  або інтенсивності всіх менше  $I_p - t$ , де  $I_p$  – інтенсивність точки  $p$ ,  $t$  – порогова величина. Далі необхідно порівняти інтенсивність у вертикальних і горизонтальних точках на окружності з інтенсивністю в точці  $p$ . При виконанні цієї умови далі досліджується значення яскравості на окружності під номерами 1, 5, 9, 13 (рисунок 1.2). Якщо для трьох пікселів з чотирьох виконується умова  $I_i < I_p - t$  або  $I_i > I_p + t$ ,  $i = 1, \dots, 4$ , то тільки в цьому випадку проводиться повний тест для усіх 16 точок.

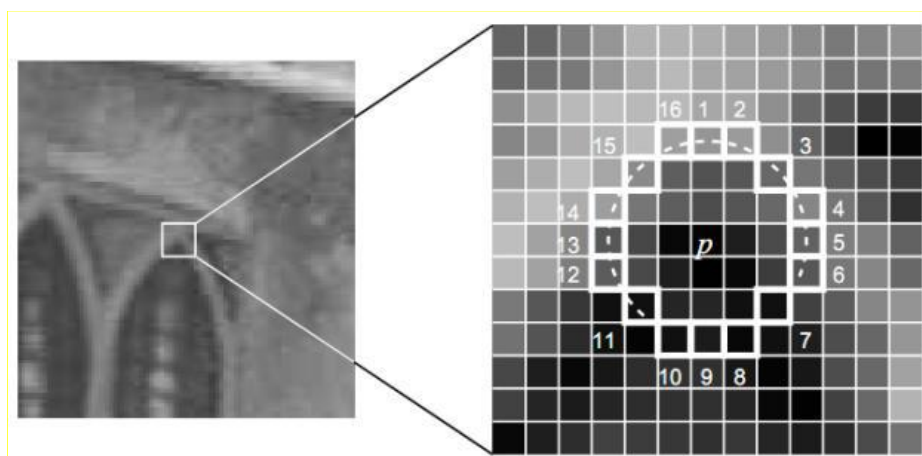


Рисунок 1.2 – Аналіз околиці точки  $p$  детектором FAST

Обмеженням вибору лише 4 пікселів на окрузі є можливість швидко відфільтрувати непідходящі точки, але у деяких випадках це може призвести до пропуску різних особливостей в межах однієї округлої області.

BRIEF (Binary Robust Independent Elementary Features) – це перший аналізований алгоритм, що будує бінарні дескриптори, обчислення яких відбувається після знаходження особливих точок. BRIEF є швидким евристичним дескриптором, який будується на основі порівняння яскравості пікселів на розмитому зображенні. Алгоритм спочатку дозволяє перетворити вихідне зображення в бінарні рядки, побудовані для кожної області. Згладжене зображення розділяється на ділянки, і для них унікальним чином вибирається набір точок. Потім для них порівнюється інтенсивність. Бінарний тест між точками  $x$  і  $y$  визначається так [9].

Для знаходження КТ спочатку розглядається окіл з радіусом 16 пікселів навколо кожного пікселя  $p$  за допомогою алгоритму Брезенхема. Якщо на цьому колі є кілька пікселів, інтенсивність яких вища за піксель  $p$ , то точка  $p$  вважається підозрілою на те, що вона є кутовою  $I_p + t$  або інтенсивності всіх менше  $I_p - t$ , де  $I_p$  – інтенсивність точки  $p$ ,  $t$  – порогова величина. Далі необхідно порівняти інтенсивність в вертикальних і горизонтальних точках на окружності з інтенсивністю в точці  $p$ . При виконанні цієї умови далі досліджується значення яскравості на окружності під номерами 1, 5, 9, 13 (рис. 1.2). Якщо для трьох пікселів з чотирьох виконується умова  $I_i < I_p - t$  або  $I_i > I_p + t$ ,  $i = 1, \dots, 4$ , то тільки в цьому випадку проводиться повний тест для усіх 16 точок.

Метод BRISK (Binary Robust Invariant Scalable Keypoints) – це алгоритм детектування та опису КТ зображення з використанням бінарних дескрипторів. BRISK є ефективним методом порівняно з іншими методами, такими як SIFT та SURF, зокрема, завдяки бінарному опису точок, що робить його більш швидким та меншим за обсягом пам'яті.

Алгоритм BRISK складається з кількох етапів. Першим етапом є визначення позицій КТ зображення з використанням пірамідального аналізу. На кожному рівні піраміди виконується пошук локальних максимумів детектором Харріса, після чого точки порівнюються між рівнями піраміди, щоб отримати найкращі ключові точки.

Другий етап – це обчислення бінарного дескриптора для кожної ключової точки. Дескриптор складається з двох частин: радіального та орієнтаційного дескрипторів. Радіальний дескриптор визначається як різниця між відстанями від ключової точки до внутрішнього та зовнішнього кілець. Орієнтаційний дескриптор визначається на основі напрямку градієнту на околицях ключової точки. Обидва дескриптори об'єднуються в один бінарний вектор фіксованої довжини [9-11].

Третій етап – порівняння бінарних дескрипторів для знаходження відповідностей між ключовими точками на двох зображеннях. Для порівняння використовуються відстань Хеммінга, що вимірює кількість бітів, що відрізняють два бінарних вектори.

Метод BRISK має кілька переваг порівняно з іншими методами детектування та опису особливостей зображень.

Швидкість: BRISK є одним з найшвидших методів детектування та опису особливостей. Це досягається за рахунок використання бінарних дескрипторів та бінарної операції порівняння, що дозволяє прискорити обчислення та порівняння дескрипторів.

Інваріантність до масштабу та повороту: BRISK використовує орієнтаційні пікселі для розпізнавання орієнтації та інваріантних описів особливостей. Це дозволяє отримати інваріантність до повороту та масштабу зображення.

Робастність до шуму та змін освітлення: BRISK використовує локальні відношення інтенсивності пікселів, що дозволяє зменшити вплив шуму та змін освітлення на описи особливостей.

Низька вимога до пам'яті: BRISK використовує зведення зображення до масштабу  $1/2$ , що дозволяє зменшити вимоги до пам'яті та збільшити швидкість обчислень.

Відсутність проблеми неоднорідності густини точок: BRISK використовує адаптивну сітку для детектування точок, що дозволяє уникнути проблеми неоднорідності густини точок.

Переходячи до наступного етапу – обчислення дескриптору, варто зауважити, що зразок вибірки BRISK складається з концентричних кілець. Область навколо КТ розбивається на 60 ділянок  $p$ , приклад на рисунку 1.3.

До кругових околиць потенційних КТ застосовується Гаусове згладжування. Червоний кружок на рисунку 1.3 ілюструє розмір стандартного відхилення фільтра Гауса, що застосовується до кожної точки вибору. Для визначення напрямку ключовий точки використовується сума локальних градієнтів [12].

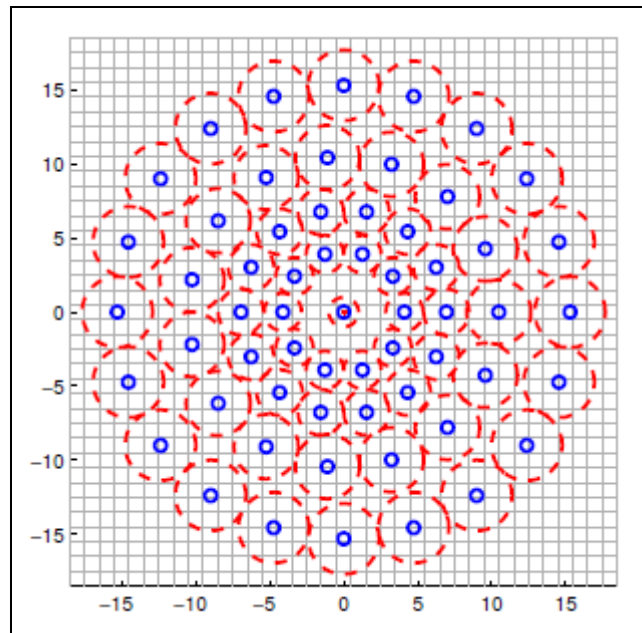


Рисунок 1.3 – Шаблон вибірки BRISK

Дескриптор складається з бінарних рядків довжиною 512, заповнених результатами проведених тестів в множині КТ. BRISK приймає набір коротких пар, що обертаються пар відповідно до орієнтації,

$$b = \begin{cases} 1, I(p_j^\alpha, \sigma_j) > I(p_i^\alpha, \sigma_i) \\ 0, I(p_j^\alpha, \sigma_j) \leq I(p_i^\alpha, \sigma_i) \end{cases}, \quad (1.1)$$

де  $I$  – згладжена інтенсивність у околиці радіусу  $\sigma_i$  відповідної точки  $p_i$  вибірки за відповідним стандартним відхиленням;

$\alpha = \arctan 2(g_y, g_x)$  – кут напрямку градієнта  $g$ .

Метод FREAK (Fast Retina Keypoint) призначений для виявлення та опису КТ на зображеннях. Метод FREAK використовує бінарні дескриптори для опису КТ на зображенні. Особливість цього методу полягає в тому, що він використовує інформацію про локальну структуру зображення, що дозволяє йому досягати високої точності виявлення ключових точок та ефективно використовувати ресурси обчислювальної системи.

Для визначення КТ метод FREAK використовує детектор особливостей SURF. Після виявлення КТ метод FREAK використовує описові орієнтації, отримані з методу SURF, щоб побудувати локальні піксельні дескриптори. Потім метод FREAK використовує бінарну нейронну мережу для кодування дескрипторів у бінарний вектор фіксованої довжини.

Метод FREAK має кілька переваг. По-перше, він працює швидше, ніж багато інших методів комп'ютерного зору, тому що використовує бінарні дескриптори. По-друге, бінарна форма дескрипторів дозволяє використовувати бінарні операції для порівняння дескрипторів та розпізнавання об'єктів, що значно спрощує обчислювальну складність задачі. По-третє, метод FREAK має дуже високу точність виявлення ключових точок на зображеннях з різними зашумленнями та розмірами.

Однак, метод FREAK має певні недоліки. Зокрема, він не є так точним, як метод SIFT, коли мова йде про опис дескрипторів. Крім того, метод FREAK погано справляється з детектуванням.

Метод AKAZE (Accelerated-KAZE) є одним з алгоритмів розпізнавання особливостей на зображеннях. Він базується на комбінації двох алгоритмів: алгоритму KAZE та алгоритму детектування особливостей MSER (Maximally Stable Extremal Regions) [13].

Ключовою особливістю AKAZE є його висока швидкість, адже він використовує методи швидкої конволюції та орієнтованих градієнтів. Це дозволяє знаходити та описувати особливості на зображеннях набагато швидше, ніж інші алгоритми. Крім того, AKAZE може використовуватися для роботи з великими наборами даних, адже він має низьку вимогу до обсягу пам'яті.

Алгоритм AKAZE використовує мультискалярний підхід для детектування особливостей, що дозволяє знаходити особливості на зображеннях з різними масштабами та роздільною здатністю. Для кожної знайденої особливості AKAZE визначає орієнтацію та розмір, що дозволяє зробити його роботу більш стійкою до різних перетворень [12].

Крім того, AKAZE використовує вдосконалену техніку опису особливостей, яка включає в себе вибір кращих орієнтаційних плоскостей та використання дескриптора BRISK. Цей дескриптор має високу швидкість та точність, що дозволяє AKAZE бути ефективним методом для розпізнавання особливостей на зображеннях.

Загалом, метод AKAZE є швидким та точним алгоритмом розпізнавання особливостей, який може бути використаний в різних додатках, включаючи комп'ютерне зір.

