

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-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
« _____ » _____ 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Жадану Олексію Віталійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення методу класифікації зображень за інтегральним поданням розподілів компонентів даних.

затверджена наказом університету від « 20 » травня 2021 року № 663Ст

2. Термін подання студентом роботи до екзаменаційної комісії 25 травня 2021 р.

3. Вихідні дані до роботи Ознаки візуальних об'єктів, ключові точки, детектори ключових точок, бінарний дескриптор ключових точок ORB, поняття та визначення медіани множини дескрипторів, відстань Хемінга, статистичний розподіл даних, класифікація зображень.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Огляд множини бінарних дескрипторів як простір ознак візуальних об'єктів.

2. Розробка методу класифікації на основі статистичних розподілів компонентів даних.

3. Програмна реалізація досліджуваної процедури побудови класифікатора.

4. Огляд отриманих результатів та формулювання висновків.

5. Впровадження методів удосконалення класифікатора.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Постановка задачі, база еталонних зображень, виділені дескриптори ORB, зображення під дією шумів, таблиці аналізованих даних, результати моделювання.

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

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	12.04.21	виконано
2	Аналіз завдання, підбір літератури	12.04.21-13.04.21	виконано
3	Аналіз літератури з досліджуваної проблеми	13.04.21-16.04.21	виконано
4	Аналіз технічних засобів	16.04.21-17.04.21	виконано
5	Розробка методу	17.04.21-28.04.21	виконано
6	Програмна реалізація	28.04.21-08.05.21	виконано
7	Оформлення пояснювальної записки	08.05.21-20.05.21	виконано
8	Перевірка на плагіат	26.05.21	
9	Рецензування	27.05.21	
10	Підготовка презентації та доповіді	28.05.21-30.05.21	
11	Занесення роботи в електронний архів	01.06.21	
12	Попередній захист кваліфікаційної роботи	02.06.21	

Дата видачі завдання 12 квітня 2021 р.

Студент _____
(підпис)

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

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 58 с., 11 табл., 23 рис., 30 джерел.

КОМП'ЮТЕРНИЙ ЗІР, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, СТРУКТУРНИЙ ПІДХІД, КЛЮЧОВА ТОЧКА, ДЕТЕКТОР ORB, СТАТИТИЧНИЙ РОЗПОДІЛ ДАНИХ.

Об'єктом роботи є метод класифікації зображень на основі статистичних розподілів для множини дескрипторів ключових точок візуальних об'єктів.

Метою роботи є моделювання та дослідження працездатності методу класифікації зображень, що базується на побудові ансамблю розподілів для компонентів структурного опису з використанням їх інтегрального подання для кожного класу.

Використаними методами є інтелектуальний аналіз даних, математична статистика, детектор ORB для формування дескрипторів ключових точок, моделі для визначення релевантності для множин векторів даних, програмне моделювання.

У результаті роботи програмно реалізована досліджувана модель класифікації та перевірена її працездатність.

Дослідження викладені у одній науковій статті, що опублікована у посібнику «Сучасні інформаційні системи» Том 5, № 1 (2021) та у тезах в рамках 25-го Міжнародного молодіжного форуму.

COMPUTER VISION, IMAGE CLASSIFICATION, STRUCTURAL APPROACH, KEY POINT, ORB DETECTOR, STATITICAL DISTRIBUTION OF DATA.

The object of the work is a method of classifying images based on statistical distributions for a set of descriptors of key points of visual objects.

The aim of this work is to model and study the performance of the method of image classification based on the construction of an ensemble of distributions for components of structural description using their integral representation for each class.

The methods used are intelligent data analysis, mathematical statistics, ORB detector for forming key point descriptors, models for determining relevance for sets of data vectors, software modeling.

The research is presented in one scientific article published in the manual "Advanced Information Systems" Volume 5, No. 1 (2021) and in theses within the 25th International Youth Forum.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ.....	7
1 Основні поняття технології структурного розпізнавання зображень	9
1.1 Різновиди ознак візуальних об'єктів	9
1.2 Подання ключових точок при аналізі зображень	12
1.3 Опис дескриптору ORB.....	16
1.4 Постановка задачі	24
2 Модель класифікації на підґрунті структурних описів.....	26
2.1 Формалізація задачі класифікації.....	26
2.2 Побудова ансамблю розподілів для компонентів.....	28
2.3 Знаходження центру класу.....	31
2.4 Модель класифікатора.....	33
3 Програмна реалізація та експериментальний аналіз результативності класифікації.....	36
3.1 Опис використаних програмних інструментів	36
3.2 Експериментальний аналіз результативності методу класифікації.....	38
3.3 Перевірка стійкості розробленого методу класифікації	43
3.4 Аналіз способів виділення інформативної частини опису	51
Висновки	54
Перелік джерел посилання	56

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

КТ – ключова точка / ключові точки

ORB – Oriented FAST and rotated BRIEF

FAST – Features from Accelerated Segment Test

BRIEF – Binary Robust Independent Elementary Features

SIFT – Scale-invariant feature transform

SURF – Speeded-Up Robust Features

BRISK – Binary Robust Invariant Scalable Keypoints

FREAK – Fast Retina Keypoint

LINQ – Language Integrated Query

Emgu CV – .NET обгортка для графічної бібліотеки OpenCV

OpenCV – Open Source Computer Vision Library

ВСТУП

Сьогодні, з розвитком обчислювальних систем та інформаційних технологій, зростає необхідність в автоматизованих системах у різноманітних промислових, наукових сферах та у повсякденному житті людей. Ефективне використання даних дозволяє значно підвищити продуктивність та розширити функціонал інформаційних систем. Як наслідок, зростає потреба у працездатних методах обробки вхідних даних, зокрема відео або зображень. Підтвердженням цих слів, є розповсюдження різноманітних програмних інструментів у новітніх смартфонах, які дозволяють за допомогою камери виміряти розмір об'єкта, або ж розблокувати систему використовуючи відбитки пальців та сканування обличчя. Та це лиш мала частина функціоналу, яка в основі використовує алгоритми комп'ютерного зору [1-8].

Не дивлячись на велику популярність, технології комп'ютерного зору до цього часу не дослідженні повною мірою і потребують дієвих та дешевих за ресурсами методів аналізу та розпізнавання зображень. Людина звичайно може легко розрізняти образи, звуки, аромати, проте для комп'ютера важко розв'язувати такі проблеми сприйняття. Причиною цього є те, що кожен шаблон зазвичай містить велику кількість інформації, і її достатньо непросто представити повною мірою у вигляді 0 та 1. Тож знаходження дієвого методу розпізнавання ілюстрованих об'єктів є досить актуальною темою на сьогодні [9-12].

Будь-яка задача розпізнавання образів поділяється на три етапи: попередня обробка, виділення ознак та класифікація. Кожен з них має свої особливості та проблеми, необхідно враховувати велику кількість факторів. Не зважаючи на велике бажання, досягти повної працездатності, коректності та дешевизни обчислюваних ресурсів розробленого методу розпізнавання задача не з легких, але спробувати варто.

Завданням даного дипломного проєкту є дослідження моделі класифікації, тобто віднесення деякого зображення до певної категорії (класу). Множину класів задає сукупність еталонних зображень. Усі зображення, що беруть участі у класифікації будуть представлені у вигляді множини векторів, що описують ознаки ілюстрованих об'єктів. Ознаками є бінарні дескриптори ключових точок з однаковою розмірністю. Це дасть перевагу під час обчислення шляхом використання двійкових операцій, що своєю чергою спрощує реалізацію заданого методу. Сама модель побудови класифікатора полягає у використанні ансамблю розподілів для компонентів структурного опису через їх агрегацію для кожного із класів, що своєю чергою використовується як апарат прийняття кваліфікаційних рішень.

1 ОСНОВНІ ПОНЯТТЯ ТЕХНОЛОГІЇ СТРУКТУРНОГО РОПІЗНАВАННЯ ЗОБРАЖЕНЬ

1.1 Різновиди ознак візуальних об'єктів

Всупереч тому, що розвиток інформаційних технологій з кожним роком набуває дедалі більших масштабів, комп'ютери ще не отримали можливості приймати рішення на основі певних почуттів та інтуїтивних прогнозів. Якщо людина може легко визначити на яких фотокартках зображені котики, а на яких собачки, то для обчислювальної машини це всього лише купа 0 і 1, яка розставляє пікселі певних кольорів у необхідній послідовності. Комп'ютерні алгоритми вимагають більш вагомого набору вхідних даних для подальшої обробки.

Сучасні методи комп'ютерного зору полягають у знаходженні особливостей, змістовної інформації про візуальні об'єкти, яка потрібна для певного обчислення та обробки з метою зіставлення зображень для знаходження необхідних відповідностей, ступеню схожості, тощо. Існує безліч алгоритмів, які використовують виявлення особливостей в якості початкового кроку. Це стало причиною розробки великої кількості методів детектування. Усі вони відрізняються за типами виявлених ознак, рівнем обчислювальної складності та самою сферою використання.

Зазвичай виділяють чотири типи ознак візуальних об'єктів: *край, ребро, кут, та крапля* [1-4].

Виявлення країв містить у собі безліч математичних методів, спрямованих на визначення точок яскравість яких різко змінюється, тобто, має неоднорідності. Точки, яким притаманна дана характеристика, зазвичай об'єднуються в набір вигнутих ліній, званих краями [1]. На рисунку 1.1 проілюстрований результат роботи алгоритму детектування країв з використанням *оператору Собеля (Sobel Edge Detector)*.

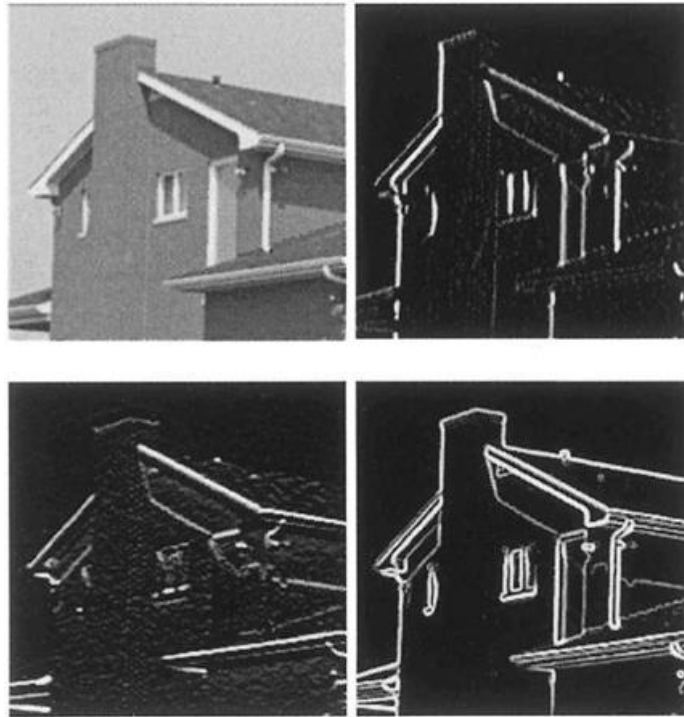


Рисунок 1.1 – Результат роботи детектору країв

У найкращому випадку результат застосування крайового детектора до зображення повинен призвести до набору з'єднаних кривих, які вказують межі об'єктів. Однак отримати такі ідеальні краї не завжди вдається. Причиною є багато факторів: складність ілюстрованих об'єктів, якість самого зображення, рівень яскравості та інші.

Дещо візуально схожим на попередній є тип ознак під назвою ребра – це тонкі лінії, які значно темніші або світліші за їх навколишню область [2]. На рисунку 1.2 представлено порівняння результатів роботи алгоритмів виявлення країв та ребер.