Для пошуку КТ застосовують алгоритм FED (Fast Explicit Diffusion) на пірамідальній схемі, що дозволяє побудувати багато-масштабну піраміду з нелінійним коефіцієнтом масштабування. Це забезпечує швидкість знаходження КТ завдяки застосуванню коефіцієнта, який змінюється при масштабуванні зображення, в порівнянні з гаусовою пірамідою [11, 12]. Обчислення цього коефіцієнта ґрунтується на зміні яскравості зображення

при масштабуванні. Для кожної октави застосовується детектор КТ  $L_i$  в піраміді обчислюється Гесіан:

$$L_{Hessian}^i = \sigma_{i,norm}^2 (L_{xx}^i L_{yy}^i - L_{xy}^i L_{xy}^i), \quad (1.2)$$

де  $\sigma_{i,norm} = \frac{\sigma_i}{2^{o^i}}$  – нормалізований щодо масштабу коефіцієнт, для обчислення

$L_{Hessian}^i$  з урахуванням розміру октави  $\sigma_i$ .

Похідні другого порядку обчислюються за допомогою фільтра Хара. Фільтр дозволяє враховувати орієнтацію особливих точок. За допомогою такого підходу відбувається пошук таких точок в октаві, значення фільтра яких вище заданого порогу і є найбільшим з околиці точки  $3 \times 3$  пікселів. Далі для кожної точки з потенційних максимумів порівнюється її значення щодо результатів в сусідніх октавах  $i+1$  та  $i-1$  у вікні розміром  $\sigma_i \times \sigma_i$  відповідно.

Алгоритм обчислення дескрипторів AKAZE подібний до BRISK, але зі змінами, які дозволяють обчислювати їх швидше [13-15]. Це досягнуто за рахунок додавання порівняння значень градієнтів яскравості по осі  $x$  та  $y$  до порівняння яскравості показників областей. Результат одного тесту складається з трьох бітів замість одного. Щоб забезпечити інваріантність до обертання та масштабування, вікно дескриптора орієнтується по орієнтації особливої точки, а розмір вікна обирається в залежності від розміру октави, в якій знайдено особливу точку. Щоб прискорити обчислення, тести проводяться не між середнім значенням всіх пікселів в області, а між заданою кількістю пікселів, залежно від розміру вікна. Результатом є бінарний дескриптор довжиною 486 по три складових.

ORB (Oriented FAST and Rotated BRIEF) – це алгоритм детекції та опису особливостей зображень, який був розроблений з метою створення

ефективного та точного методу розпізнавання об'єктів в режимі реального часу [12, 13].

ORB поєднує в собі два алгоритми – FAST для детекції особливостей та BRIEF для їх опису. Однак, відмінність ORB від попередників полягає в тому, що він також використовує додаткові методи для покращення якості детекції та опису особливостей.

Один з цих методів – це обчислення орієнтації ключових точок. Цей метод полягає в тому, що після вибору кандидатів на особливості за допомогою FAST, на кожен ключову точку накладається окільний вікінг, і в ньому шукається напрямок зміни яскравості на пікселях в цьому вікні. Орієнтація ключової точки обчислюється як середнє значення напрямків зміни яскравості на всіх пікселях в околі.

Інший метод – це додаткова фільтрація ключових точок за допомогою детектора Харріса. Цей детектор використовується для відсіювання несуттєвих точок та зменшення кількості особливостей.

Опис ключових точок в ORB здійснюється за допомогою дескриптора BRIEF, який містить бінарну послідовність 0 та 1, що вказує на наявність або відсутність зміни яскравості між пікселями в околі точки. Проте, ORB використовує модифікований варіант BRIEF, який називається гBRIEF.

Принцип роботи фільтром Харріса в тому, що це метод для виявлення кутів на зображенні. Він базується на відстеженні змін яскравості в маленьких локальних областях на зображенні.

Принцип роботи фільтра полягає у порівнянні інтенсивності пікселів в двох напрямках: вертикальному та горизонтальному. Для кожного пікселя в окрузі розміром  $N \times N$  (де  $N$  – невелике число, наприклад, 3 або 5) обчислюється сума квадратів різниць інтенсивності у двох напрямках. Це значення називається «матрицею  $M$ ».

Далі, за допомогою вивчення екстремумів характеристичного рівня визначається кут на зображенні. Якщо  $M$  має два власних числа  $\lambda_1$  та  $\lambda_2$ , то значення різниці між цими числами може вказувати на наявність кута на

зображенні. Це можна зробити, наприклад, порівнюючи значення мінімального власного числа  $\lambda_2$  з певним пороговим значенням.

В результаті роботи фільтра Харріса на зображенні будуть визначені точки, які можуть вважатися кутами. Вони можуть бути використані, наприклад, для подальшого виявлення особливостей на зображенні. У ORB радіус кола, яке досліджується навколо кожного пікселю, приймається рівним 9, оскільки саме ця відстань виявилася найбільш ефективною з точки зору продуктивності [16, 17].

Алгоритм ORB за замовчуванням може детектувати до 500 особливих точок. Якщо їх більше, то використовується детектор кутів Харріса, щоб вибрати найбільш значущі точки. Спочатку застосовується низький поріг для детекції більшої кількості точок, а потім вони упорядковуються за допомогою метрики Харріса, і відбираються перші найкращі точки. Для побудови дескриптора використовується модифікований алгоритм BRIEF (спрямований дескриптор BRIEF), що забезпечує інваріантність до масштабування за допомогою піраміди Гауса, яка складається з октав з початковим зображенням і з лінійним кроком. Кутова орієнтація вводиться для стійкості детекції при обертанні об'єкта, і вона базується на напрямках градієнта яскравості відносно центру точки, призначаючи напрямок з найбільшою інтенсивністю орієнтацією особливої точки.

Дескриптор представляється у вигляді вектора довжиною 256, що складається з результатів бінарних тестів навколо особливої точки. В околиці порівнюються середні значення яскравостей між  $x$  та  $y$ , де  $x, y$  – області  $5 \times 5$  пікселів:

$$\tau(I; x, y) := \begin{cases} 1: I_x < I_y \\ 0: I_x \geq I_y \end{cases}, \quad (1.3)$$

де  $I$  – середня яскравість обраної області.

Всі  $n = 256$  наборів  $x_i$  та  $y_i$  формують матрицю  $S$  розмірністю  $2 \times n$ .

Далі  $S$  за допомогою матриці повороту  $R_\theta$  орієнтується відповідно до кута  $\theta$ :  $S_\theta = R_\theta S$ . А сам вектор дескриптора записується як:

$$g_n(I, \theta) := f_n(I) | (x_i, y_i) \in S_\theta, \quad (1.4)$$

де  $f_n(I) := \sum_{i=1}^n 2^{i-1} \tau(I; x_i, y_i)$ .

Алгоритм ORB виробляє майже вдвічі більше дескрипторів, ніж SURF, проте кількість дескрипторів можна контролювати. Завдяки коротшим дескрипторам з 32 елементів, час розпізнавання дескрипторів ORB на порядок менший, ніж у SURF [18].

### 1.3 Класифікація зображень за множиною дескрипторів

Метою класифікації є надання швидкого та надійного виявлення, опису та зіставлення образів. Оцінка продуктивності всіх цих етапів є важливою. Алгоритм BRISK можна порівняти з SIFT та SURF [19, 20]. На рисунку 2.1 показані криві точність-відгук, використовуючи порівняння подібності з порогом для вибору різних пар зображень з різних наборів даних. Пороги виявлення налаштовуються таким чином, щоб вони видаляли приблизно однакову кількість кореспонденцій для кожного з алгоритмів. Результати виявлення різними детекторами проаналізовано у [20], де дескриптори вилучалися з регіонів, отриманих за допомогою детектора Fast-Hessian.

BRISK демонструє конкурентоздатність з SIFT та SURF на всіх наборах даних і навіть перевищує їх у деяких випадках. Зниження продуктивності BRISK на наборі даних Trees пов'язане з продуктивністю детектора: у той час як SURF виявляє 2606 та 2624 регіонів у зображеннях, BRISK виявляє лише 2004 регіони у зображенні 4 порівняно з 5949,

знайденими у зображенні 1, щоб отримати приблизно однакову кількість кореспонденцій. Те саме стосується до іншого набору даних із розмиттям, Vikes: яскравість, оцінена за допомогою FAST, є більш чутливою до розмиття, ніж детектори подібних до крапки. Також показуємо оцінку дескрипторів BRISK, отриманих з областей, визначених за допомогою SURF для набору даних Trees.

Метою класифікації зображень є швидке та надійне виявлення, опис та порівняння образів. У цьому контексті було порівняно алгоритм BRISK з SIFT та SURF на різних наборах даних. За результатами порівняння, яке було проілюстровано на рисунку 1.4, виявлено, що BRISK конкурентоздатний з SIFT та SURF на всіх наборах даних, а в деяких випадках перевищує їх. Проте, продуктивність BRISK була нижчою на наборі даних Trees, що пов'язано з продуктивністю детектора. У порівнянні з SURF, BRISK виявив меншу кількість регіонів у зображенні, що призвело до меншої кількості кореспонденцій. Аналогічно, на наборі даних Vikes, яскравість, виміряна за допомогою FAST, була більш чутливою до розмиття, ніж детектори, подібні до крапки. Дослідження також включало оцінку дескрипторів BRISK, отриманих з областей, визначених за допомогою SURF на наборі даних Trees.

Метод BRISK – це метод детекції, опису та зіставлення ключових точок на зображенні. Формула обчислення коефіцієнта вагомості елемента опису використовується для визначення важливості кожної ознаки в описі залежно від її корисності для класифікації зображень. Ця формула може бути використана для оптимізації процесу класифікації зображень за допомогою методу BRISK, дозволяючи зменшити вплив менш корисних ознак і підвищити точність класифікації.

Структурні методи використовують дескриптори ключових точок як характеристики об'єкта, що дозволяє визначити його клас на основі оцінок для набору дескрипторів. Однією з ключових особливостей дескрипторів є їх інваріантність до геометричних перетворень візуальних об'єктів, яка є критично важливою у структурних методах [19-21].

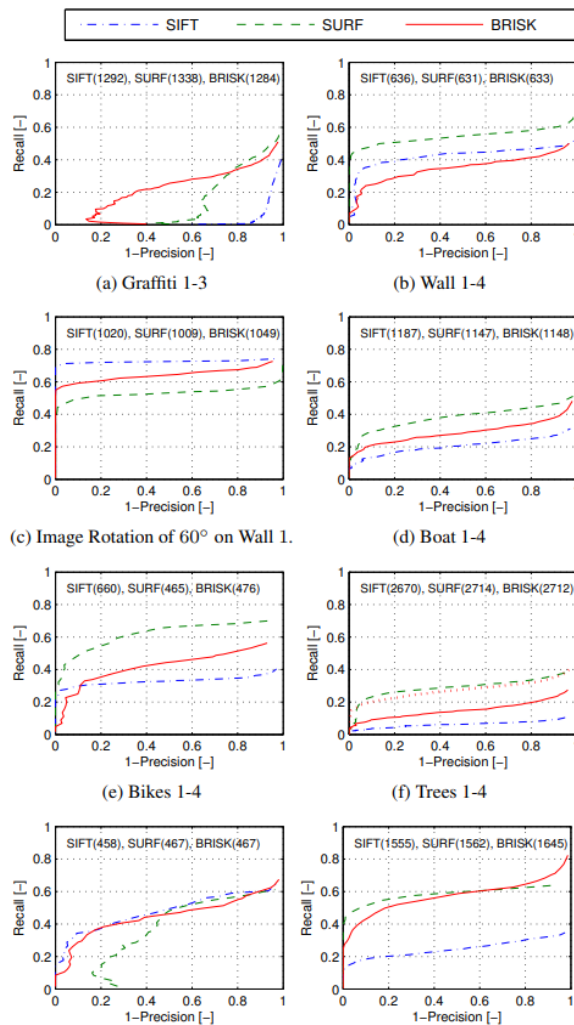


Рисунок 1.4 – Результати оцінки, що показують криві точність-відгук (з урахуванням всіх етапів виявлення, видобутку та відповідності) для BRISK, SURF та SIFT

Визначимо множину дескрипторів бази зображень як  $D = \{D^j\}_{j=1}^J$ , де  $D^j$  – опис еталону, а  $J$  – число класів. Зазначимо, що  $D \subseteq B^n$ , де  $B^n$  – простір бінарних векторів розмірності  $n$ . Додатково позначимо кількість елементів у описі еталону:  $D^j$  як  $s(j) = \text{card } D^j$ . Кількість  $J$  еталонів визначає число класів, що підлягають розпізнаванню.