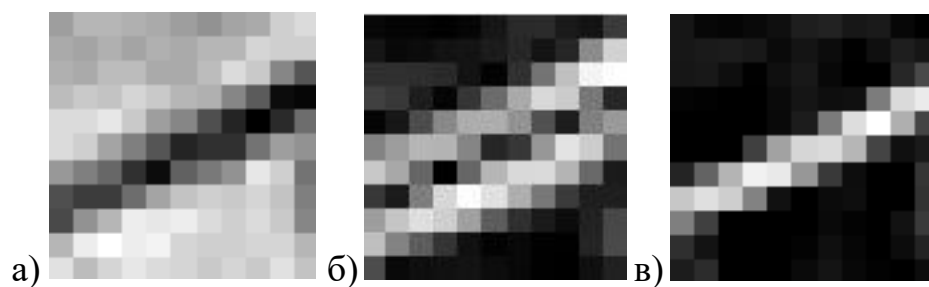


Рисунок 1.2 – Порівняння результатів роботи алгоритмів виявлення ознак:

а) – вхідне зображення; б) – виявлені краї; в) – виявлені ребра

Наступним типом ознак графічних об'єктів є кут. Його можна визначити як точку, для якої є два домінуючі та різні за напрямками ребра у локальній місцевості. У літературі можна часто побачити, що тема виявлення кутів часто накладається на тему виявлення точок інтересу, але ці поняття трішки відрізняються [3]. Більш детально про ключові точки у п.п. 1.2.

На рисунку 1.3 зображений результат роботи *кутового детектора Харріса (Harris Corner Detector)*.

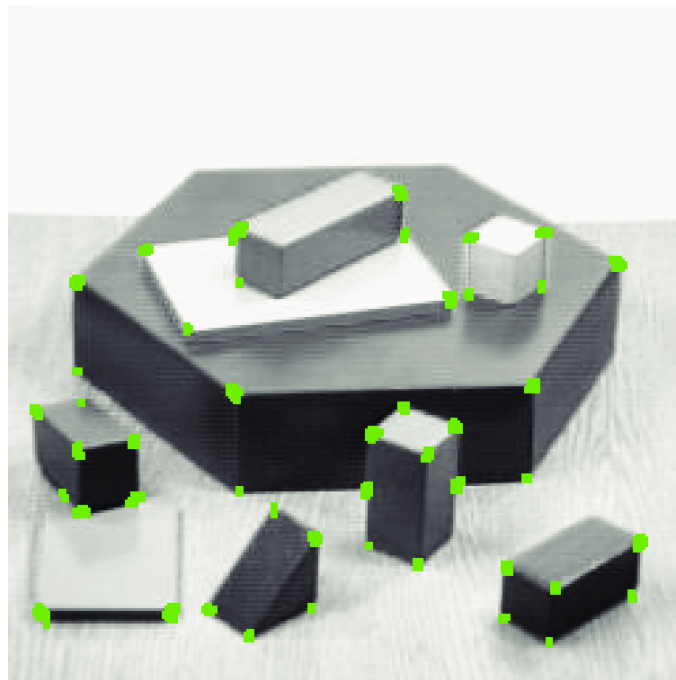


Рисунок 1.3 – Приклад роботи алгоритму виявлення кутів

Завершає перелік ознак крапля. Крапля – це область зображення, в якій деякі властивості є постійними або приблизно постійними, тобто, всі точки в краплі можна у певній мірі вважати подібними між собою [4]. Дані детектори було розроблені для отримання додаткової інформації про вагомні області зображення, яку не могли отримати алгоритми виявлення кутів та країв через свої розміри. Також, існує деякий зв'язок з детекторами ребер, що, як наслідок, допомагає знаходити продовжені об'єкти. Приклад роботи такого детектору проілюстровано на рисунку 1.4.

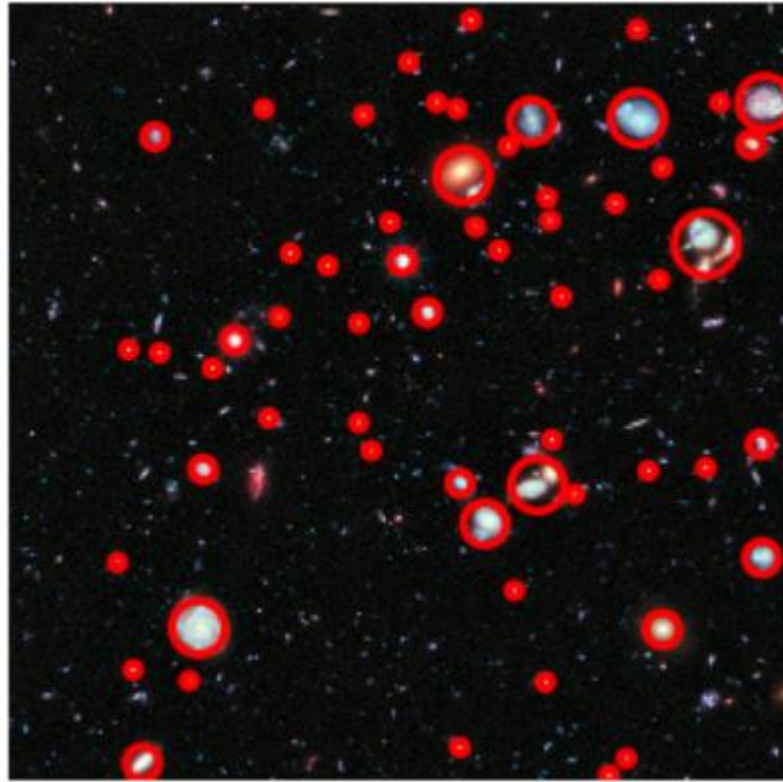


Рисунок 1.4 – Приклад роботи алгоритму розпізнавання краплі

У даному дипломному проєкті характеристиками візуальних об'єктів виступають кути/ключові точки. Саме вони дозволяють отримати зручну для обробки множину вхідних даних шляхом побудови дескрипторів КТ (детальніше у п.п. 1.2), які надалі вирішують завдання розпізнавання. У подальшому тексті, під час використання словосполучень *«ознаки / особливості / характеристики візуальних об'єктів»* мається на увазі що це особливі точки.

1.2 Подання ключових точок при аналізі зображень

Ключові (особливі) точки – це множина точок з деякими характеристиками, що відрізняють їх від інших точок на зображенні, містять вагому для аналізу інформацію та можуть бути стабільно виявлені за певним ступенем варіацій [5]. Як правило, це кутові, граничні точки, з різким

перепадом кольорів або яскравості, тощо. У літературі [6] дослідники виділили наступні характеристики особливих точок:

- *відмінність (distinctness)* – точка явно виділяється на фоні інших в межах своєї місцевості;

- *інваріантність (invariance)* – виявлення ключової точки не повинно залежати від будь-яких геометричних перетворень;

- *стабільність (stability)* – наявність шумів не повинно значною мірою впливати на кінцеву множину виділених КТ;

- *унікальність (uniqueness)* – характеристики ключової точки повинні робити її унікальною на усій аналізованій поверхні для підвищення різниці повторюваних патернів;

- *здатність до інтерпретації (interpretability)* – здатність використовувати КТ у зручному форматі даних для подальшого аналізу зображення.

У ряді досліджень сформовано дещо інші властивості [7]:

- *повторюваність (repeatability)* – особлива точка знаходиться в одному і тому ж місці незалежно від ракурсу та освітлення;

- *відмінність / інформативність (distinctiveness / informativeness)* – околиці ключових точок повинні мати великі відмінності для того, щоб мати можливість виділити та віднести одну до одної;

- *локальність (locality)* – оточення особливої точки повинна мати невеликий розмір для того, щоб зменшити чутливість до різних перетворень;

- *кількість (quantity)* – множина повинна мати у собі достатню кількість ключових точок, щоб їх вистачило для виділення навіть невеликих об'єктів;

- *точність (accuracy)* – виявлені особливі точки повинні відповідно розміщуватися як на оригінальному зображенні, так і зі змінним масштабом;

- *ефективність (efficiency)* – час виявлення повинен бути максимально оптимальним для застосування.

У цілому можна вважати, що це однакові за змістом, але по-різному інтерпретовані набори властивостей, що характеризують ключові точки.

Зазвичай, алгоритми обробки зображень із використанням КТ містять у собі два етапи. Перший полягає у використанні так званих детекторів, що автоматично виявляють вказане число КТ за деяким алгоритмом. Наступним кроком знайдена множина репрезентується за допомогою дескрипторів для представлення особливостей кожної із них.

Дескриптор – це числовий опис певної області, на яку посилається КТ, що дозволяє виділити її з усієї множини на відповідному зображенні. Самі по собі ключові точки ніякої інформації не несуть, окрім їх позиціонування, тобто координат у двовимірному просторі. Через це були створені дескриптори, які певним чином аналізують окіл відповідної точки, враховуючи яскравість, інтенсивність кольорів та інші характеристики, які б могли допомогти ідентифікувати КТ. Дескриптори подаються у вигляді векторів з однаковою розмірністю, у залежності від обраного алгоритму та його вхідних опцій.

Вміст дескриптору має бути незалежним від позицій КТ та залишатися стійким до графічних перетворень. Це дозволить порівнювати зображення на предмет змістовної схожості незалежно від ракурсу, палітри кольорів, тощо. Та, все рівно, домогтися повної незмінюваності неможливо.

Нижче наведені основні етапи обчислення, які використовують сучасні алгоритми для формування набору дескрипторів [8]:

– *виявлення множини точок інтересу* – здійснюється пошук по всій області зображення з використанням певних функцій, як, наприклад, DoG (difference-of-Gaussian), для знаходження потенційних ключових точок;

– *локалізація ключових точок* – дослідження локальної місцевості пікселя для підтвердження теорії про його особливість;

– *призначення орієнтації* – кожній КТ обчислюється параметр орієнтації на основі напрямків градієнта зображення;

– *побудова дескриптору* – інтерпретація локальної місцевості КТ у деякий набір інформації, що підлягає обробці за допомогою різноманітних математичних моделей.

Існує значна кількість алгоритмів для детектування КТ та дескрипторів. До перших можна віднести ORB, MSER, SIFT, SURF, FAST, кутовий детектор Харріса, SUSAN, FREAK та BRISK, до других – SURF, ORB, BRIEF, BRISK, SIFT, FREAK. Деякі назви повторюються, бо дані методи у своїй реалізації поєднують функцію виділення множини КТ та побудову дескрипторів. На рисунку 1.5 зображено графічне подання двійкових дескрипторів.

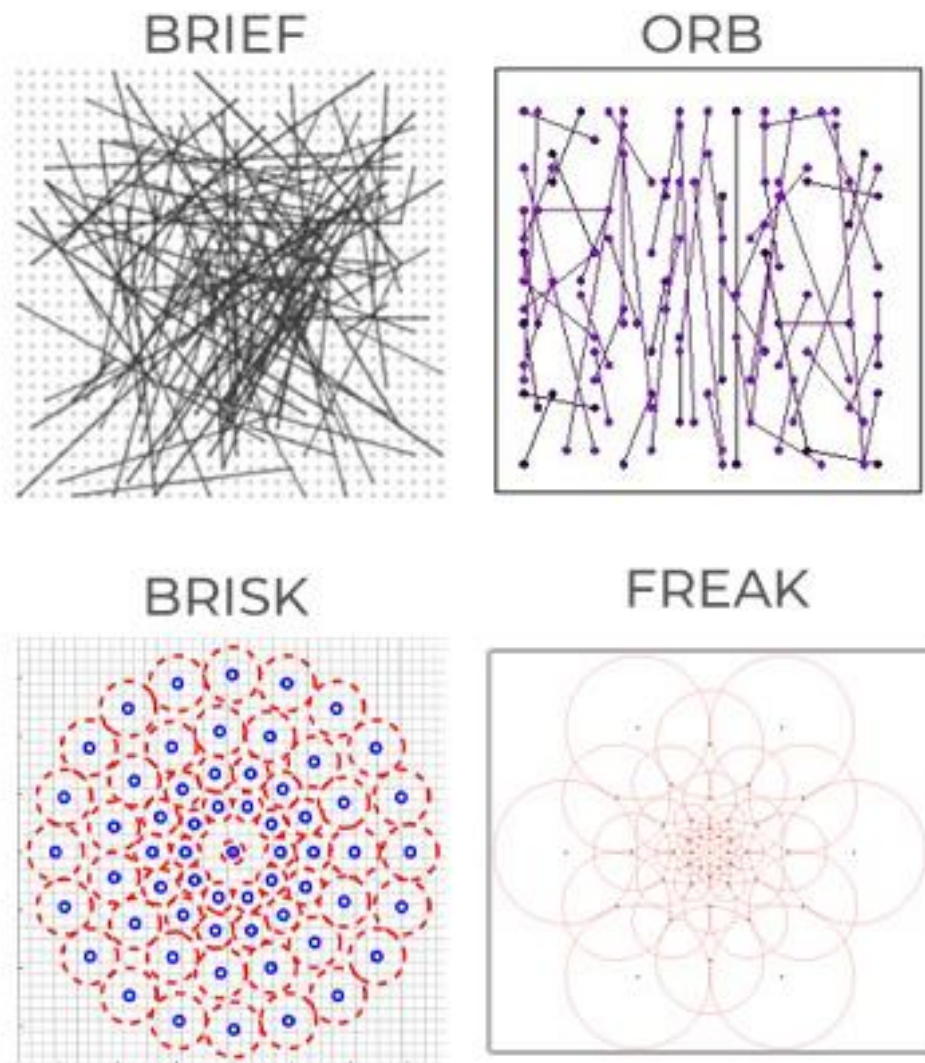


Рисунок 1.5 – Ілюстрація бінарних дескрипторів BRIEF, ORB, BRISK та FREAK

У роботі [9] дослідники порівняли деякі алгоритми на предмет стійкості у випадку, коли зображення підлягає різноманітним перетворенням та трансформаціям: ефект від зміни масштабу, повороту об'єктів, розмиття зображення, підвищення рівня освітленості, стиснення JPEG та порівняння фото чистої стіни та з нанесеним графіті. Також, окремо проведені експерименти по розрахунку часу виявлення на кожну ключову точку. Отримані дані по останньому подано у таблиці 1.1.