Було використано термін «центр опису», що означає бінарний вектор, який обчислюється для заданої множини  $D^j$ . ORB-дескриптори мають бінарний вид [16, 17], для кожного еталону  $D^j$  можна визначити бінарний вектор  $d_j$  його центру [18, 22-24]

$$d_j(a) = \begin{cases} 1, & f(D^j, a) \geq s(j) / 2 \\ 0, & f(D^j, a) < s(j) / 2 \end{cases}, \quad (1.5)$$

де  $d_j(a)$  – значення біту за індексом  $a$  для центра опису  $d_j$ , а діапазон значень  $a$  для ORB-дескриптору  $a = 1, \dots, 256$ . Функція  $f(D^j, a)$  визначає число одиничних бітів з індексом  $a$  у множині дескрипторів еталону  $D^j$ .

Функцію  $f(D^j, a)$  можна обчислити додаванням бітів:

$$f(D^j, a) = \sum_{d=1}^{s(j)} x_d(a), x_d \in D^j, \quad (1.6)$$

де  $x_d(a)$  – біт за індексом  $a$  для дескриптора з номером  $d$ .

Для оцінки впливу дескриптора можна використовувати ваговий коефіцієнт, який відображає значимість ознаки в порівнянні з іншими елементами складу опису [12]. Цей коефіцієнт можна використовувати для кожного окремого біту центру опису. Для цього визначається нормований вектор, що дозволяє відображати вагу кожного біту в описі  $\lambda = \{\lambda_i\}_{i=1}^n$ :

$$\lambda_i(D^j) = \frac{1}{s(j)} \sum_{d=1}^{s(j)} x_d(i), \lambda_i \in [0, 1], \quad (1.7)$$

де  $i$  – номер біту у описі  $D^j$ .

Оскільки в бінарному векторі значення 0 та 1 мають однакову значимість, отже можемо ввести ваговий коефіцієнт для кожного біту центру опису, щоб відобразити їх класифікаційну значимість та відносну важливість у порівнянні з іншими ознаками опису. Для цього можна визначити нормований вектор, що містить вагові коефіцієнти для кожного окремого біту центру опису.

$$w_i(D^j) = \begin{cases} \lambda_i, d_j(i) = 1 \\ 1 - \lambda_i, d_j(i) = 0 \end{cases}. \quad (1.8)$$

Для кожного біту центру опису  $i$  обчислюється ваговий коефіцієнт, який відображає його важливість та ймовірність зустрічі в множині дескрипторів. Модель (1.9) дозволяє встановити більш значущі біти, які зустрічаються частіше, та менш значущі біти, які зустрічаються рідше [21].

Використовуємо отримані значення векторів вагових коефіцієнтів центрів для класифікації зображень. Щоб оцінити схожість між дескриптором вхідного зображення та центром еталону, використовуємо евклідову відстань для нецілих даних та відстань Хемінга для бінарних даних. У процесі класифікації беремо до уваги зважену відповідність дескриптора до кожного центру. Тепер при класифікації враховуємо не тільки кількість голосів дескрипторів, але і ваговий коефіцієнт для кожного центру [18, 25-27].

Для використання у бінарному класифікаторі можна обчислити відповідність шляхом додавання вагових коефіцієнтів, які відповідають бітам, що відрізняються між дескриптором та центром опису.

$$p(b, d_j, w) = \sum_{i=1}^n \begin{cases} 0, b(i) = d_j(i) \\ w_i, b(i) \neq d_j(i) \end{cases}, \quad (1.9)$$

де  $b$  – бінарний вектор об'єкту,  $b \in B^n$ . Відповідність (1.10) має сенс зваженої метрики Хемінга, так як це значення наближається до 0 зі зростанням числа однакових бітів.

Застосування ваг  $\lambda_i$  вектора  $\lambda$  при обчисленні манхеттенської метрики між аналізованим бінарним дескриптором та вектором  $\lambda$ :

$$\sigma(b, d_j, \lambda) = \sum_{i=1}^n |\lambda_i - b(i)|. \quad (1.10)$$

## 1.4 Постановка задачі

Дослідження методів класифікації та оцінки ознак на множині ключових точок, є актуальною для швидкісного розпізнавання зображень. Тому ставиться завдання дослідження методу відбору ознак, на основі показнику вагомості для дескрипторів ключових точок.

Об'єктом роботи є дослідження методів класифікації та кластеризації описів за допомогою відбору показників вагомості.

Метою цієї роботи є застосування та дослідження способів класифікації зображень, за рахунок обрахунку відстаней Хеммінга в множині дескрипторів ключових точках та за параметрами вагомості.

Для реалізації цієї мети необхідно виконати наступні завдання:

- вивчення методів опису зображень та алгоритми їх обробки;
- розробка та реалізація алгоритмів класифікації зображень з використанням коефіцієнтів вагомості для елементів опису;
- створення програмної реалізація алгоритму класифікації вхідного зображення за допомогою обчислення відстані в межах фіксованої бази еталонів;
- проведення аналізу результативності розробленого алгоритму класифікації зображень.

## 2 МОДЕЛІ ДЛЯ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ ВАГОМОСТІ

Деякі з моделей класифікації зображень, які можна використовувати для даної теми, включають:

Convolutional Neural Networks, CNN – це тип нейронних мереж, який використовується для обробки зображень. Вони вміють автоматично виявляти візуальні ознаки на зображеннях, такі як різні форми і текстури.

Нейронні мережі з довільною архітектурою (Deep Neural Networks, DNN) – це клас нейронних мереж, які мають багато прихованих шарів і здатні навчитися виявляти складні залежності в даних. Вони можуть бути використані для класифікації зображень, які містять багато різних візуальних ознак.

Random Forest – це алгоритм машинного навчання, який використовує дерева рішень для класифікації даних. Він може бути використаний для класифікації зображень, якщо вони містять достатньо візуальних ознак, які можуть бути використані для прийняття рішення про класифікацію.

Support Vector Machine – це алгоритм машинного навчання, який може бути використаний для класифікації зображень, якщо вони містять візуальні ознаки, які можуть бути перетворені на числові вектори. Він використовує гіперплощини для розділення даних на класи.

Binary Robust Invariant Scalable Keypoints – це метод для детектування та опису ключових точок зображень, який може бути використаний в задачах класифікації зображень.

### 2.1 Коефіцієнти вагомості для елементів опису

Коефіцієнти вагомості (інформативності) для елементів опису – це значення, які показують важливість кожного елемента опису при

класифікації зображень. Кожен елемент опису, наприклад, ознака зображення, може мати різну значимість для конкретної задачі класифікації. Коефіцієнти вагомості дозволяють враховувати цю значимість при виборі найважливіших ознак для класифікації [26-29].

Коефіцієнти вагомості можуть бути визначені різними методами, такими як статистичний аналіз, кореляція, машинне навчання та ін. [30-33]. В залежності від конкретної задачі класифікації та використовуваних методів опису та обробки зображень, коефіцієнти вагомості можуть бути обчислені за різними моделями.

Використання вагових коефіцієнтів інформативності для елементів опису дозволяє підвищити точність класифікації та зменшити час обробки зображень, оскільки вибір найважливіших ознак дозволяє скоротити розмірність ознакового простору та покращити якість класифікації [34].

Проведемо аналіз різних методів визначення коефіцієнтів вагомості та розроблено алгоритм класифікації зображень з використанням цих коефіцієнтів.

Застосування коефіцієнтів вагомості дозволяє враховувати важливість кожного елемента опису при класифікації зображень. Це знижує кількість неважливих ознак та дає зосередитися на найбільш інформативних, що зменшує обчислювальну складність та покращує якість класифікації.

З іншого боку, визначення коефіцієнтів вагомості може виявитися складним та трудомістким процесом, особливо при роботі з великими наборами даних. Крім того, залежно від методу визначення коефіцієнтів вагомості, можуть бути отримані різні результати та різні точності класифікації [35-38].

Отже, використання коефіцієнтів вагомості є ефективним підходом для поліпшення класифікації зображень. Проте, потрібно уважно підходити до вибору методу визначення коефіцієнтів вагомості та аналізувати отримані результати класифікації (рис. 2.1).

Плюси	Мінуси
Підвищення точності класифікації	Збільшення часу обробки зображень
Зменшення кількості признаков	Потреба в додаткових обчислювальних ресурсах
Покращення якості класифікації	Вимога до якості і точності вхідних даних
Зменшення розмірності простору ознак	Чутливість до шуму та випадкових аномалій в даних
Оптимізація роботи систем обробки зображень	Потреба в додатковій підготовці даних

Рисунок 2.1 – Плюси та мінуси коефіцієнтів інформативності

Значущість може бути ключовим фактором у задачах класифікації, оскільки дозволяє класифікатору зосередитися на найбільш інформативних або дискримінаційних характеристиках дескриптора, що може значно покращити результативність методу або алгоритму.

Кожній ознаці  $z_v^k$  еталонного опису значущість можна також призначити експертним способом, статистично чи шляхом навчання, отримавши деякий ваговий коефіцієнт  $0 \leq \lambda \leq 1$ .

Коефіцієнт значущості відображає важливість певної ознаки у порівнянні з іншими ознаками опису в задачах класифікації. Обчислення вагових коефіцієнтів передбачає оцінювання впливу кожної ознаки на результат класифікації. Зазвичай, в процесі класифікації використовуються інтегральні показники значущості опису на основі вагових коефіцієнтів складових елементів. Це дозволяє зосередитися на найбільш інформативних ознаках та покращити кінцевий результат класифікації.

Таким чином, кожен елемент  $z_v^k$  опис або блок опису може бути оцінений за його інформативністю на основі інших елементів у множині, включаючи елементи з різних класів. Це можна зробити шляхом порівняння

відстаней між елементом та його власним описом з відстанями до інших еталонних описів [35, 39-41]. На основі цього для множини, що складається з  $N$  еталонних описів, кожен з яких містить  $s$  елементів, можна побудувати матрицю.

$$\Lambda = \begin{Bmatrix} \lambda_{1,1} \cdots, \lambda_{1,s} \\ \lambda_{2,1} \cdots, \lambda_{2,s} \\ \cdots, \cdots, \cdots \\ \lambda_{N,1} \cdots, \lambda_{N,s} \end{Bmatrix}. \quad (2.1)$$

Матриця  $\Lambda$  містить значення інформативності, обчислені для усіх елементів бази еталонів і відображає класифікаційну значущість кожного із  $N \times s$  базових дескрипторів.

Функція належності може бути використана для визначення значущості окремого дескриптора в множині класів. Вона може бути розглянута як міра нечіткості, яка вказує на ступінь відповідності компонента даних певному поняттю, що формалізується як нечітка множина. Для окремої компоненти структурного опису зображення, функція належності може бути представлена у вигляді вектора  $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_N)$  з цілими чи дійсними елементами, що відображають ступінь релевантності дескриптора для спектру класів. Як правило, приймають умову  $\sum_{i=0}^N \varphi_i = 1$ .