Таблиця 1.1 – Порівняння часу виявлення для кожної КТ

Алгоритм (детектор + дескриптор)	Кількість КТ	Час, мс	Час на одну КТ, мс
SIFT + SIFT	3665	5,989	0,0016341
SURF + SURF	3634	1,083	0,0002977
MSER + SIFT	323	0,889	0,0027533
BRISK + FREAK	466	2,531	0,0054322
BRISK + BRISK	466	0,235	0,0005046
ORB + ORB	500	0,236	0,0004715
FAST + BRIEF	11880	0,083	0,0000070

Після всіх проведених експериментів дослідники зробили висновки по кожному із них. У результаті, найкращими виявилися пари ORB + ORB та FAST + BRIEF (обидва являються складовими ORB, детальніше у п.п. 1.3), тож це доволі вагома причина обрати ORB за основу досліджуваного методу класифікації.

1.3 Опис дескриптору ORB

Oriented FAST and Rotated BRIEF (ORB) – відносно новий бінарний дескриптор, в основі якого лежить детектор ключових точок FAST та

дескриптор BRIEF. Являється інваріантним, стійким до шумів та дешевим за обчислювальними ресурсами [10].

Features from accelerated segment test (FAST) – це метод виявлення ключових точок, який може бути використаний для відстеження та картографування об'єктів у багатьох завданнях комп'ютерного зору. Особливістю цього детектора є його обчислювальна ефективність, що дає змогу використовувати його для обробки зображень та відео в режимі реального часу.

Послідовність алгоритму наступна [9]:

- обирається піксель p , що являється потенційною ключовою точкою. Його яскравість позначимо як I_p ;
- обирається поріг для допустимої яскравості t ;
- розглядаються 16 пікселів, що лежать на колі з центром у точці p , воно називається колом Брезенхема з радіусом 3 (рис. 1.6);

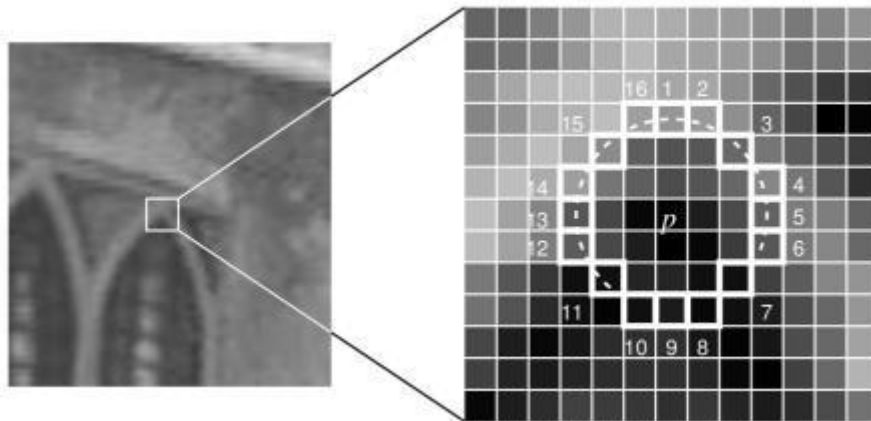


Рисунок 1.6 – Фрагмент зображення з потенційною КТ та з 16 точками, що розглядаються

- далі, якщо N точок (із попередніх 16) яскравіші за $I_p + t$ або темніші за $I_p - t$, то p можна вважати кутом. Для підвищення швидкості обчислення, спочатку можна порівняти яскравості точок під номером 1, 5, 9, 13. Якщо три

з чотирьох підходять за вищеописаною умовою, то p залишає за собою статус точки інтересу;

– попередні кроки повторюються для усіх пікселів зображення.

Таким чином, можна виділити формулу детектору FAST

$$S_{p \rightarrow x} = \begin{cases} d, & I_{p \rightarrow x} \leq I_p - t \\ s, & I_p - t < I_{p \rightarrow x} < I_p + t. \\ b, & I_p + t \leq I_{p \rightarrow x} \end{cases} \quad (1.1)$$

Через те, що FAST не враховує міру кутів, детектор дає великий відгук на прямі краї об'єктів. Це може стати причиною не інформативності знайденої множини КТ. У кутовій області візуального об'єкта градієнт значно змінюється в будь-якому напрямку, на відміну від краю, де зміни проходять вздовж самого напрямку ребра, або ж на рівній поверхні, де їх зовсім не має (рис. 1.7). Тож, кути вважаються кращими у ролі особливої точки і, як правило, більш стабільні при зміні ракурсу.

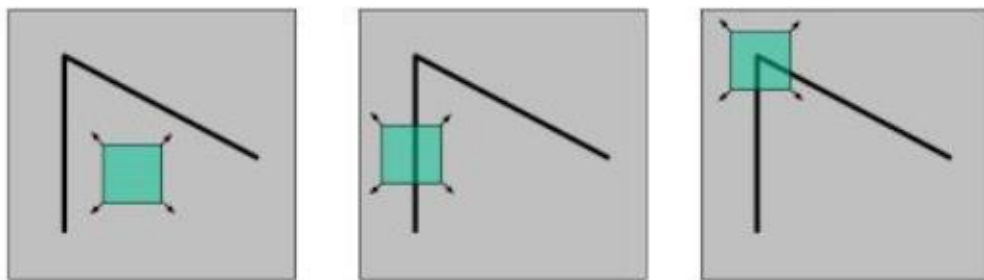


Рисунок 1.7 – Приклади порівнюваних областей об'єкту на зображенні

Для отримання більш точного ансамблю КТ, вводиться додаткова обробка множини, отриманої у результаті детектування методом FAST. У даному випадку, це кутовий детектор Харріса (рис. 1.8). Тоді, детектування N ключових точок проходить дещо по-іншому. Спочатку в хід йде FAST, але значення порогової яскравості t збільшується, щоб отримати множину

розміром більше за N . Після цього, за допомогою алгоритму детектування Харріса відбирається необхідна кількість особливих точок.

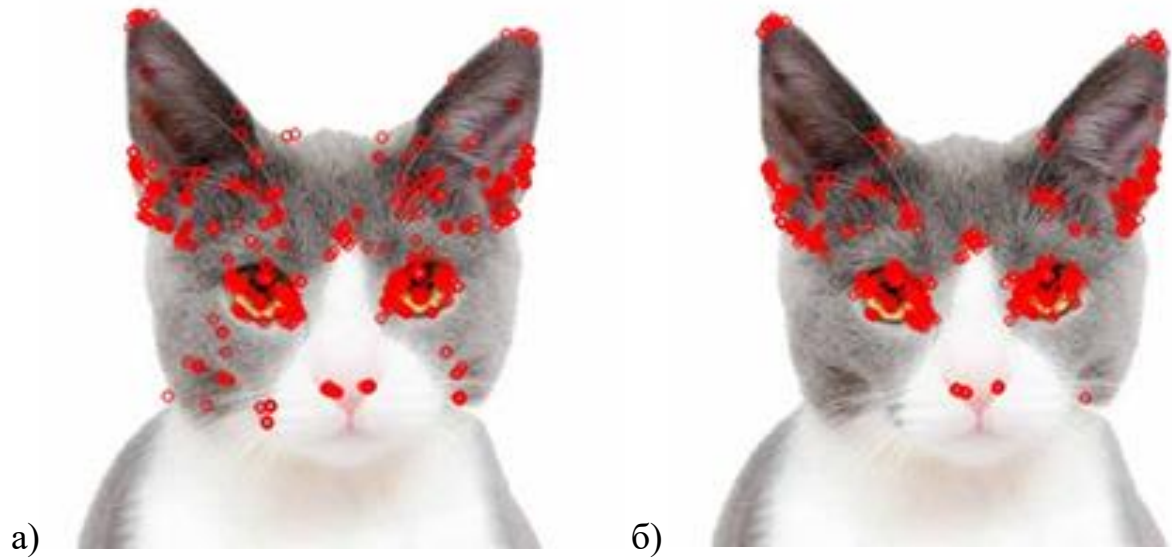


Рисунок 1.8 – Виявлення КТ:

а) – результат детектору FAST; б) – результат детектору Харріса

Ще одна особливість роботи ORB, що варто зазначити, полягає у побудові так званої піраміди зображень. *Піраміда зображень* – певна послідовність одного і того ж зображення, але з різним масштабом (рис. 1.9).

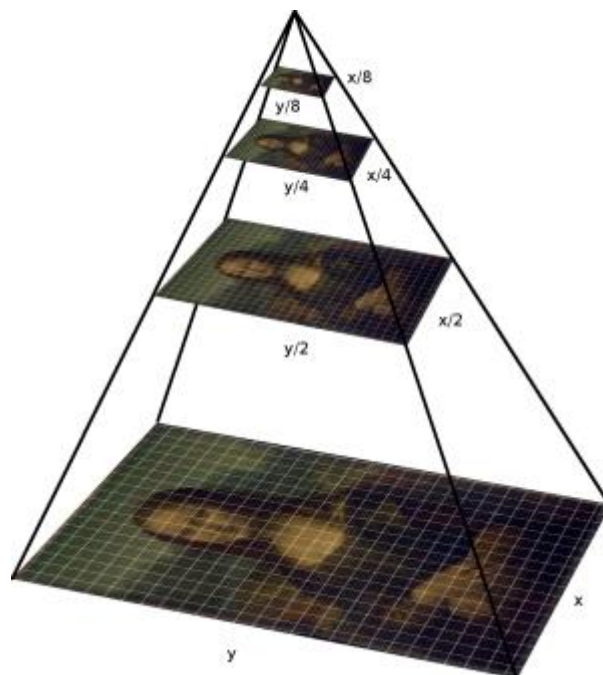


Рисунок 1.9 – Ілюстрація багатомасштабної піраміди зображення

Після її побудови, на кожному рівні піраміди викликається алгоритм FAST для виявлення ключових точок. Отримавши множини КТ ORB позиціонує їх, тим самим набуваючи властивості часткової інваріантності масштабу.

Далі, кожній точці вводиться параметр орієнтації, в залежності від того, як змінюється рівень яскравості навколо КТ. Для розрахування цієї зміни використовується центроїд інтенсивності C . Вважається, що C зміщений від ключової точки O , тож знайшовши вектор \overline{OC} , його можна використати для знаходження кута орієнтації (рис. 1.10).

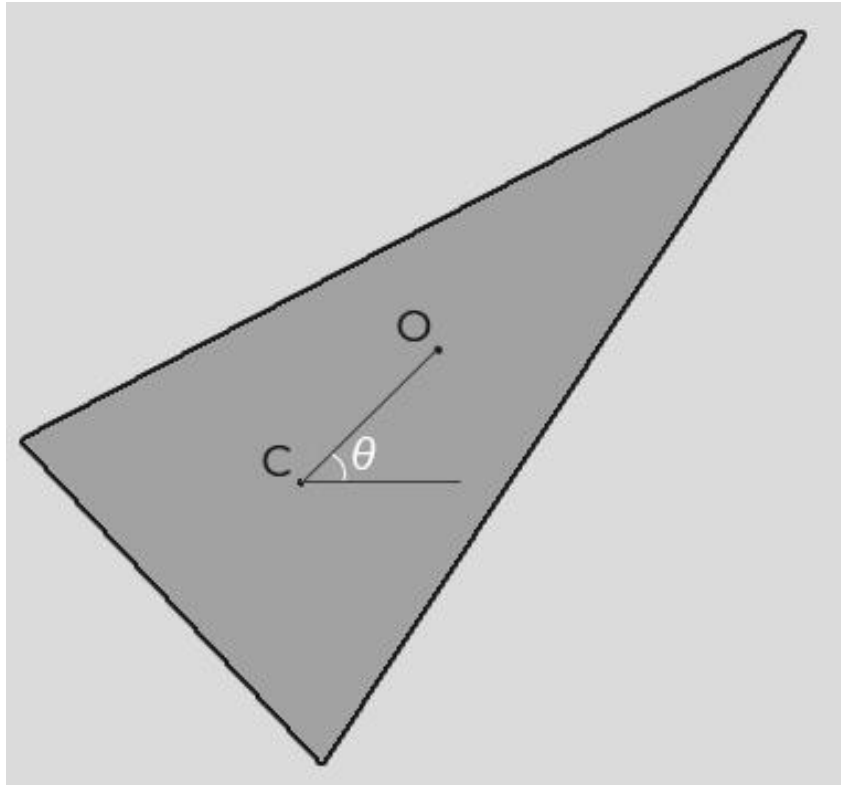


Рисунок 1.10 – Ілюстрація вектору \overline{OC} , де O – ключова точка, C – центроїд інтенсивності, θ – кут орієнтації

Формула координат центроїду наступна

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right), \quad (1.2)$$

де m_{pq} розраховується як

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y). \quad (1.3)$$

Тоді кут θ дорівнює

$$\theta = a \tan 2(m_{01}, m_{10}). \quad (1.4)$$

Після того як ORB розрахував орієнтацію даної ключової точки, вона повертається на θ , тим самим набуває свого канонічного вигляду. Це дозволяє отримати певну інваріантність до повороту зображення.

Наступним кроком після отримання множини особливих точок є знаходження дескрипторів для кожної із них. Для ORB це дескриптор *BRIEF* (*Binary Robust Independent Elementary Features*). BRIEF представляється у вигляді вектору розмірністю 128, 256 або 512 біт. Для його знаходження розглядається квадратний окіл певної КТ розміром n пікселів (патч), де виконуються бінарні тести τ між парами $(x; y)$ [9]

$$\tau(I; x, y) = \begin{cases} 1, & I_x < I_y \\ 0, & I_x \geq I_y \end{cases}, \quad (1.5)$$

де x, y – випадково обрані пікселі;

I_x, I_y – значення яскравості.

Для ORB $n = 31$, а як x, y виступають області 5×5 , при цьому I_x, I_y – середні значення яскравості для цих областей [10].

Сам вектор записується як

$$f_n(I) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(I; x_i, y_i). \quad (1.6)$$

На визначення дескрипторів досить сильно впливають шуми на зображенні, тож перед порівнянням застосовують згладжування по Гаусу [10].

Знаходження дескриптору залишає велику кількість варіантів вибору пар $(x; y)$. Для вирішення цього завдання обирається один із п'яти способів випадкового відбору (геометрії вибірки) зображених на рисунку 1.11 [11]. Вважається, що ключова точка знаходиться по центру області, що розглядається.

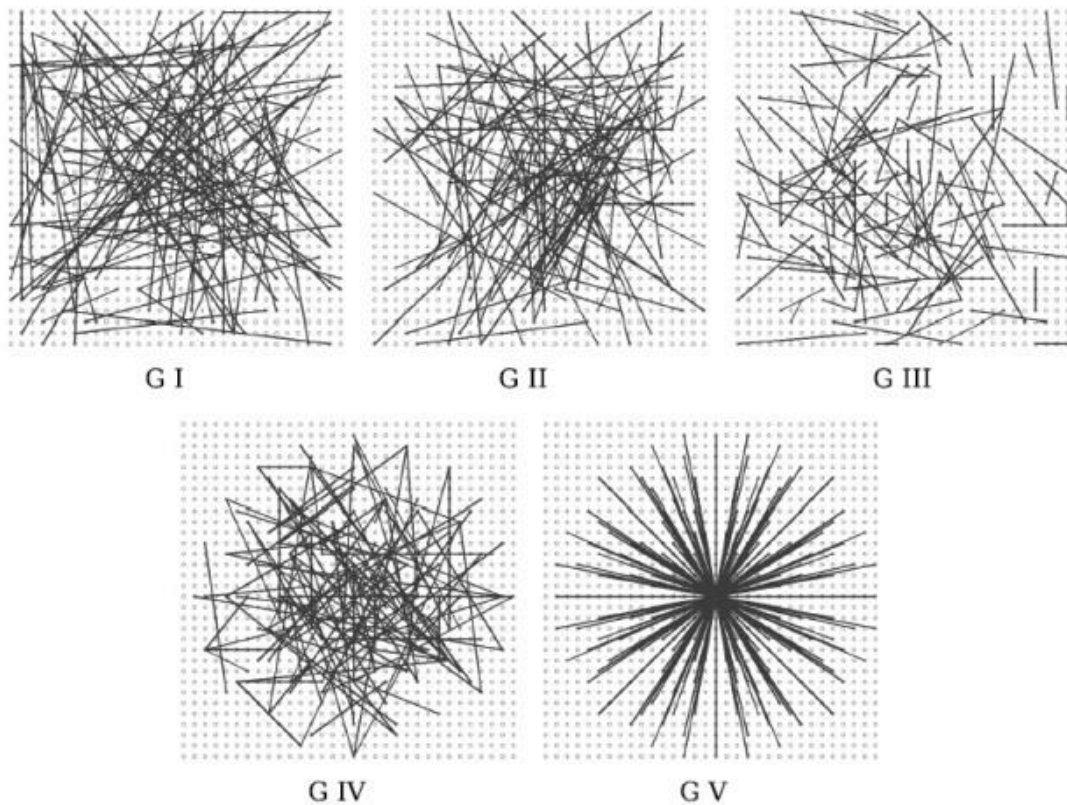


Рисунок 1.11 – Ілюстрація п'яти різних способів вибору точок для порівняння під час побудови вектору дескриптора BRIEF

G I: Точки x та y обираються випадкового за рівномірним розподілом.

G II: Точки x та y обираються випадково за розподілом Гауса.

G III: Точки x та y обираються випадково у два етапи. Спочатку, за розподілом Гауса обирається x відносно координат центру. Далі обирається y відносно x .

G IV: Точки x та y обираються випадково за допомогою дискретної радіальної сітки.

G V: Точка x – це центр, а y випадково обрана за допомогою дискретної радіальної сітки.

У літературі [11] дослідники провели експерименти з метою порівняння швидкості даних підходів. Отримані результати проілюстровані на рисунку 1.12.

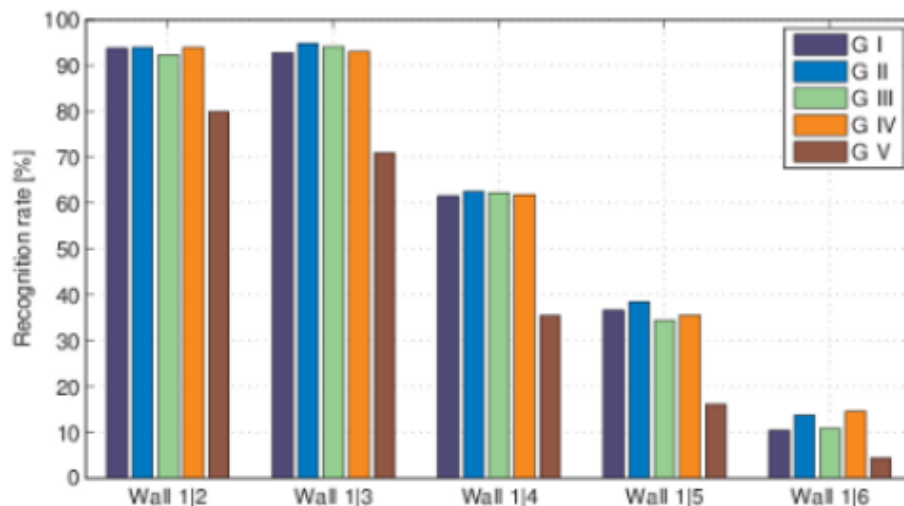


Рисунок 1.12 – Швидкість розпізнавання для п'яти різних шаблонів вибору пар для порівняння

З результатів дослідження видно, що патерн G V значно програє своїм суперникам, тож його використання не має сенсу. G II має невелику перевагу над іншими трьома. Саме цей спосіб використовується у алгоритмі ORB [10].

Звичайний дескриптор BRIEF не має можливості враховувати орієнтацію точки при детектуванні, але в ORB цей недолік вирішується за рахунок розрахованого кута θ («steered» BRIEF) [10].

Усі бінарні тести формують матрицю S

$$S = \begin{pmatrix} x_1, \dots, x_n \\ y_1, \dots, y_n \end{pmatrix}. \quad (1.7)$$

Використовуючи параметр орієнтації θ та матрицю повороту R_θ розраховується керована версія матриці (1.7)

$$S_\theta = R_\theta S. \quad (1.8)$$

Тоді вектор дескриптору дорівнюватиме

$$g_n(I, \theta) = f_n(I) | (x_i, y_i) \in S_\theta. \quad (1.9)$$

1.4 Постановка задачі

Сучасні задачі комп'ютерного зору мають потребу у швидко дієвих та ресурсозберезних функціях побудови класифікаторів з використанням множини дескрипторів КТ зображення для зіставлення візуальних об'єктів. Дослідження подібної процедури являється головним завданням даного дипломного проекту.

Ключова ідея полягає у використанні множини дескрипторів, які представлені у вигляді бінарних векторів, що визначаються заданою базою еталонних зображень для дослідження об'єктів класифікації на фундаментальному рівні.

Ґрунтуючись на цей простір ознак, знаходяться центри класів. У даному випадку це узагальнені медіани скінченної множини дескрипторів КТ. За допомогою відстані (метрики) знаходиться спектр аналізованих даних шляхом обчислення деякої функції належності для кожного з дескрипторів до класів, що, своєю чергою визначає статистичний розподіл даних.

Далі встановлюється клас аналізованого об'єкта через агрегацію розподілів кожного із класів за всією множиною складових опису. Класифікація відповідає найбільш правдоподібному рішення, бо побудована на додаванні значень однотипних розподілів.

Отже, спираючись на все вищесказане, об'єктом роботи є метод класифікації зображень на основі статистичних розподілів для множини дескрипторів ключових точок візуальних об'єктів.

Метою роботи є моделювання та дослідження працездатності методу класифікації зображень, що базується на побудові ансамблю розподілів для компонентів структурного опису з використанням їх інтегрального подання для кожного класу.

Досягнення мети включає у собі наступні етапи роботи:

- аналіз та огляд теоретичного матеріалу;
- підбір зображень для побудови бази еталонів;
- вибір технологій та програмна реалізація методу побудови класифікатора за моделлю, що розглядається;
- дослідження результатів класифікації;
- проведення додаткових експериментів з попередньою обробкою зображень;
- впровадження способів удосконалення реалізованої процедури класифікації.