Можна створити матрицю класифікаційної значущості для множини дескрипторів, використовуючи базу еталонних описів та масиви ключових точок. Для порівняння дескрипторів об'єкта з еталонними, потрібно враховувати класифікаційну значущість. Якщо дескриптори схожі, то ключова точка є впливовішою. Застосовуючи вагові коефіцієнти, можна використовувати «зважені» дескриптори ключових точок у класифікації зображень. Класифікатор буде використовувати ці вагові коефіцієнти для визначення схожості зображень.

Існує кілька сучасних методів для створення класифікатора. Один з них полягає в застосуванні методів зменшення розмірності, таких як аналіз головних компонент (PCA) або лінійний дискримінантний аналіз (LDA), щоб зменшити кількість дескрипторів та зберегти при цьому відмінності між зображеннями.

Інший підхід полягає у використанні метрики для порівняння дескрипторів. Наприклад, алгоритм ORB зазвичай використовує відстань Хеммінга, яка вимірює кількість різних бітів у двійковому коді дескрипторів. Застосовуючи вагові коефіцієнти, можна також використовувати зважену евклідову відстань для порівняння дескрипторів відстань

$$d(x, y) = \sqrt{\sum_{j=0}^J \lambda_j (x_j - y_j)^2}, \text{ де } \lambda_j \text{ – ваговий коефіцієнт дескриптора з}$$

матриці  $A$ . Для класифікації зображень можна використовувати різні методи, такі як зменшення розмірності за допомогою аналізу головних компонент або лінійного дискримінантного аналізу, або метрики для порівняння дескрипторів, такі як відстань Хеммінга або зважена евклідова відстань. Після порівняння дескрипторів з еталонними можна визначити клас, набравши найбільшу кількість голосів чи найбільшу агреговану значущість на множині класів. Інші можливі варіанти класифікатора включають логічну обробку величин значущості чи метрики, а також застосування інтегрального критерію на основі цих параметрів.

## 2.2 Вагомість інтегрованих ознак

Створено систему ознак, яка буде інтегрована в межах окремих фрагментів, використовуючи однобітові розподіли. Для досягнення цього знайдемо суми 1-бітових фрагментів в кожному стовпці матриці опису, і таким чином отримаємо вектор  $t = (t_1, \dots, t_j, \dots, t_n)$ , де  $t_j = \sum_{i=1}^{s\Sigma} d_{i,j}$ ,  $j = \overline{1, n}$ . Для створення системи ознак, яка буде інтегрована в межах окремих

фрагментів, можемо використати однобітові розподіли. Щоб цього досягнути, знайдемо суми 1-бітових фрагментів в кожному стовпці матриці опису, що представляє 1-бітовий фрагмент, і таким чином отримати вектор ознак для кожного фрагменту. Цей вектор може бути використаний для подальшої обробки та класифікації фрагментів.

$$u_k = \sum_{j=k}^{k+b-1} t_j, \quad (2.2)$$

де  $b = n / m$  – розмір,

$k = 1, b+1, 2b+1, \dots, n-b+1$  – номер фрагмента.

Ознаки (2.1) реалізують крос-кореляційне оброблення матриці  $D$  з прямокутною маскою розміром  $b \times s$ . У результаті обчислення (2.1) отримуємо цілочисельний вектор  $u_k$  розмірністю  $m$ . Параметр  $m$  є характеристикою системи фрагментів, він зменшується від  $n$  до 1 зі збільшенням розміру фрагмента від 1 до  $n$ .

Значення вектора  $u = (u_1, \dots, u_k, \dots, u_m)$  можуть бути використані як самостійні структурні ознаки статистичного виду. Опрацювання (2.1) залежить від параметра  $b$  та реалізує просторове оброблення (інтегрування даних) на множині дескрипторів. Зважаючи на таку просту модель обчислення функцій (2.3), всі вони для довільного розміру фрагмента визначаються нескладно (логічно або додаванням цілих чисел), базуючись на значеннях статистичного розподілу.

На основі (2.1) може бути застосовано також ієрархічний спосіб розпізнавання, що використовує для зіставлення з еталонами систему ознак  $u_k$  з різним ступенем інтегрування даних, що визначається розміром блока.

Діапазон цілих значень для ознак  $u_k$  безпосередньо можна визначити за розміром фрагмента як  $u_k \in \{0, \dots, sb\}$ . Загалом вектор  $u$  є інтегрованою

характеристикою опису, його можна окремо нормувати за кількістю дескрипторів або розміром фрагменту.

Обсяг значень для ознак (2.3) змінюється від одного числа до вектора розміру  $n$  (розмір дескриптора) і зменшується зі збільшенням розміру фрагмента. При  $m=1$  маємо  $b=n$ , тобто кожний дескриптор містить один фрагмент, а величина  $u_1$  визначається одним значенням і дорівнює сумі бітів усього опису. При  $m=n$  маємо іншу граничну ситуацію:  $n$  1-бітових фрагментів, а  $u_k$  обчислюється як сума стовпця матриці  $D$ .

Модель (2.3) реалізує процедуру зменшення інформаційної надмірності просторового сигналу за рахунок допустимого зниження (з точки зору якості розрізнення) ступеня розподільної здатності системи ознак подання опису.

Оброблення виду (2.3) є різновидом просторового аналізу значень для множини дескрипторів вздовж координат їх компонентів. Іншими варіантами такого оброблення є фільтрація чи розклад дескриптора за системою ортогональних функцій [40-44].

Правила базуються на математичних методах аналізу даних, які дозволяють оцінити та зменшити очікувану величину помилки розпізнавання. Більшість методів, як детерміновані, так і статистичні, належать до дискримінантного аналізу, який є одним з розділів багатовимірного статистичного аналізу даних.

У синтаксичних правилах, множина образів одного класу задається породжувальною процедурою. Для визначення, чи належить образ до класу, необхідно розібрати його за допомогою правил породжувальної процедури. В синтаксичному розпізнаванні ймовірність помилок залежить від властивостей аналізованих даних та правильності опису класів. Синтаксичні правила використовуються в складних конфігураціях образів, які складаються з атомарних елементів, між якими існують певні зв'язки. Зв'язки різних типів можуть бути описані за допомогою символів-дескрипторів, в тому числі й при аналізі сцен [45].

### 2.3 Статистичне та блокове подання описів

Статистичні моделі дедалі більше застосовуються в інтелектуальних методах аналізу даних, включаючи класифікацію зображень. Ці моделі дозволяють виявляти закономірності та створювати системи знань на основі даних [21]. Одна з переваг статистичних моделей полягає у використанні узагальненої інформації про властивості класів розпізнаваних об'єктів для прийняття рішення. Це дозволяє більш ефективно враховувати особливості об'єктів у просторі ознак. Є декілька підходів до статистичної моделювання, один з яких використовує деякі статистичні характеристики, такі як математичне очікування, дисперсія, тощо. Однак, щоб отримати найточніший результат, краще використовувати безпосередньо розподіл даних. Це дозволяє більш чутливо враховувати відмінності та особливості значень даних, що відображають властивості зображень, що розпізнаються [46].

Бітовий склад дескрипторів КТ у просторі  $B^n$  бінарних векторів (де  $n$  – ступінь двійки) дає можливість запровадити аналіз дескриптора як кортежу послідовних елементів, діапазон значень яких відомий. Ця методика дозволяє створювати синтезований простір даних, що містить структурний опис (набір бінарних векторів), та проводити статистичний аналіз розпізнаваних зображень, враховуючи властивості складу еталонних описів. Усі елементи мають однакову вагу, але їх розташування в дескрипторі чітко визначене, що дозволяє аналізувати послідовності елементів. Така структура дозволяє використовувати інтелектуальний аналіз для вирішення питання про належність об'єкту до класу еталону.

Опис  $Z$  розпізнаваного візуального об'єкту визначимо у вигляді скінченної множини  $Z = \{z_v\}_{v=1}^s$ ,  $z_v \in B^n$ ,  $Z \subset B^n$  з  $s$  бінарних дескрипторів КТ, наприклад, сформованих детекторами ORB, BRISK, AKAZE, які розглядалися у підрозділі 1.2.

Для фіксованого числа  $n$  подамо опис об'єкту бінарною матрицею  $D = \{\{d_{i,j}\}_{i=1}^s\}_{j=1}^n$ , сформовану довільною послідовністю отриманих детектором  $s$  дескрипторів КТ.

Для розв'язання задачі розпізнавання та класифікації представимо множину дескрипторів у вигляді матриці  $s \times n$ :

$$D = \begin{pmatrix} d_{11} & \dots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{s1} & \dots & d_{sn} \end{pmatrix}, \quad (2.3)$$

де  $s$  – обрана кількість дескрипторів,

$$s = 1, 2, \dots, 500, \quad d_{sn} \in [0; 1].$$

Для спрощення аналізу припускаємо, що значення  $s$  однакові для всіх еталонів. Це можна досягти випадковим або цілеспрямованим відбором необхідної кількості дескрипторів. Кожен дескриптор КТ можна розбити на непересічні фрагменти, які послідовно слідуєть один за одним і повністю складають початковий розмір (рис. 2.2). Кожен елемент такого дескриптора може приймати значення певного діапазону, наприклад, якщо елемент представлений одним бітом, то маємо два можливих значення (0, 1), а якщо байтом – множину значень від 0 до 255.

елемент 1	елемент 2	...	елемент $m$
-----------	-----------	-----	-------------

Рисунок 2.2 – Ланцюжкова структура дескриптора даних

Отриманий новий простір даних має перевагу в тому, що дозволяє аналізувати внутрішню структуру та статистичні властивості опису, що дозволяє віднести його до одного з еталонних класів. Блочний спосіб формування ознак забезпечує додаткову гнучкість під час обробки.

Після розбиття  $\forall z_v \in Z$  отримуємо фіксовану структуру  $z_v = z_v^1 \& z_v^2 \& \dots \& z_v^m$  «зчеплених» в ланцюжок елементів, де  $z_v^k$  –  $k$ -й блок дескриптора  $z_v$ . Кожний із дескрипторів тепер складається із  $m$  однотипних бінарних даних, значення яких у послідовності можна записати кортежем цілих чисел. У результаті опис  $Z$  набуває виду матриці із  $s$  рядків по  $m$  елементів (число блоків) у рядку.

Оскільки кожен фрагмент має рівноцінне представлення у межах свого діапазону значень, а також відображає властивості об'єкту, то можемо скласти розподіли даних шляхом аналізу матриці [45, 46]. Для цього задаємо відображення  $\Omega: Z \rightarrow Q$ ,  $Z \subset B^n$  із множини бінарних векторів у множину  $Q$  цілочисельних векторів потужності  $w < n$ , що містять статистичні розподіли значень фрагментів. Відображення  $\Omega$  призначене забезпечити ідентифікацію та розрізнення об'єктів у синтезованому просторі.

Визначимо розподіл (рейтинг зустрічальності)  $q \in Q$  як вектор  $q = \{q_1, \dots, q_w\}$  цілих чисел, де  $q_i$  – це число значень для  $k$ -го фрагмента, що дорівнюють  $i$ ,  $i = \overline{1, w}$ , серед відповідних фрагментів усієї множини  $Z$ .

$$q_i = \text{card}\{z_v^k \in z_v, z_v \in Z \mid z_v^k = i\}. \quad (2.4)$$

Тоді виконується умова  $\sum_i q_i = s$ , так як сума значень ланок для розподілу  $q$  дорівнює загальному обсягу  $s$  дескрипторів опису  $Z$ . Зауважимо, що  $w$  – це число ланок розподілу, які належать діапазону  $0, \dots, w-1$ .