2 МОДЕЛЬ КЛАСИФІКАЦІЇ НА ПІДГРУНТІ СТРУКТУРНИХ ОПИСІВ

2.1 Формалізація задачі класифікації

За основу візьмемо багатовимірний простір бінарних векторів B^n з розміром n [12]. Вважаємо, що дана сукупність задає головний простір ознак для бази еталонних зображень. Як опис візуального об'єкту слід вказати окрему мультимножину векторів $E_i \subseteq B^n$, яка являється множиною дескрипторів КТ [13-15]

$$E_i = \{e_v(i)\}_{v=1}^s, \quad (2.1)$$

де i – індекс еталону;

$s = \text{card } E_i$ – загальна кількість побудованих дескрипторів.

Скінченна множина векторів $e_k \in B^n$ задає ознаки, які є описом об'єкта на зображенні.

Для $\forall(e_k, e_\tau)$, де $e_k \in B^n$, $e_\tau \in B^n$ зазначимо відстань $\rho: B^n \times B^n \rightarrow [0, \infty]$ у просторі B^n . Оскільки дескриптори, тобто бінарні вектори, мають однаковий розмір, можна скористатися *метрикою* або, інакше, *відстанню Хемінга* [16]. Відстань Хемінга – число позицій, в яких відповідні символи двох двійкових слів однакової довжини рівні. На рисунку 2.1 проілюстровано відмінність двох пар двійкових рядків: 010 – 111 (результат 2) та 100 – 011 (результат 3). Дана відстань якнайкраще підходить для визначення розрізнення у метричному просторі для даного випадку. Потрібно лише знайти суму результатів операції XOR між двома двійковими дескрипторами, щоб мати можливість порівнювати їх [9].

Оскільки розмірність кожного вектору дорівнює n , то значення ρ лежатиме на інтервалі $[0, n]$.

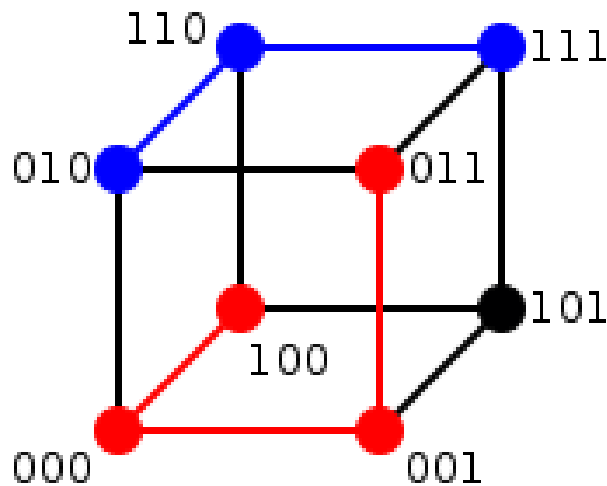


Рисунок 2.1 – Графічне подання Хемінгової метрики

Відстань являє собою фундаментальною мірою еквівалентності сукупності $\{e_k\}$, оскільки відображає візуальну подібність пікселів на околі деякої ключової точки для функції яскравості, яку використовують дескриптори [17]. Еквівалентність $e_k \sim e_\tau$ для двох дескрипторів e_k та e_τ визначається на основі порогового значення δ_ρ для результуючих значень метрики:

$$e_k \sim e_\tau : \rho(e_k, e_\tau) \leq \delta_\rho. \quad (2.2)$$

Для класифікації необхідно мати певну базу E описів еталонних зображень розмірністю N , тоді $E = \{E_1, E_2, \dots, E_N\}$. Для класифікації, дана сукупність представляє множину класів, де кожен із них являється скінченним набором бінарних дескрипторів ключових точок із загального простору B^n .

Головна мета досліджуваного методу полягає у використанні певної бази $E = \{E_1, E_2, \dots, E_N\}$ для побудови класифікатора $K : B^n \rightarrow [1, 2, \dots, N]$, шляхом знаходження ймовірнісної системи ознак, використовуючи фундаментальні відомості про зображення – набір дескрипторів КТ.

Основна ідея даної класифікації: використовуючи всю множину побудованих дескрипторів певного візуального об'єкту, встановити ступінь належності кожного до заданих класів у вигляді статистичного розподілу. Далі, на основі отриманої системи розподілів, шляхом агрегації побудувати ансамблеву міру релевантності для визначення відповідності даного об'єкту до кожного з центрів та використати її як апарат прийняття класифікаційних рішень.

2.2 Побудова ансамблю розподілів для компонентів

Перед початком класифікації, необхідно певним чином трансформувати кожний опис еталонного об'єкту (2.1) із заданої бази E для отримання центру класифікації у вигляді бінарного вектору, який обчислюється на підставі множини дескрипторів E_i та відносно якого будуватимуться статистичні розподіли [18]

$$\alpha(i) = (\alpha_1(i), \alpha_2(i), \dots, \alpha_n(i)). \quad (2.3)$$

Обчислення $\alpha(i)$ для кожного із еталонів являється основою класифікації (більш детально у п.п. 2.3).

Своєю чергою, вектор $e_v(i) \in E_i$, як і будь-який дескриптор, можна задати як [19]

$$e_v(i) = (e_{v,1}(i), e_{v,2}(i), \dots, e_{v,n}(i)). \quad (2.4)$$

Далі, необхідно представити вхідні дані для аналізу, тобто множину дескрипторів, що представляє опис деякого об'єкта, у вигляді обчисленої функції належності μ до заданих класів із значенням від 0 до 1 [20-22]:

$$\mu: B^n \rightarrow [0,1], \mu(e_v(i)) \in [0,1]. \quad (2.5)$$

Параметрами функції μ є вектор $e_v(i) \in E_i$ та номер класу. Саму функцію належності μ можна визначити за допомогою співвідношення значень мір, що задають число сприятливих випадків та загального числа випадків N що відповідає кількості класів [21]

$$\mu(e_v(i)) = \frac{\eta(e_v, i)}{\sum_{i=1}^N \eta(e_v, i)}, \quad (2.6)$$

де $\eta(e_v, i)$ – числове значення міри подібності для відповідного дескриптору та класу.

Вважаємо, що $\alpha(i) \in B^n$, детальніше у п.п. 2.3. Через те, що $e_v(i) \in B^n$ міру подібності $\eta(e_v, i)$ можна визначити за допомогою раніше згаданої Хемінгової метрики $\chi(e_v, i)$ [19]. Тоді формула $\eta(e_v, i)$ буде виглядати як

$$\eta(e_v, i) = n - \chi(e_v, i). \quad (2.7)$$

Розрахувавши усі значення $\eta(e_v, i)$ для елемента $e_v(i)$, можна обчислити значення вектору d_v , що представляє собою статистичний розподіл дескриптора $e_v(i)$ за множиною еталонів N , скориставшись виразом (2.6) [18]

$$d_v = (d_v(1), d_v(2), \dots, d_v(N)),$$

$$d_v(i) = \mu(e_v(i)), \quad (2.8)$$

$$\sum_{i=1}^N d_v(i) = 1.$$

Знайшовши усі значення d_v для всієї множини E_i , можна побудувати матрицю $D = \{\{d_v(i)\}_{v=1}^s\}_{i=1}^N$, що, за допомогою (2.6), визначає числове значення міри залежності для усіх дескрипторів. Для спрощення подальшого обчислення, вважаємо, що значення s (розмір множини дескрипторів для одного зображення) однакове для усієї бази еталонів. Таблиця 2.1 ілюструє структуру матриці D .

Таблиця 2.1 – Структура матриці розподілів D

Елементи опису	Класи		
	1	...	N
1	$d_1(1)$...	$d_1(N)$
...
s	$d_s(1)$...	$d_s(N)$

Побудова матриці D дозволяє запровадити попередню обробку вхідних даних, щоб виділити найбільш вагомні дескриптори шляхом аналізу значень (2.6) або (2.7). Однією із переваг даного підходу є те, що можна використати певні порогові значення для фільтрації отримуваної інформації на відповідному етапі класифікації, тим самим зменшивши час обчислення на наступних за рахунок стиснення множини опису. Це дозволить здобути більш інформативний опис об'єктів та, безпосередньо, поліпшить точність результатів.

Використовуючи значення матриці D будується класифікатор $K : D \rightarrow [1, 2, \dots, N]$ для візуального об'єкта, що аналізується. Варто зазначити, що результат такої процедури цілком залежить від вхідної бази еталонних зображень. На рисунку 2.2 зображена схема даної моделі побудови класифікатору.



Рисунок 2.2 – Схема класифікації

2.3 Знаходження центру класу

Для даної моделі класифікації центром класу виступає *медіана*. Медіана в математичній статистиці – це числова характеристика множини, що ділить її на дві рівні половини, де в одній значення менші за неї, в іншій більші [23]. Для знаходження медіани, множини необхідно упорядкувати за зростанням. Елемент, що знаходиться посередині відсортованої вибірки і є медіана. Якщо кількість значень в множині парна, то, зазвичай, обираються два центральних сусідніх числа та знаходять середнє значення між ними.

На рисунку 2.3 зображено порівняння медіани із середнім значенням.

Інше рівносильне визначення, медіана – це число, у якого сума модулів (відстаней) до усіх інших елементів множини мінімальна. Дана інтерпретація більше підходить у даному випадку, так як множина представлена множиною двійкових векторів, відстань яких можна визначити у метричному просторі.

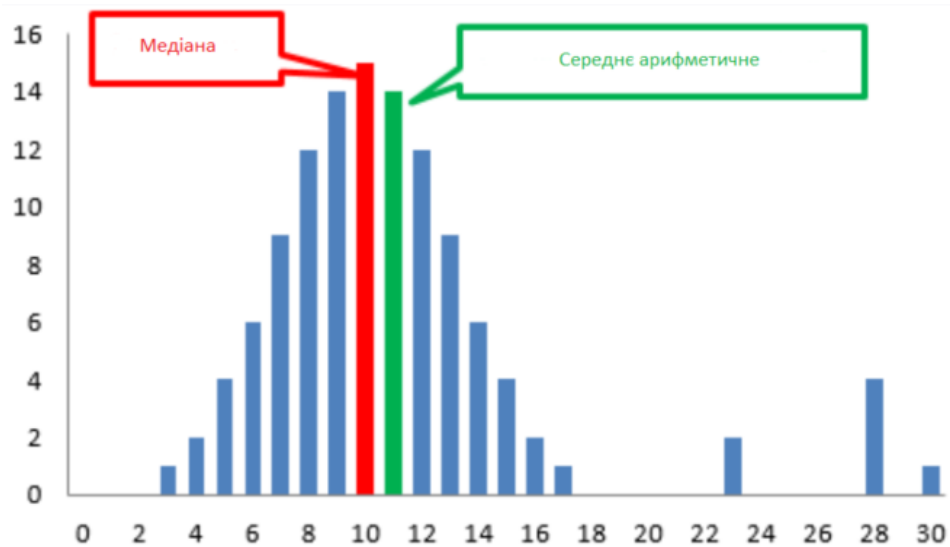


Рисунок 2.3 – Порівняння медіани та середнього значення

У даному випадку, медіана m^i множини (2.1) визначається шляхом мінімізації функціоналу D_i – матриці відстаней між кожною парою дескрипторів [24-25]:

$$m^i = \arg \min_{v \in E_i} D_i(v), \quad (2.9)$$

$$D_i(v) = \sum_{e_v(i) \in E_i} \rho(e_v(i), v),$$

де $\rho(e_v(i), v)$ – відстань між елементами множини $E_i \subset B^n$.

Чим менше значення D_i , тим менше відрізняється число елементів зліва та справа від значення m^i за відстанню до m^i . Використавши це, можна виділити кроки алгоритму знаходження медіани m^i для опису еталонного зображення E_i [22]:

– знаходження симетричної матриці відстаней розміром $s_i \times s_i$ між кожною парою елементів множини E_i ;

– розрахування сум $D_i(1), D_i(2), \dots, D_i(s_i)$, шляхом додавання елементів в стовпцях / рядках матриці D_i ;

– знаходження мінімального значення серед знайдених сум

$$m^i = \arg \min_{v \in E_i} D_i(v).$$

У результаті отримуємо медіану m^i як один із елементів еталону E_i , тобто це бінарний вектор опису з такою же розмірністю як і інші у своєму локальному середовищі. Це дає змогу використовувати відстань Хемінга у виразі (2.7), щоб розраховувати (2.6) замість більш складних за обчисленнями метрик.

Підсумовуючи, варто сказати, що визначення центрів класів забезпечує підвищення швидкості алгоритмів класифікації, бо дозволяє використовувати порівняння окремих векторів замість обчислення вагомих мір релевантності для множин.

2.4 Модель класифікатора

У основі побудови класифікатора лежать збудовані розподіли (2.8) за кожним класом у відповідності до наявної бази еталонних зображень. Далі, завданням є для кожного існуючого $E_i \in E$ отримати класифікатор K , значення якого повинно лежати на інтервалі $[1, 2, \dots, N]$ та відповідати певному номеру еталону.

Це є головним показником правильного функціонування класифікатора, який повинен впевнено розрізняти описи із множини еталонів [26]. Наприклад, у розподілі даних для першого еталону (перший стовпець матриці D) перша компонента повинна домінувати над іншими. Відповідно до цього принципу, знаходяться значення для усієї бази еталонів.

У даному дипломному проекті, принцип побудови класифікатора лежить на основі обчислення сум кожного стовпця матриці D та виділення максимального значення серед знайдених значень

$$K : j = \arg \max_i \sum_{v=1}^s d_v(i). \quad (2.10)$$

У результаті того, що дана класифікація побудована на додаванні значень однотипних розподілів, вона відповідає найбільш правдоподібному рішенню [27].

Використання ансамблю класифікаторів, зазвичай, забезпечує кращу точність отримуваних результатів аналізу або навчання, та під час виконання процедури можуть виникнути проблеми зв'язані зі збільшенням часу, підвищенням потреби у обчислювальних ресурсах, бо необхідно враховувати великий обсяг вхідних даних. Для цього можна використати систему порогів, яка дозволить виділити тільки вагомні складові, тим самим оптимізуючи алгоритм у результаті зменшення кількості даних для обробки, та й, загалом, підвищуючи надійність класифікації.

Дивлячись на це, модель (2.10) можна дещо модифікувати, шляхом додаткової фільтрації на момент розрахунку (2.9). Вводиться додаткова умова, що порівнює значення $d_m = \max_{i=1,2,\dots,N} \{d_v(i)\}$ з порогом δ_d , або з найближчим локальним оптимумом d_{m-1} (λ – числовий коефіцієнт) [28]:

$$\begin{aligned} d_m &> \delta_d, \\ \text{або } d_m &> \lambda d_{m-1}. \end{aligned} \quad (2.11)$$

Варто зазначити, що значення порогу δ_d можна встановити за результатами навчання із вчителем на множині структурного опису (дескрипторів) для еталонних зображені. За умову можна взяти те, що результативне значення моди має бути не меншим за його значення для відповідного класу. Цей аналіз треба проводити на етапі навчання класифікатора для усіх еталонів і у якості порогу δ_d обрати найбільше серед усіх отриманих мод.

Ще один із способів підвищення результативності класифікації полягає у введенні додаткових логічних процедур для отримання найбільш інформативної множини структурного опису, яка стане фундаментом для подальшої обробки. Якщо опис візуального об'єкта заданий множиною побудованих дескрипторів КТ (як у даному випадку), то можна ввести попередню обробку на знаходження фіксованої підмножини елементів найближчих до заданого центру.

Нехай $Z \subset B^n$ – множина з розміром $s = \text{card } Z$, що задає опис певного об'єкта. Тоді введемо

$$L(Z) \rightarrow Z^*,$$

$$Z^* \subset B^n,$$

(2.12)

$$\text{card } Z^* = s^*,$$

$$s^* < s.$$

Використання (2.12), як і (2.11), зменшує час обчислення та сприяє покращенню результативності. Перевагою даного методу є те, що обробка проводиться на початковому етапі, під час побудови опису еталонів. Тож, в результаті, вона не впливає на сам час класифікації.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНИЙ АНАЛІЗ РЕЗУЛЬТАТИВНОСТІ КЛАСИФІКАЦІЇ

3.1 Опис використаних програмних інструментів

Для реалізації алгоритму класифікації зображень за описаною моделлю був створений застосунок мовою програмування C# з використання бібліотеки для обробки графіки Emgu CV.

C# – це сучасна високорівнева мова, що підтримує об'єктно-орієнтовану модель програмування з безпечною системою типізації для платформи .NET, та має великий багаж заздалегідь реалізованого інструментарію для швидкого створення застосунків. Через те, що дипломний проєкт має аналітичний та експериментальний характер, й програмний продукт не потребує подальшого використання, вирішено створювати консольний застосунок. Це дозволило зберегти час, відмовившись від побудови інтерфейсу користувача.

Для обробки множини дескрипторів та інших масивів даних використана технологія під назвою LINQ. LINQ (Language-Integrated Query) являється простою та зручною мовою запитів до даних. Це не зовнішня бібліотека, а вбудований інструментарій в платформу .NET, що знаходиться у просторі імен System.Linq. Джерелами даних можуть виступати об'єкти, колекції, XML файли та інші. Не залежно від типу джерела, LINQ дозволяє застосовувати один і той же підхід обробки. Цей інструмент дає можливість швидко реалізувати задану модель класифікації, не витрачаючи час на додаткову програмну логіку, бо має значну кількість методів для сортування, знаходження мінімального та максимального значень, пошук елементів за умовою, групування та багато іншого.

Emgu CV – це .NET обгортка з відкритим кодом для графічної бібліотеки OpenCV, що дозволяє використовувати різноманітні функції та алгоритми обробки зображень під час написання додатків на сумісних з .NET

мов для Windows, Linux, Mac OS, iOS та Android. Інструментарій даної бібліотеки дозволяє створювати рішення для різноманітних задач з 2D графікою, розпізнаванням об'єктів на фото і відео та інші.

Функціональність Emgu CV дозволяє лише за допомогою однієї функції знайти необхідний масив даних, при цьому даючи можливість корегувати наступні параметри:

- максимальне число дескрипторів (за замовчуванням 500);
- коефіцієнт, на який ділиться розмір одного рівня піраміди на інший (за замовчуванням 1,2);
- кількість рівнів піраміди зображень (за замовчуванням 8);
- число пікселів на границі зображення, де КТ не визначаються (за замовчуванням 31);
- початковий рівень піраміди для дослідження (за замовчуванням 0);
- число точок, які необхідні під час бінарного тесту для визначення значення дескриптору (за замовчуванням 2);
- можливість використання додаткової обробки використовуючи кутовий детектор Харріса (за замовчування ввімкнено);
- розмір місцевості КТ, використаний при визначенні значення дескриптору (за замовчуванням 31);
- поріг яскравості для FAST (за замовчуванням 20).

Також, використання Emgu CV дає можливість показувати ключові точки на вхідних зображеннях, бо бібліотека дозволяє з легкістю використовувати, обробляти та зберігати їх, підтримуючи різні типи (формати) статичних медіафайлів.

Для реалізації необхідного застосунку використовується середовище розробки Visual Studio 2019 від компанії Microsoft. Воно дозволяє створювати різноманітні програмні продукти, як, наприклад, консольні програми, програми з графічним інтерфейсом, віконні додатки Windows Forms, WEB-застосунки та багато іншого. Visual Studio дозволяє використовувати для розробки різні мови програмування: C#, Visual Basic,

F#, C, C++, Python, тощо. Для даного дипломного проєкту була використана версія Community, яка є абсолютно безплатною для учнів, студентів та розробників програм з відкритим вихідним кодом. Вона відрізняється від Pro версії тільки наявністю додаткових функцій для аналізу, тестування застосунків та іншим, що не є необхідним у даному випадку.

Інструментарій обраних технологій дає змогу просто та за короткий час створити застосунок, щоб дослідити задану модель класифікації, але при цьому жертвуючи оптимізацією. Напевно, використання таких мов як C++ та Python із засобами OpenCV дало б можливість написати більш ресурсозбережний застосунок, але в рамках даного проєкту це не потрібно.

Ще одним недоліком використаного інструментарію є доволі низька популярність в мережі Інтернет у порівнянні із парою Python + OpenCV, де на кожне запитання можна відразу знайти відповідь. Як результат, витрачено додатковий час на вивчення офіційної документації Emgu CV.

3.2 Експериментальний аналіз результативності методу класифікації

У ході виконання дипломної роботи проведено моделювання відповідно до методу, викладеному у розділі 2. Для дослідження обрано базу у вигляді трьох еталонних зображень тварин: kota, собаки та свині. Зображення мають розмір 512×512 пікселів та у форматі JPEG. Кожний візуальний об'єкт розташовується на світлому фоні (рис. 3.1). За допомогою засобів Emgu CV будується 500 дескрипторів ORB розміром 256 біт для кожного зображення. Для ORB детектора, усі додаткові параметри використовують значення за замовчування. Весь цей перелік наведений у п.п. 3.1.

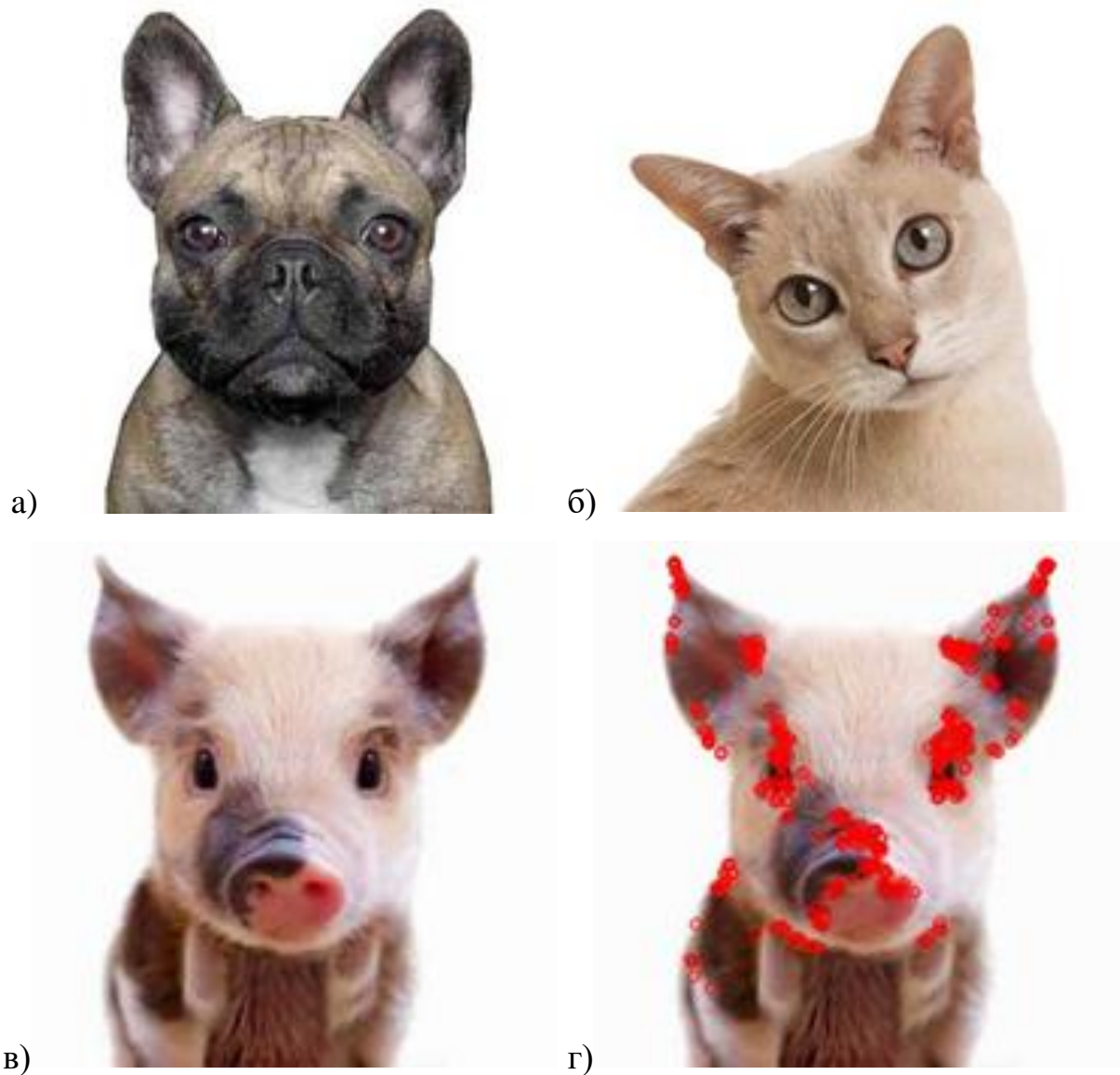


Рисунок 3.1 – База еталонних зображень:

а), б), в) – вхідні зображення; г) – виділені ORB дескриптори

Наступним кроком методу є знаходження центру опису для класифікації, у даному випадку – медіани m для кожного еталону. Для цього необхідно побудувати матрицю відстаней між дескрипторами множини опису та за формулою (2.9) виділити один із них, для якого сума відстаней, у порівнянні з іншими, мінімальна.