Зараз необхідно створити функцію, яка буде відображати множину бінарних векторів, які є дескрипторами КТ, у множину значень розподілів, які мають значно меншу потужність  $w$ , ці методи дозволяють проводити ідентифікацію та розрізнення візуальних об'єктів на основі їх описів. Для кожного з  $m$  фрагментів будується відповідний розподіл, який

використовується для аналізу та порівняння з еталонними розподілами, що відповідають певним класам візуальних об'єктів.  $q = \{q_1, \dots, q_w\}$ , величина  $w$  визначена діапазоном значень даних для фрагмента, що визначається його розміром. Наприклад, для дескриптора ORB при розбитті на байти для  $n = 256$  маємо  $m = 32$ ,  $w = 256$ . Загалом множина  $Z$  під дією відображення  $\Omega$  описується матрицею  $Q = \{\{q_{i,v}\}_{i=1}^w\}_{v=1}^m$ , що містить розподіли системи фрагментів.

Матриця  $Q$  та вектор  $Z$  дозволяють отримати кортеж розподілів значень складових опису та його статистичні властивості. Ці розподіли узагальнюють знання відповідно до основних принципів машинного навчання, що дозволяє використовувати їх для ідентифікації та розрізнення візуальних об'єктів [28, 30].

Одне із важливих завдань полягає у визначенні еквівалентності між об'єктами на основі їх структурних описів. Але наявність геометричних перетворень ускладнює процес визначення відповідності між описами. Тому вирішення проблеми інваріантності є важливим як на локальному рівні формування описів, так і на етапі порівняння описів [47]. У цьому контексті визначення відповідності між множинами структурних описів зводиться до оцінки ступеня їх еквівалентності. У простіших випадках можна вирішити задачу визначення відповідності на основі еквівалентності векторів.

Важливо підкреслити, що в формулі (2.3) обчислення виконуються тільки з цілими числами, а значення  $r$  збільшується тільки через додавання додатних значень. Також важливо зауважити, що умова використання тільки цілих значень розподілів базується на однаковості потужностей описів.

Інформація, що міститься у розподілах окремих бітів чи їх пар, може бути недостатньою для досягнення необхідного рівня розрізненості між об'єктами, що порівнюються. У цьому випадку доведеться збільшувати кількість обчислень. Для спрощення аналізу і оброблення ланцюжкових даних можна використовувати значення окремих параметрів побудованих розподілів. Для визначення релевантності цього підходу можна обчислювати

відстань між векторами або проводити по-елементний аналіз. Аналіз, що описаний у формулі (2.3), базується на принципі максимуму апостеріорної ймовірності і може досягнути необхідного результату для значно відрізняються об'єктів [47].

Представлення та аналіз даних у вигляді множини фрагментів стали популярними в теорії комп'ютерного зору, оскільки це дає можливість робити рішення на основі окремих візуальних елементів. Однак, визначення оптимального розміру елемента, порогів для визначення еквівалентних за описом елементів та встановлення значущості опису елемента, повинно виконуватися користувачем, виходячи з бази зображень, яка використовується для розпізнавання.

Основною моделлю зіставлення ХО в більшості систем структурного розпізнавання є порівняння числових векторів  $\lambda$  в лінійному нормованому просторі  $R^n$ ,  $\lambda \in R^n$ . Так як метрики в цілому забезпечують більш прийнятний діапазон властивостей, ніж функції подібності, то основне застосування в роботі при зіставленні на локальному рівні отримали метричні характеристики і відповідні метричні підходи.

Метрики – важливий інструмент вирішення багатьох завдань розпізнавання образів та інтелектуального аналізу даних. Значення метрики пов'язане з ймовірними характеристиками віднесення елемента до класу [40]. обчислення метрики  $\rho_\Lambda(\Lambda^1, \Lambda^2)$  для множин  $\Lambda^1, \Lambda^2$  реалізується двоетапної процедури: спочатку обчислюються відстані  $\rho(\lambda^1, \lambda^2)$ ,  $\lambda^1 \in \Lambda^1, \lambda^2 \in \Lambda^2$  між елементами, а потім на основі множини  $\{\rho(\lambda^1, \lambda^2)\}$  визначається значення метрики для множин у вигляді функції  $\rho_\Lambda(\Lambda^1, \Lambda^2) = f(\{\rho(\lambda^1, \lambda^2)\})$  [43, 45].

Нехай  $\lambda, \lambda^0$  – вектори дескрипторів. В роботі було прийнято рішення використовувати Манхетенську відстань ( $R_1^n$ )

$$\rho(\lambda, \lambda^0) = \sum_{i=1}^n |\lambda_i - \lambda_i^0|. \quad (2.5)$$

Створення опису зображення у вигляді множини характеристичних ознак є одним з найбільш складних завдань у розробці систем структурного розпізнавання. Вибір підходящих ознак часто здійснюється евристичним способом [47]. Характеристичні ознаки відіграють важливу роль у вирішенні завдань класифікації об'єктів за їхніми загальними властивостями. На найнижчому рівні ієрархії структурного подання передбачається точне визначення характеристичних ознак об'єкта.

### 3 РЕЗУЛЬТАТИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ

#### 3.1 Вибір фреймворку та мови програмування

Для реалізації програмного застосунку було обрано мову об'єктно-орієнтованого програмування C# на базі фреймворку .NET 6 і бібліотека комп'ютерного зору OpenCV.

C# – це сучасна, об'єктно-орієнтована мова програмування, розроблена компанією Microsoft. Вона є частиною платформи .NET, яка надає універсальне середовище для розробки, виконання та управління програмами. C# призначена для створення різноманітних додатків, таких як консольні, настільні, веб-сайти, мобільні додатки, графічні програми тощо. Мова має синтаксис, що спирається на C++ та Java, що робить її зручною для вивчення програмістами, знайомими з цими мовами.

.NET 6 – це версія платформи .NET, випущена у листопаді 2021 року. Вона представляє собою єдину, уніфіковану платформу для розробки, яка замінила .NET Core та .NET Framework. Основні відмінності .NET 6 від попередніх версій полягають у покращеній продуктивності, збільшенні швидкості виконання коду, підтримці крос-платформеності та нових можливостях для розробників.

Основні особливості .NET 6 складаються з кількох ключових елементів. По-перше, .NET 6 володіє крос-платформеністю, підтримуючи ряд операційних систем, включаючи Windows, macOS, Linux, iOS, Android та інші. Це дає змогу розробникам створювати додатки, які можуть працювати на різних платформах з мінімальною потребою в зміні коду.

По-друге, .NET 6 включає покращення, які забезпечують високу продуктивність, особливо у швидкості виконання коду. Це веде до створення високопродуктивних додатків.

Третім ключовим елементом є зручність розробки. Платформа .NET 6 пропонує розробникам нові можливості, такі як нові синтаксичні

вдосконалення в C# 10, які забезпечують зручність написання коду. Крім того, є Blazor – фреймворк, що дозволяє створювати інтерактивні веб-додатки за допомогою C# замість JavaScript.

Четвертим нововведенням у .NET 6 є забезпечення безпеки та надійності. .NET 6 підтримує сучасні стандарти безпеки і має вбудовані механізми для управління помилками та винятками, що забезпечують надійність програмного забезпечення.

Останнім, але не менш важливим, є єдність платформи. .NET 6 об'єднує різні версії .NET, такі як .NET Core та .NET Framework, що спрощує процес розробки та розгортання додатків.

Усі ці особливості разом формують мову програмування C# та платформу .NET 6 як потужний інструментарій для розробки сучасних програмних рішень. Від крос-платформенності до зручності розробки, від високої продуктивності до забезпечення безпеки та надійності, C# та .NET 6 пропонують комплексний набір можливостей, що задовольняють вимоги сучасного програмного забезпечення. Ці технології продовжують розвиватися, враховуючи потреби ринку та відгуки від розробників, що забезпечує їх актуальність та привабливість для широкого кола професіоналів в галузі ІТ.

Для розробки програмного застосунку було використано середовище розробки Visual Studio (рис. 3.1).

Visual Studio – це інтегроване середовище розробки (IDE) від Microsoft, яке включає в себе потужний редактор коду. Він підтримує синтаксичне виділення, автозаповнення і навігацію по коду. Ці особливості спрощують написання коду, допомагають уникнути помилок і підвищують продуктивність розробника. Visual Studio також має вбудовані інструменти для відладки, що дозволяють розробникам стежити за виконанням коду, перевіряти стан змінних, встановлювати точки зупинки і багато іншого. Ці інструменти допомагають розробникам швидко знайти і виправити помилки в коді.

Важливим аспектом Visual Studio є інтеграція з системами керування версіями, такими як Git. Це означає, що розробники можуть відстежувати зміни у кодї, відновлювати старі версії та співпрацювати з іншими розробниками прямо в середовищі розробки. Крім того, Visual Studio тісно інтегровано з хмарною платформою Microsoft, Azure. Це полегшує розгортання та керування додатками в хмарі, що може значно спростити процес розробки та впровадження програмного забезпечення.

Visual Studio також включає в себе різноманітні інструменти для розробки інтерфейсу користувача. Розробники можуть створювати впізнаваний і дружній до користувача інтерфейс для своїх додатків. Нарешті, Visual Studio можна налаштовувати та розширювати за допомогою розширень. Це дозволяє розробникам адаптувати середовище розробки під свої потреби.

```

18     var projectPath = Directory.GetParent(Directory.GetCurrentDirectory()).Parent.Parent.FullName;
19     var imagePaths = new List<string>
20     {
21         projectPath + "\\Images\\karl1",
22         projectPath + "\\Images\\karl2",
23         projectPath + "\\Images\\karl3",
24     };
25
26     BriskDetector detector1 = new BriskDetector(imagePaths[0]);
27     BriskDetector detector2 = new BriskDetector(imagePaths[1]);
28     BriskDetector detector3 = new BriskDetector(imagePaths[2]);
29
30     var descriptors1 = detector1.Descriptors;
31     var descriptors2 = detector2.Descriptors;
32     var descriptors3 = detector3.Descriptors;

```

Рисунок 3.1 – Приклад коду в середовищі розробки

OpenCV, відкрита бібліотека програмного забезпечення, є ключовим інструментом для спеціалістів у галузі комп'ютерного зору та машинного навчання. Ця багатофункціональна бібліотека включає понад 2500 оптимізованих алгоритмів, які покривають широкий діапазон методів обробки зображень, алгоритмів комп'ютерного зору та машинного навчання.

Однією з сильних сторін OpenCV є її здатність до роботи з дескрипторами зображень – векторами особливостей, які описують

зображення або його частину. Ці дескриптори використовуються в широкому спектрі задач, включаючи розпізнавання образів та відстеження об'єктів. Крім того, OpenCV дозволяє створювати «склад опису» – базу даних дескрипторів зображень, яка є важливою для таких задач, як пошук зображень, класифікація зображень та розпізнавання обличчя.

OpenCV відрізняється своєю універсальністю, працюючи на різних операційних системах та архітектурах, включаючи Windows, Linux, MacOS, iOS, Android, x86 та ARM. Вона також взаємодіє з різними мовами програмування, включаючи C++, Python, Java, C#, що робить її доступною для широкого кола розробників.

OpenCV має активну спільноту розробників, що постійно допомагає у вдосконаленні бібліотеки та надає підтримку новим користувачам. Це особливо важливо для бібліотеки, яка є оптимізованою для високої продуктивності та може використовуватися в реальному часі для обробки зображень та відео – це робить її незамінною для застосунків, як-то системи відстеження об'єктів, автономні транспортні засоби та інтерактивні медіа.