Оскільки кожен дескриптор має однаковий розмір, метрикою служитиме відстань Хемінга, яка дорівнює числу розбіжностей між двома двійковими виразами. Зрозуміло, що максимально можливе значення складатиме 256, бо це загальна розмірність дескрипторів.

Для простоти подальшого опису, кожному еталону присвоєно порядковий номер: 1 – собака, 2 – кіт та 3 – свиня.

Конкретно відстані між медіанами склали:

- для 1-го та 2-го еталонів – 108 (42% від максимальної відстані);
- для 1-го та 3-го – 119 (46,5% від максимальної відстані);
- для 2-го та 3-го – 41 (16% від максимальної відстані).

Отримані результати показали, що у даному аспекті, що відображає використовуваний простір даних, еталонні зображення 2 та 3 досить схожі, але під номером 1 достатньо відрізняється від попередніх.

Далі застосовуємо описану у розділі 2 модель класифікації, що встановлює агрегований ступінь належності зображення до певного еталону. Для прикладу розглянемо класифікацію 1-го еталонного зображення.

Для початку необхідно знайти числову характеристику подібності кожного дескриптору зображення з кожним еталонним центром. Це можна знайти, розрахувавши відстань Хемінга між кожним дескриптором та кожною медіаною, після чого відняти отримані значення від максимальної відстані (2.7). Результати обчислення занесені до таблиць 3.1 та 3.2.

Таблиця 3.1 – Відстані Хемінга між першими 10 дескрипторами еталонного зображення та кожним класом

Номер дескриптору	Відстані до класів		
	1	2	3
1	136	136	141
2	106	76	69
3	130	152	169
4	97	83	88
5	110	88	77
6	104	94	79

Продовження таблиці 3.1

Номер дескриптору	Відстані до класів		
	1	2	3
7	128	118	119
8	112	124	111
9	91	107	96
10	143	111	118

Таблиця 3.2 – Числова характеристика подібності для 10 дескрипторів

Номер дескриптору	Характеристика подібності до класу		
	1	2	3
1	120	120	115
2	150	180	187
3	126	104	87
4	159	173	168
5	146	168	179
6	152	162	177
7	128	138	137
8	144	132	145
9	165	149	160
10	113	145	138

Використовуючи отримані значення подібності, за виразом (2.6) розраховуємо функцію належності μ та отримуємо матрицю D . Таблиця 3.3 ілюструє отримані результати.

Таблиця 3.3 – Результати розрахунку функції належності для 10 дескрипторів

Номер дескриптору	Значення функції належності		
	1	2	3
1	0,338	0,338	0,324
2	0,290	0,348	0,362
3	0,397	0,328	0,274
4	0,318	0,346	0,336
5	0,296	0,341	0,363
6	0,310	0,330	0,360
7	0,318	0,342	0,340
8	0,342	0,314	0,344
9	0,348	0,314	0,388
10	0,285	0,366	0,348

Склавши усі значення кожного стовпця матриці, отримано вектор сум – (171,675; 165,451; 162,87), за допомогою якого можна класифікувати зображення під номером 1, знайшовши максимальне значення у даному векторі. Шукане значення відповідає 1-му стовпцю, тож класифікація здійснена вірно.

Повторивши попередні кроки для інших двох зображень, результат залишився незмінно правдивим: (156,74; 172,32; 170,95), (152,46; 173,07; 174,47).

Введемо величину $\Delta = h_1 - h_2$, що дорівнює різниці між максимальним значенням вектору сум та найближчим до нього конкурентом у відсотках. Експериментальні значення Δ для розглянутих еталонів дорівнюють 3,63%, 0,79%, 0,81%. Незважаючи на правильний результат класифікації, значення Δ досить низькі. Вони вказують на суттєву схожість образів у сформованому просторі ознак, що ставить застереження щодо забезпечення

достатньої результативності запропонованого алгоритму класифікації у складних умовах геометричних перетворень та впливу завад на аналізоване зображення.

3.3 Перевірка стійкості розробленого методу класифікації

Щоб переконатися в працездатності розробленої моделі класифікації, необхідно провести додаткові експерименти з використанням різних фільтрів, перетворень на зображеннях. Це дозволить перевірити стійкість алгоритму та при наявності позитивних результатів виділити його серед інших у сфері розпізнавання об'єктів.

Суть експериментів у наступному. Як і у попередньому пункті використані 3 еталони, що задають 3 центри класів. Далі, одне із поданих зображень (рис. 3.2) підлягає деякій обробці, після чого починається процес класифікації. Завданням є порівняти результати з попередніми та знайти закономірності погіршення роботи алгоритму.

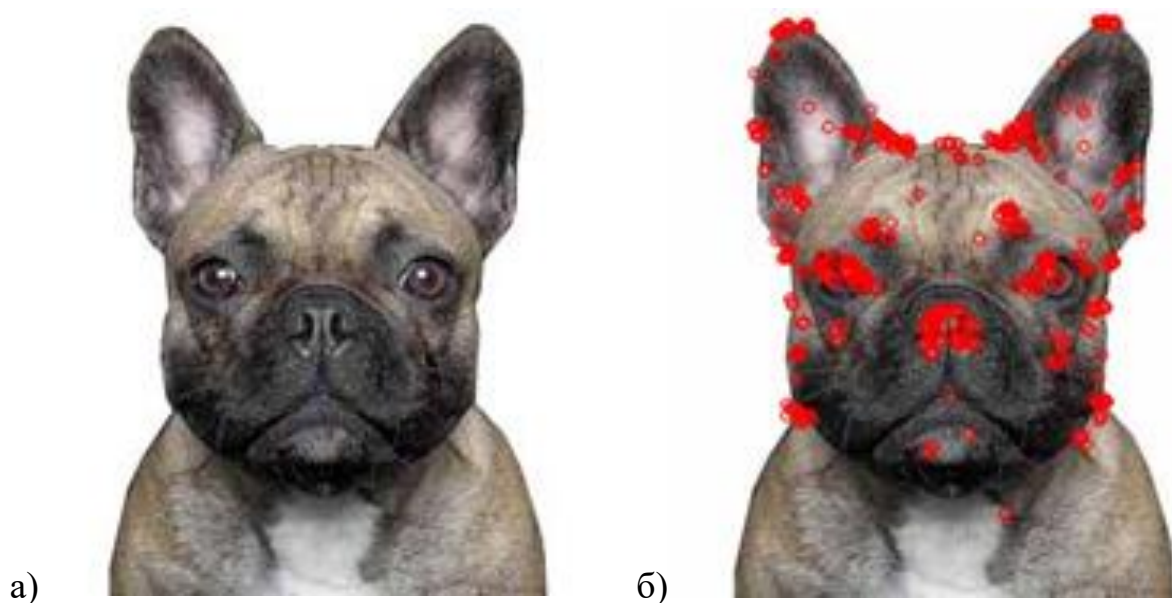


Рисунок 3.2 – Еталонне зображення, що підлягатиме обробці:

а) – вхідне зображення; б) – виділені дескриптори КТ ORB

Першим об'єктом перевірки стає шум «солі та перцю» (*Salt & Pepper noise*) [29]. Він отримав таку назву за рахунок свого відображення – це сукупність випадкових білих або чорних пікселів на зображенні. Цей шум зазвичай викликаний, відмовою певних областей пам'яті, помилками під час передачі даних або роботи аналого-цифрового перетворювача.

Інструментарій EmguCV не має у собі реалізацію для імітації даного шуму, тож знадобилося створити її самостійно. Програмний код алгоритму зображено на рисунку 3.3.

```

/*
 *
 * image - вхідне зображення
 * amount - об'єм шуму на зображенні: [0, 1]
 * saltAmount - об'єм білих крапок на зображенні: [0, 1]
 *
 */
public static void AddSaltPapperNoise(Image<Bgr, byte> image, float amount, float saltAmount)
{
    // знаходження максимальної кількості білих крапок на зображенні
    int saltCount = (int)Math.Ceiling(image.Width * image.Height * amount * saltAmount);
    SetColorOnRandomCoordinates(image, saltCount, new Bgr(255, 255, 255));

    // знаходження максимальної кількості чорних крапок на зображенні
    int papperCount = (int)Math.Ceiling(image.Width * image.Height * amount * (1.0f - saltAmount));
    SetColorOnRandomCoordinates(image, papperCount, new Bgr(0, 0, 0));
}

private static void SetColorOnRandomCoordinates(Image<Bgr, byte> image, int count, Bgr color)
{
    Random random = new Random();

    int x, y;
    for (int i = 0; i < count; i++)
    {
        x = random.Next() % image.Width;
        y = random.Next() % image.Height;

        image[x, y] = color;
    }
}

```

Рисунок 3.3 – Програмна реалізація алгоритму додавання шуму «солі та перцю»

Даний метод приймає у вхідні параметри зображення (*image*), об'єм шуму (*amount*) та об'єм білих пікселів (*saltAmount*). Під об'ємом шуму мається на увазі відсоток пошкоджених точок у еквіваленті від 0 до 1. За допомогою цього параметру і розміру зображення обчислюється максимальна кількість пікселів, що підлягає обробці. Максимальна, тому що

координати є випадковими та можуть повторюватися. Через це, розмір вихідної множини проявлених дефектів може бути менший за знайдену кількість. Подібний недолік можна було б вирішити, але, у даному випадку, це не грає великої ролі та цим можна знехтувати, бо розмір досліджуваних зображень достатньо великий – 512×512 , що дорівнює 262 144 пікселям.

Наступним параметром є об'єм білих точок, який подібно попередньому варіюється на інтервалі від 0 до 1 та відповідає відсотку білих пікселів на зображенні. Вираз $(1 - saltAmount)$ дозволяє знайти обсяг чорних крапок.

Для тесту використані наступні значення параметрів: об'єм шуму дорівнює 0,005, а об'єм білих точок – 0,5. З рисунка 3.4 можна побачити, що положення КТ значно змінилося від «чистого» зображення. Вони розбіглися по всій поверхні, що в значній мірі може змінити результати класифікації.