OpenCV також активно використовує глибоке навчання в своїх модулях, інтегруючись з такими популярними бібліотеками глибокого навчання, як TensorFlow та PyTorch. Це відкриває доступ до сучасних алгоритмів комп'ютерного зору, що забезпечує більші можливості для розробки і вдосконалення програмних рішень.

Ще однією важливою особливістю OpenCV є інструменти для 3D реконструкції з 2D зображень. Вони включають методи для виявлення ключових точок, співставлення ключових точок та відновлення структури та руху. Ці інструменти відкривають нові можливості в області комп'ютерного зору, дозволяючи створювати тривимірні моделі на основі двовимірних зображень.

Як відкритий проект, OpenCV надає можливість розробникам вивчати вихідний код бібліотеки, адаптувати його до своїх потреб, а також внести власний вклад в його розвиток. Це допомагає збільшити гнучкість та

адаптивність бібліотеки, роблячи її ще більш корисною для широкого спектру задач, від простих операцій обробки зображень до складних задач, як розпізнавання образів, відстеження об'єктів та аналізу відео.

В цілому, всі ці особливості роблять OpenCV потужною бібліотекою для роботи з комп'ютерним зором та машинним навчанням (рис. 3.2).

The screenshot displays the NuGet package page for Emgu.CV. At the top, there is a navigation bar with links for Packages, Upload, Statistics, Documentation, Downloads, and Blog. Below this is a search bar. The main content area shows the package name 'Emgu.CV' with version '4.7.0.5276'. There are buttons for '.NET 6.0' and '.NET Standard 2.0'. A dark blue bar contains the command: `> dotnet add package Emgu.CV --version 4.7.0.5276`. Below this, there are tabs for README, Frameworks, Dependencies, Used By, Versions, and Release Notes. The 'Frameworks' tab is active, showing a grid of target frameworks. The 'Product' column lists '.NET'. The 'Versions' column lists various target frameworks, with 'net6.0-ios11.0' highlighted in blue.

Product	Versions
.NET	net5.0, net5.0-windows, net6.0, net6.0-android, net6.0-ios, net6.0-ios11.0
	net6.0-maccatalyst, net6.0-maccatalyst13.2, net6.0-macos, net6.0-tvos
	net6.0-windows, net7.0, net7.0-android, net7.0-ios, net7.0-maccatalyst
	net7.0-macos, net7.0-tvos, net7.0-windows

Рисунок 3.2 – NuGet Package для бібліотеки OpenCV

Вибір та налаштування алгоритмів попередньої обробки зображень: перед подальшими операціями вимагають певної підготовки. Налаштування цих алгоритмів може відрізнятися залежно від використовуваної моделі та типу зображень. Було використано бібліотеку Emgu.CV для обробки зображень за допомогою дескриптора BRISK. Для встановлення цієї бібліотеки необхідно підключити NuGet Package і щоб версія цього пакету мала підтримку версії .NET фреймворку на якому реалізована програма (рис. 3.3), для даної роботи було використано версію .NET 6 та версію бібліотеки 4.7.0.

Бібліотека Emgu.CV має обширний функціонал та забезпечує багато функцій (рис. 3.3) в тому числі виділення ключових ознак та дескрипторів, що і є необхідним для цієї роботи.

Emgu CV є .NET обгорткою для OpenCV, яка дозволяє використовувати всю потужність OpenCV безпосередньо з .NET мов. Це відкриває нові можливості для розробників, що працюють з .NET екосистемою, оскільки вони можуть застосовувати передові алгоритми комп'ютерного зору та машинного навчання безпосередньо у своїх додатках.

Однією з основних переваг Emgu CV є те, що вона робить OpenCV доступним для широкого спектра мов програмування, що включає C#, VB.NET, C++/CLI та F#. Це означає, що розробники можуть використовувати знайомі їм мови і інструменти для реалізації складних алгоритмів комп'ютерного зору та машинного навчання.

Ще однією важливою перевагою є те, що Emgu CV надає високий рівень абстракції, що спрощує роботу з OpenCV. Наприклад, ви можете використовувати зручні структури даних .NET, такі як List<> або Dictionary<>, замість роботи з нативними структурами даних OpenCV. Тобто колекції простору імен System.Collection.Generic що дозволить використовувати Generic рішення для цих колекції і не залучати процеси boxing/unboxing, що дозволяє значно прискорити процес виконання за рахунок скорочення кількості копіювання значимих значень колекцій.

З огляду на те, що Emgu CV є комерційним продуктом, він також пропонує підтримку та розширені можливості, як-то CUDA та OpenCL для прискорення обчислень на GPU. Це може бути важливим фактором для додатків, що вимагають високої продуктивності та швидкості обробки.

Все разом, Emgu CV є потужним інструментом, який забезпечує зручний доступ до функціоналу OpenCV для розробників на платформі .NET (рис. 3.3).

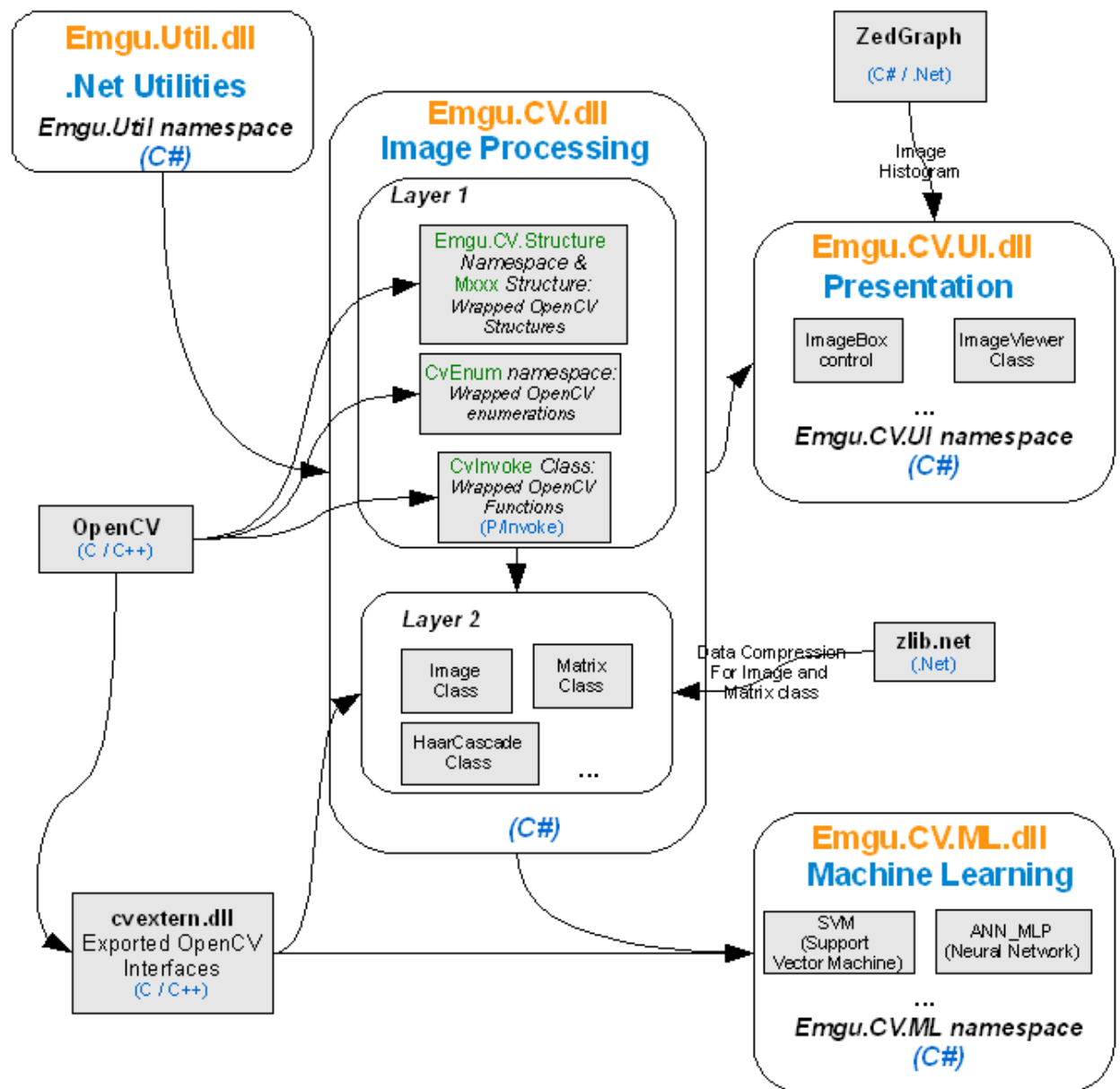


Рисунок 3.3 – Структура бібліотеки Emgu.CV

### 3.2 Аналіз особливостей програмної моделі

Процес моделювання методу класифікації зображень із застосуванням коефіцієнтів вагомості для елементів опису полягає в наступному.

Вибір наборів даних: для моделювання необхідно мати набір зображень, який включає навчальні та тестові набори. Набір використовується для навчання моделі, а тестовий – для оцінки її ефективності. Мій набір містив зображення персонажа Карлсона (рис. 3.4) з

різним розташуванням рук, голови та предметів, щоб людське око могло помітити різницю в зображеннях, але не детально порівняти їх одне з одним.

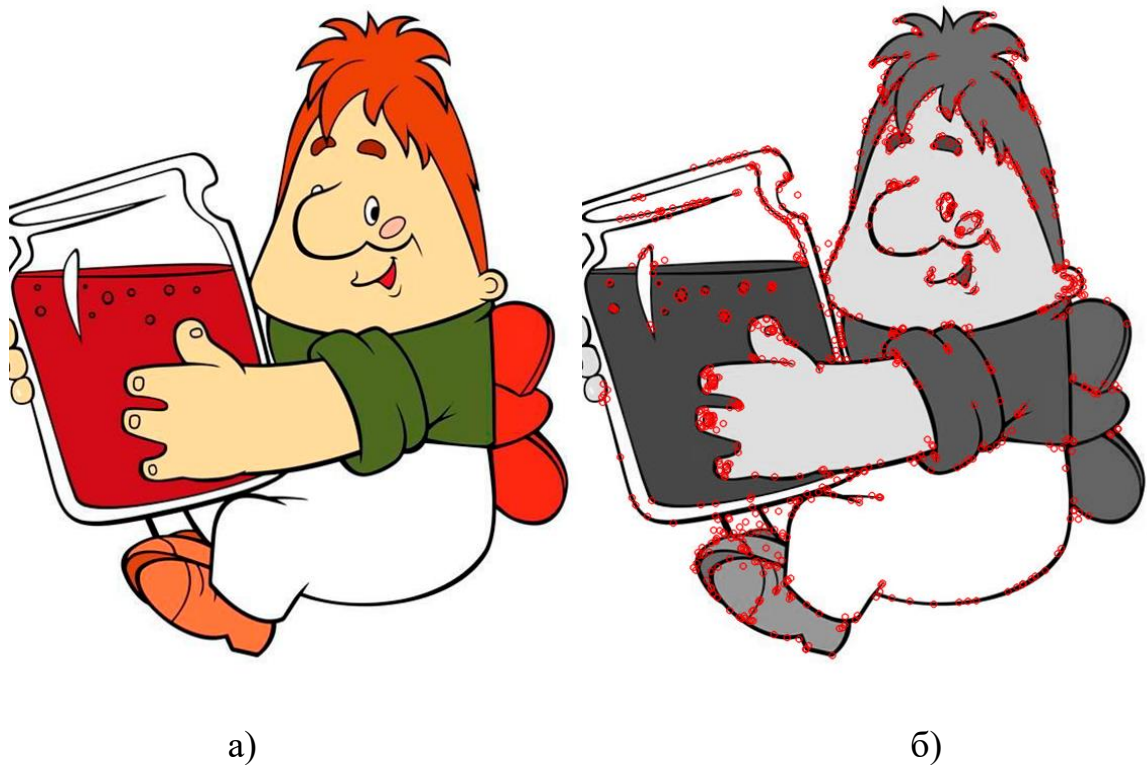


Рисунок 3.4 – Зображення та виділені ключові точки:

а) вхідне зображення; б) координати виділених ключових точок

Клас `BriskDetector` містить метод, який завантажує зображення з файлу, застосовує до нього метод `BRISK` для пошуку ключових точок та обчислення їх дескрипторів. Потім з усіх знайдених ключових точок вибираються перші 500, і вони записуються в об'єкт класу `VectorOfKeyPoint`, а також їх дескриптори зберігаються в матрицю `Descriptors`. Можемо побачити результат роботи методу `BRISK` у вигляді зображення з позначеними ключовими точками.

Для обробки зображень та знаходження хеммінгової відстані між дескрипторами було реалізоване наступне: клас `Hamming` містить методи для знаходження хеммінгової відстані між двома матрицями дескрипторів, а також для знаходження еталону за допомогою методу голосування на основі хеммінгової відстані. В методі `Etalon` до кожного дескриптору з матриці

дескрипторів застосовується знаходження хеммингової відстані з еталонами, які передаються у вигляді списку `Mat`. `Emgu CV` – це .NET обгортка для `OpenCV`, відкритої бібліотеки комп'ютерного зору. `Mat` є основним об'єктом для зберігання даних в `Emgu CV` і використовується для представлення зображень, матриць та інших структур даних. `Mat` є скороченням від `Matrix`, і це є основним типом даних для зберігання і маніпулювання даними зображення. `Mat` має ряд важливих властивостей та методів, зокрема:

- `Split`: Цей метод ділить канали зображення на декілька матриць;
- `Merge`: Цей метод об'єднує канали зображення в одну матрицю;
- `Save`: Цей метод зберігає матрицю в файл;
- `Dispose`: Цей метод видаляє матрицю та звільняє пам'ять, яку вона використовує;
- `GetRawData` – дозволяє отримати рядки з матриці за індексами які надані як параметри функції.

Це лише деякі з методів та властивостей, доступних в класі `Mat`. Якщо хеммингова відстань між дескриптором та еталоном менша за заданий поріг 128 який був знайдений як оптимальний для заданої вибірки (3.1), то даний еталон отримує голос. Кінцевий результат голосування повертається у вигляді об'єкту, що містить список результатів голосування для кожного еталону

$$m = \frac{1}{4}nl, \quad (3.1)$$

де  $m$  – величина порогу для Хемінгової відстані;

$n$  – кількість дескрипторів знайдених детектором;

$l$  – кількість бітів в кожному з дескрипторів.

Приклад результатів голосування для еталонів на прикладі зображення з 1 еталону:

- проходження порогу 1 з 1 зображень: 500;

- проходження порогу 1 з 2 зображень: 0;
- проходження порогу 1 з 3 зображень: 0.

### 3.3 Аналіз результатів експериментів

Після отримання дескрипторів для всіх зображень, проводилась процедура нормалізації та вагомості дескрипторів. Для кожного дескриптора був обчислений коефіцієнт вагомості, який відображає його значимість для класифікації. Коефіцієнти вагомості були визначені на основі аналізу внеску кожного дескриптора в розрізнення між класами зображень.

Після обчислення коефіцієнтів вагомості виконувалась побудова класифікаційної моделі на основі алгоритмів машинного навчання. Для цього був використаний навчальний набір даних, що складався з зображень, що були раніше позначені відповідним класом. В якості алгоритмів були обрані BRISK. Проведено процедуру налаштування параметрів з метою досягнення найкращих результатів.

Після побудови моделі проводилась її оцінка на тестовому наборі даних, що не використовувався в процесі навчання моделі. Результати оцінки моделі включали точність класифікації та матрицю помилок.

Для подальшого поліпшення результатів класифікації, було використано метод вибору найкращих дескрипторів з використанням коефіцієнтів вагомості як на (рис. 3.5-3.7). Таким чином, кількість використовуваних дескрипторів була зменшена до значення 100, яке було визначене як найбільш оптимальне для цього алгоритму.

Після проведення всіх етапів моделювання отримано класифікаційну модель, що показала високу точність класифікації зображень. Результати вказують на ефективність даного методу.

```

V1 results:
19 -16 15 50 -14 -11 38 80 58 42 7 -42 23 91 75 37 35 43 16 28 60 6 3 61 -2 27 32 26 40 5 -17 21 -10 7 33 11 -19 43 111
17 20 23 26 16 9 -4 16 -14 49 25 15 13 10 22 -4 -35 -18 -21 20 -2 -19 29 1 59 21 11 41 27 11 -7 -24 24 -1 37 32 4 63 80
12 17 -1 -67 24 4 78 41 91 34 11 28 94 110 -5 28 -7 9 -2 70 87 -10 7 -19 -10 45 5 4 1 11 34 19 8 -7 72 7 79 15 9 -15 43
114 26 19 0 25 55 -25 26 1 62 63 2 -6 -10 33 52 17 -17 29 28 46 -1 36 25 10 0 16 43 17 63 3 52 -16 -12 9 149 8 1 -18 20
-2 13 59 -2 20 -1 6 57 6 -3 93 8 21 6 38 17 1 19 49 -34 -4 52 29 25 20 -12 21 13 21 3 12 19 -35 44 10 3 33 42 -8 0 -11 2
0 46 -9 26 -3 31 23 5 36 38 43 -2 -7 32 -3 17 -4 -26 6 3 1 10 47 29 22 53 81 -14 11 17 27 50 -4 -12 -24 56 13 -4 82 -42
-9 -8 -22 5 -17 -2 4 -15 14 16 2 3 16 13 14 6 9 -11 15 -23 -17 30 26 20 5 37 14 19 -18 24 -4 -6 4 5 34 -8 -3 11 7 56 9 5
5 27 12 49 27 -12 20 -6 8 -6 -12 -19 4 39 26 -61 -7 50 25 7 0 36 30 -1 -10 -14 -5 2 -22 0 15 26 55 9 -7 49 47 11 13 25 6
1 48 52 52 -20 -49 4 2 5 -8 62 1 30 -6 13 -3 30 10 43 66 -12 -7 137 -32 -10 133 -20 -32 -4 19 -5 7 3 72 71 -3 13 24 39 1
3 2 4 0 -9 56 54 17 6 23 1 6 11 24 19 -2 -16 17 -11 9 43 1 4 32 33 56 86 50 18 22 32 52 1 7 34 -15 6 34 109 -10 17 14 14
-7 5 -30 -6 -1 29 46 12 39 25 5 7 -12 35 17 -8 28 35 -16 48 6 0 38 68 82 1 18 -12 -5 31 -10 63 118 -15 -15 12 2 -6 -1 -
22 84 -16 -9 -11 -5 28 -14 65 55 29 56 14 47 12 -3 67 5 -16 19 146 6 27 -20 -6 55 49 80 -1 38 -15 38 -5 51 54 69 52 -1 4
7 49 8 21 -14 25 42 -8 46 -4 -29 28 41 45 39 -12 13 27 53 -3

```

Рисунок 3.5 – Результати пошуку показника вагомості для першого еталону

```

V2 results:
-15 33 -30 -26 15 -12 34 9 3 11 -11 29 -2 -11 43 -3 0 11 6 21 12 13 -13 -30 33 21 18 -28 -24 -17 -21 6 6 80 5 -9 20 -24
-1 29 1 8 1 -19 -7 -10 13 14 12 -32 -8 -14 1 0 8 21 8 13 -7 -3 12 -9 7 -51 -21 25 -7 -16 -12 6 34 -3 -5 2 17 -5 2 7 -7 2
-2 -13 -2 -1 3 35 3 -9 -4 21 22 -31 -3 -20 22 18 31 4 -3 50 -2 20 15 57 74 5 23 77 77 8 -21 69 42 -31 -10 25 5 10 50 35
31 -9 34 56 40 -17 21 -14 -36 20 61 -7 39 2 -2 -3 -2 -38 -30 16 -35 -13 9 4 -23 -2 11 -14 -9 -15 48 -7 64 -15 19 25 -41
-20 -40 10 -8 -12 30 1 -3 -27 -33 -9 -5 29 -1 -5 -6 3 48 -11 -29 48 -6 -12 -8 17 44 53 29 -15 -9 -22 20 24 21 5 -3 -13
17 2 2 2 -23 -6 32 -47 -10 79 9 7 47 43 1 8 67 6 38 -1 -2 -39 24 -2 -8 -12 -4 5 -5 -1 -21 7 35 6 27 9 24 36 -32 -10 6 -1
8 23 17 -3 -17 39 -7 12 4 13 87 -2 87 36 18 8 23 9 10 46 26 45 -6 14 39 35 68 58 -4 -19 -6 -34 -20 14 -7 -26 18 11 -5 -4
1 14 -22 -1 -16 -22 -4 -1 24 57 59 72 71 -6 95 6 62 108 57 -4 38 105 36 97 107 26 39 3 -2 -19 -10 36 12 -37 -24 -14 3 -
4 11 -5 10 102 -6 19 107 8 4 27 20 2 68 55 76 44 25 81 68 -16 -19 75 67 38 89 99 93 98 84 1 44 -24 19 -9 7 -2 -10 -38 10
-9 1 -14 -14 -9 -15 -22 -28 6 22 -3 -23 5 50 33 42 77 15 0 115 62 116 25 -14 -3 58 -22 -22 3 -2 -8 -36 -15 0 -19 -8 -15
-13 -14 -5 22 57 -35 32 -9 -39 14 7 23 -8 -23 -3 2 4 4 41 0 9 -19 -14 18 31 -32 -8 21 -13 18 -12 28 -20 -48 -10 -5 -11
7 29 -4 -47 16 -5 -17 23 -16 9 -8 -10 -2 -16 3 -16 -29 -4 13 11 -9 -11 -8 -35 0 3 -16 9 -14 -15 44 -7 15 4 -4 -36 81 14
-6 3 12 -9 11 2 -30 11 -12 -4 -3 -5 1 -10 19 -3 -10 30 0 12 -13 11 -13 -15 4 -4 -17 9 2 0 5 -17 -18 -23 -41 7

```

Рисунок 3.6 – Результати пошуку показника вагомості для другого еталону

```

V3 results:
17 30 42 47 21 22 -16 -30 14 -10 11 -7 0 20 33 6 59 8 43 35 -15 -23 31 -9 -8 0 22 24 10 28 24 17 -20 54 52 -5 3 -3 -11 -
43 -13 -5 12 93 -18 23 100 -8 -30 13 38 -11 18 -10 -3 8 1 24 -18 16 28 15 -46 -47 -14 29 -1 -15 0 10 -6 29 48 -8 -57 -5
-12 37 16 -10 33 30 24 40 18 54 13 10 -28 10 15 -1 -10 16 10 6 -2 -44 1 -6 12 30 9 14 -7 41 6 10 10 35 38 15 -2 -24 51 1
7 25 47 9 15 47 22 25 15 2 29 7 73 39 21 -4 4 71 34 -12 -3 12 27 18 37 30 17 30 18 48 30 -52 -2 36 45 0 22 -6 -3 45 33 1
6 24 40 42 20 24 42 -21 33 23 28 32 46 0 36 25 31 29 19 50 22 34 7 24 82 74 19 -15 34 -14 -1 -18 -7 -11 2 30 59 6 17 46
44 3 8 -4 -2 -8 17 11 -5 12 40 6 52 -5 8 1 5 36 52 -25 -14 29 15 13 9 30 16 6 35 68 30 0 -4 69 -27 67 31 11 -4 -2 25 -6
17 16 -2 -7 16 31 13 -23 6 26 9 7 4 20 12 20 34 22 -35 44 4 -16 -20 34 4 38 32 2 25 -7 7 34 -26 -5 87 92 68 8 -5 43 -13
7 65 -10 -8 20 37 10 15 2 51 6 24 19 -1 -25 61 -19 68 65 15 40 39 52 17 74 10 37 13 69 96 0 2 101 25 54 43 31 24 28 0 61
-2 54 39 41 24 44 39 0 69 44 19 12 -11 -6 -13 42 36 -8 105 89 91 49 45 -2 78 1 28 12 52 2 66 7 33 4 -42 71 12 21 23 30
-7 29 1 -8 14 70 87 112 37 52 108 3 74 34 117 4 23 21 -5 119 27 39 25 100 47 45 53 21 -12 0 69 0 95 39 16 37 1 30 -11 22
31 95 15 71 16 11 10 78 -12 6 13 10 -4 5 22 -25 0 36 -22 -24 -2 81 -5 -2 -8 78 56 -2 93 19 42 50 41 46 10 29 9 61 18 11
14 28 8 -42 26 30 23 34 9 -22 4 31 11 -4 19 -4 29 14 -18 35 6 55 47 -9 50 55 41 47 17 29 4 33 38 22 67 52 59 34 91 49 7
9 45 49 21 57 -10 37 56 30 19 -15 8 27 30 17 38 22 23 49 27

```

Рисунок 3.7 – Результати пошуку показника вагомості для третього еталону

Для кожного зображення в наборі даних проводиться екстракція дескрипторів. Можуть бути використані різні методи, але важливо, що результатом є бінарний вектор, що представляє зображення. Хемінгова відстань між двома бінарними векторами – це кількість позицій, в яких відповідні біти векторів відрізняються. Отже, для кожної пари зображень обчислюється хемінгова відстань між їхніми дескрипторами. Для кожного зображення, що віднесено до класифікації, проводиться голосування на основі хемінгових відстаней до всіх інших зображень. Зображення вважається віднесеним до того класу, який отримав найбільшу кількість «голосів», тобто до якого було найменше хемінгових відстаней.

Результати проведеного класичного методу голосування представлено на (табл. 3.1). Як видно з результатів, класифікація працює правильно та відносить відповідні зображення до тих же еталонів за порогом проходження в 128 одиниць відстані. Також результати показують, що для цих еталонів зображення є досить подібними, так як велику кількість голосів відносить до інших еталонів.

Таблиця 3.1 – Результати класичного методу голосування

Класичний метод голосування			
Еталони	1	2	3
1	500	198	165
2	179	500	209
3	175	293	500
Витрачено часу			27537

Для результатів з (табл. 3.2) можна зробити висновок, що методи відбору за критерієм важливості теж є досить достовірним і має подібне до класичного методу співвідношення кількості голосів, що пройшли поріг в 128 одиниць для хемінгової відстані.

Таблиця 3.2 – Результати голосування дескрипторів за показником вагомості

Голосування за відбором за показником вагомості			
Еталони	1	2	3
1	100	19	14
2	28	100	29
3	13	31	100
Витрачено часу			8235

Якщо порівнювати ці два методи, то стає очевидним, що другий метод дає можливість провести класифікацію зображень з подібною точністю, але при цьому швидкість виконання цих методів суттєво відрізняється, так як для 100 значущих дескрипторів вона краща мінімум у 3 рази. Тож є доцільним використання подібного методу в задачах класифікації.

## ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований метод класифікації зображень з відбором дескрипторів з найбільшими показниками вагомості.

Проведені результати показали, що запропонований метод дає вигоду у часі оброблення в порівнянні з традиційним методом приблизно у три рази. Класифікація на підставі підмножини дескрипторів з найбільшим значенням вагомості дає можливість зменшити часові затрати на розрахунок та звільнити більше ресурсів пам'яті при обрахунках.

Практичні рекомендації із проведеної роботи полягають у зменшенні об'єму необхідних ресурсів для проведення класифікації за рахунок попереднього обрахування показників вагомості для еталонів.

Новизна роботи полягає у використанні методу класифікації, в якому дескриптори відбираються на показника вагомості.

Практична значущість роботи полягає у розробленні програмного застосунку, який можна використати у задачах комп'ютерного зору.

Перспективи роботи полягають у розширенні варіантів для способів визначення значущості і методів класифікації із використанням вагових коефіцієнтів.

Результати роботи апробовано у вигляді тез доповідей під час XXVII Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ» [31].

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Гороховатський, В. А., & Полякова, Т. В. (2018). Застосування просторових структур ознак для класифікації зображень у комп'ютерному зорі. С. 120.
2. Гонсалес, Р., & Вудс, Р. (2012). Цифрове оброблення зображень, 2005. С. 1072.
3. Гороховатський, В. А. (2014). Структурний аналіз та інтелектуальна обробка даних у комп'ютерному зорі. С. 316.
4. Берестовський, О. Є., Власенко, О. Н., & Гороховатський, В. А. (2015). Нейросетельні технології самонавчання у системах структурного розпізнавання візуальних об'єктів. Реєстрація, зберігання та обробка даних. С. 19–27.
5. Гороховатський, В. А., Куликов, Ю. А., & Берестовський, О. Є. (2015). Побудова та застосування функції приналежності для результативного структурного аналізу зображень. С. 78–85.
6. Верес, О. М., Кісь, Я. П., Кугівчак, В. А., & Рішняк, І. В. (2018). Вибір методів для пошуку однакових або схожих зображень. *Вісник Національного університету "Львівська політехніка". Серія: Інформаційні системи та мережі*, (887), 43-50.
7. Tuytelaars, T., & Mikolajczyk, K. (2008). Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision*, 3(3), 177-280.
8. Шапиро, Л., & Стокман, Д. (2013). Комп'ютерний зір. С. 752
9. Гороховатський, В. А., & Путятін, Е. П. (2008). Структурне розпізнавання зображень з урахуванням моделей голосування ознак характерних точок. Реєстрація, зберігання та обробка даних. С. 75–85.
10. Lowe, D. G. (1999, September). Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision* (Vol. 2, pp. 1150-1157). Ieee.

11. Calonder, M., Lepetit, V., Strecha, C., & Brief, F. P. Binary robust independent elementary features. In *Proceedings of the European Conference on Computer Vision* (pp. 778-792).

12. Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011, November). BRISK: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision* (pp. 2548-2555). Ieee.

13. Гороховатський В., Творошенко І., Сидоренко Д. (2021) Класифікація зображень із використанням кластерного подання, Міжн. науковий симпозіум «Інтелектуальні рішення-С». Обчислювальний інтелект (результати, проблеми, перспективи). Теорія прийняття рішень: праці міжн. наук. симпозіуму (Вересень 29, 2021). Київ – Ужгород, С. 44-45.

14. Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision* (pp. 2564-2571). Ieee.

15. Ebrahimi, M., & Mayol-Cuevas, W. W. (2009, June). SUSurE: Speeded up surround extrema feature detector and descriptor for realtime applications. In *2009 IEEE computer society conference on computer vision and pattern recognition workshops* (pp. 9-14). IEEE.

16. 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. С. 5–13.

17. 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.

18. Han, J., Pei, J., & Tong, H. (2022). *Data mining: concepts and techniques*. Morgan kaufmann. P. 754

19. Yakovleva O., and Nikolaieva K. (2020) Research of descriptor based image normalization and comparative analysis of SURF, SIFT, BRISK, ORB, KAZE, AKAZE descriptors. *Advanced Information Systems*, 4 (4), pp. 89-101.

20. 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.

21. Гороховатський, В. О., Гадецька, С. В., & Пономаренко, Р. П. (2019). Статистичні розподіли та ланцюжкове подання даних при визначенні релевантності структурних описів візуальних об'єктів. С. 87–92.

22. 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), 787-797.

23. Gorokhovatskyi V.A., Zamula A.A. (2016) Employment of Intelligent Technologies in Multiparametric Control Systems. *Telecommunications and Radio Engineering*. Vol. 75, No 19, p. 1775–1785.

24. Duda, R. O., Hart, P. E., & Stork, D. G. (1973). *Pattern classification and scene analysis* (Vol. 3, pp. 731-739). New York: Wiley.

25. Gadetska, S. V., & Gorokhovatskyi, V. O. (2018). Statistical measures for computation of the image relevance of visual objects in the structural image classification methods. *Telecommunications and Radio Engineering*, 77(12), pp. 1041– 1053.

26. Fu, K. S. (1976). Pattern recognition and image processing. *IEEE transactions on computers*, 100(12), 1336-1346.

27. 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.

28. 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.

29. 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.

30. Kaehler, A., & Bradski, G. (2016). *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. " O'Reilly Media, Inc."

31. Гончаров Д. 2023 Редукція опису зображення за критерієм інформативності. 27-ий міжнародний молодіжний форум «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ», С. 94-95.

32. 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.

33. Gorokhovatskyi, V., Gadetska, S., & Stiahlyk, N. (2020). Image structural classification technologies based on statistical analysis of descriptions in the form of bit descriptor set.

34. Gorokhovatskyi, V., Tvoroshenko, I., & Chmutov, Y. (2022). Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень. *Advanced Information Systems*, 6(3), 5-12.

35. Gorokhovatskyi, V., & Vlasenko, N. (2021). Редукція опису зображення у складі множини дескрипторів на основі метричного критерію інформативності. *Advanced Information Systems*, 5(4), 10-16.

36. Flach, P. (2012). *Machine learning: the art and science of algorithms that make sense of data*. Cambridge university press.

37. Гороховатський, В. О., & Творошенко, І. С. (2022). Аналіз багатовимірних даних за описом у формі множини компонент.

38. Гороховатський, В. О., & Гадецька, С. В. (2020). Статистичне оброблення та аналіз даних у структурних методах класифікації зображень. С. 128.

39. Szeliski, R. (2022). *Computer vision: algorithms and applications*. Springer Nature.
40. Gadetska, S., Gorokhovatskyi, V., Stiahlyk, N., & Vlasenko, N. (2022). *Aggregate Parametric Representation of Image Structural Description in Statistical Classification Methods*.
41. Gorokhovatsky, V., Stiahlyk, N., & Tsarevska, V. (2021). *Combination method of accelerated metric data search in image classification problems*. *Advanced Information Systems*, 5 (3), 5-12.
42. Gorokhovatskyi, O., Gorokhovatskyi, V., & Peredrii, O. (2018). *Analysis of application of cluster descriptions in space of characteristic image features*. *Data*, 3(4), 52.
43. Gorokhovatskyi, O., Peredrii, O., Gorokhovatskyi, V., & Vlasenko, N. (2023). *Explanation of CNN image classifiers with hiding parts*. In *Explainable Deep Learning AI* (pp. 125-146). Academic Press.
44. Gorokhovatskiy, V. A. (2011). *Compression of descriptions in the structural image recognition*. *Telecommunications and Radio Engineering*, 70(15).
45. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Zeghid, M. (2022). *Tools for Fast Metric Data Search in Structural Methods for Image Classification*. *IEEE Access*, 10, 124738-124746.
46. Gorokhovatskiy, V. A. (2016). *Efficient Estimation of Visual Object Relevance during Recognition through their Vector Descriptions*. *Telecommunications and Radio Engineering*, 75(14).
47. Gorokhovatsky, V. O., Pupchenko, D. V., & Solodchenko, K. G. (2018). *Аналіз властивостей, характеристик та результатів застосування новітніх детекторів для визначення особливих точок зображення. Системи управління, навігації та зв'язку. Збірник наукових праць*, 1(47), 93-98.