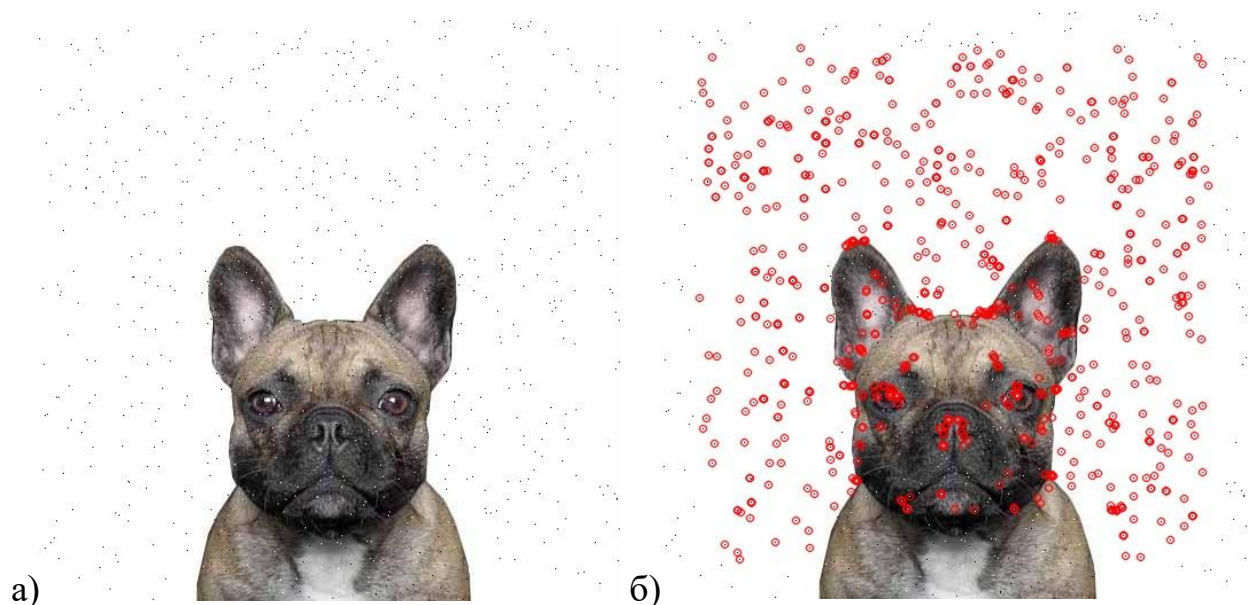


Рисунок 3.4 – Зображення під дією шуму «солі та перцю»:

а) – вихідний результат; б) – з виділеними дескрипторами КТ

Обчислені результати експерименту занесені до таблиці 3.4. Варто зазначити, що для більшої точності дослідження, необхідно провести

тестування усіх зазначених еталонних зображень та через обмежені часові рамки було прийняте рішення використати тільки один рисунок.

Таблиця 3.4 – Результат експерименту з шумом «солі та перцю»

Стан зображення	Сума розподілів		
	1	2	3
Без шуму	171,675	165,451	162,87
З шумом «солі та перцю»	172,076	163,113	164,811

Дивлячись на отримані результати, можна припустити, що алгоритм пройшов дане випробування. Класифікатор відповідає необхідному еталону (під номером 1), а значення різниці між двома максимальними числами вектору сум Δ дорівнює 5,2%. Деякою мірою це навіть покращило попередні результати. Та варто не забувати, що об'єм шуму, який використовувався достатньо малий, всього 0,005. Уже при збільшенні його до 0,015, значення векторів сум майже не відрізняються: (167,092; 166,144; 166,763). Тобто можна сказати, що при пошкодженні зображення білим шуму близько до 1,5% і більше, якість класифікації значно зменшується, що робить даний підхід нестійким до шуму «солі та перцю» для даної бази еталонів. Має сенс провести додаткові випробування для зображень з різнокольоровим фоном, так як у даному випадку чорні крапки сильно виділяються на білому кольорі. Це стало причиною розкиду множини КТ та зменшення фокусу на самому об'єкті, а саме від цього аспекту залежить успішність класифікації.

Вирішено провести додаткові тести для усунення пошкоджень на зображеннях та повторної класифікації. Бібліотека Emgu CV має достатню кількість заздалегідь реалізованих методів згладжування, тож даний дослід не забере багато часу. Щоб протидіяти шуму «солі та перцю» був використаний *медіанний фільтр* [29]. Ідея цього методу полягає в обробці кожного пікселя зображення, шляхом знаходження медіани значень множини сусідніх точок. Особливістю шуму «солі та перцю» є те, що він покриває

тільки окремі точки, а не повністю все зображення. Тож, медіанний фільтр, який враховує окіл для розрахування значення пікселя, якнайкраще підходить для фільтрації.

Метод *SmoothMedian* відповідає за медіанний фільтр у бібліотеці Emgu CV. Приймає один параметр *size*, який відповідає висоті та ширині області, яка приймає участь у розрахуванні значення пікселя. Для експерименту, *size* візьмемо вагою 3, а об'єм шуму нехай дорівнює 0,015, бо, під час першого тесту, починаючи з цього значення з'являлися проблеми класифікації. Отримані результати у таблиці 3.5.

Таблиця 3.5 – Результат експерименту з шумом «солі та перцю» та медіанним згладжуванням

Стан зображення	Сума розподілів		
	1	2	3
Без шуму	171,675	165,451	162,87
З шумом «солі та перцю»	167,092	166,144	166,763
З медіанним згладжуванням	171,716	165,498	162,786

Як і очікувалося, медіанний фільтр покращив класифікацію. Отримано коректне значення класифікатору та самі значення вектору сум майже не відрізняються від вихідних. Навіть зображення із значними пошкодженнями (рис. 3.5), використовуючи даний фільтр правильно класифікується: при об'ємі шуму 0,6 отримано відносно вагомі результати (173,966; 162,351; 163,683).

Підсумовуючи вищесказане, можна зробити висновок, що даний вид шуму значно впливає на кінцевий результат класифікації та це все вирішується правильним вибором алгоритму фільтрації. Тож, під час використання даного алгоритму необхідно враховувати початкову базу еталонних зображень для зменшення ймовірності хибних результатів.



Рисунок 3.5 – Сильно пошкоджене шумом «солі та перцю» зображення

Наступний експеримент проведений під дією *Гауссівського шуму* [29]. Він будується шляхом обчислення нового значення інтенсивності для кожного пікселя зображення. Нове значення дорівнює сумі фактичного та випадково знайденого числа за нормальним розподілом. Для імітації додавання даного шуму, був реалізований метод *AddGaussianNoise*, для роботи якого потрібне вхідне чорно-біле зображення та об'єм шуму. Детальніше на рисунку 3.6.

```
public static Image<Gray, byte> AddGaussianNoise(Image<Gray, byte> image, float amount)
{
    var noise = image.Clone();
    noise.SetRandNormal(new MCvScalar(128), new MCvScalar(30));
    noise = image + noise * amount;

    return noise;
}
```

Рисунок 3.6 – Програмна реалізація алгоритму додавання Гауссівського шуму

Було взято математичне очікування вагою 128 та середнє квадратичне відхилення – 30. Об'єм шуму дорівнюватиме 0,3.

З рисунка 3.7 можна зробити висновок, що експеримент набуває більш позитивного характеру у порівнянні з попереднім, так як множина КТ зосереджена на самому об'єкті, що дозволить більш точно описати його у даному просторі ознак.

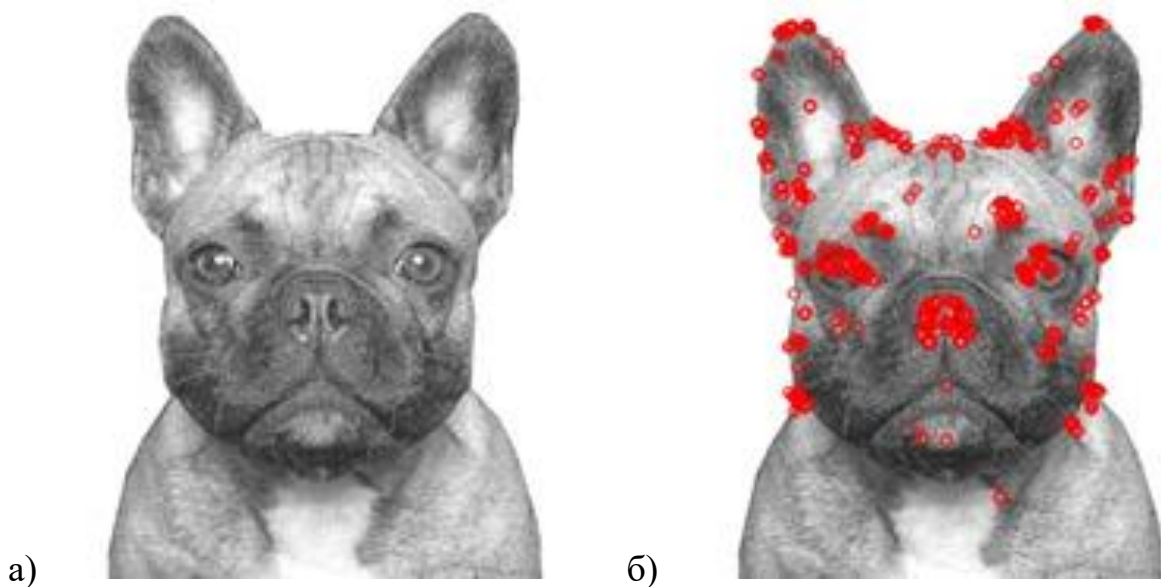


Рисунок 3.7 – Зображення під дією Гауссівського шуму:

а) – вихідний результат; б) – з виділеними дескрипторами КТ

Результати даного експерименту зазначені у таблиці 3.6.

Таблиця 3.6 – Результат експерименту з Гауссівським шумом

Стан зображення	Сума розподілів		
	1	2	3
Без шуму	171,675	165,451	162,87
З Гауссівським шумом	172,740	164,242	163,017

За заданими умовами класифікація пройшла успішно. Побудований класифікатор відповідає необхідному класу. Значення Δ дорівнює 4,9%. При збільшенні кількості шуму до 0,6 результативність значно погіршилася. Число Δ зменшилося до 0,39%, хоча саме зображення віднесено до потрібного класу. Тож, як і у попередньому випадку, необхідно враховувати

пошкодження самого зображення та намагатися покращити якість сторонніми методами.

Окремо варто зазначити зміни у значеннях медіан. Оскільки даний центр класу цілком залежить від усієї множини побудованих дескрипторів, то при її значних змінах значення медіани також трансформується.

У таблицях 3.7 та 3.8 показані різниці між медіаною «чистого» зображення та під дією шумів «солі та перцю» та Гауссівського відповідно, зі збільшенням об'єму шуму.

Таблиця 3.7 – Зміни значення медіани під дією шуму «солі та перцю»

Об'єм шуму	Відстань між медіанами	Відсоток від максимального значення, %
0,01	107	41,8
0,1	113	44,1
0,5	114	44,5
0,8	119	46,5

У випадку, коли зображення пошкоджене шумом «солі та перцю», спостерігалася схожість в отримуваних результатах. Незважаючи на збільшення дії шуму, відстань між медіанами коливалась на інтервалі 41,8%...46,5%.

Таблиця 3.8 – Зміни значення медіани під дією Гауссівського шуму

Об'єм шуму	Відстань між медіанами	Відсоток від максимального значення, %
0,1	29	11,3
0,3	66	25,8
0,6	63	24,6
0,9	93	36,3

Із Гауссівським шумом ситуація склалася по іншому. При збільшенні об'єму шуму, збільшувалась і сама відстань між медіанами «чистого» зображення та пошкодженого.

Отримавши дані результати, можна зробити висновок, що даний алгоритм класифікації та використання медіани як центру класу досить сильно залежить від рівня якості досліджуваних зображень. Необхідна додаткова обробка вхідних даних для отримання більш точного результату.

3.4 Аналіз способів виділення інформативної частини опису

Для покращення результатів класифікації спробуємо здійснити стиснення множини КТ зображення шляхом виділення фіксованої кількості дескрипторів, найближчих до медіани у відповідності до методу (2.12) [30].

Початкова кількість КТ становить 500. Для кожного зображення проведені експерименти, під час яких кількість дескрипторів поступово зменшувалась до 200, 100 та 50. Отримані результати внесені до таблиці 3.9.

Таблиця 3.9 – Вектори сум для різної кількості дескрипторів

КТ	Еталон 1			Еталон 2			Еталон 3		
	1	2	3	1	2	3	1	2	3
500	171,675	165,451	162,87	156,738	172,315	170,947	152,457	173,069	174,474
200	74,340	63,366	62,294	56,109	72,118	71,773	52,813	72,196	74,991
100	39,134	30,942	29,924	27,209	36,81	35,982	26,157	35,739	38,104
50	20,357	15,186	14,457	13,238	18,746	18,016	12,921	17,833	19,246

Розрахувавши Δ , можна побачити, що цей спосіб має право на використання, так як різниця між еталонними зображеннями стає більш

суттєвою при зменшенні числа дескрипторів. Для 50 КТ Δ становить 25,4%, 3,89%, 7,34%, у протидію раніше розрахованих 3,63%, 0,79%, 0,81% для 500. Більш наглядно покращення ілюструється на рисунку 3.8.

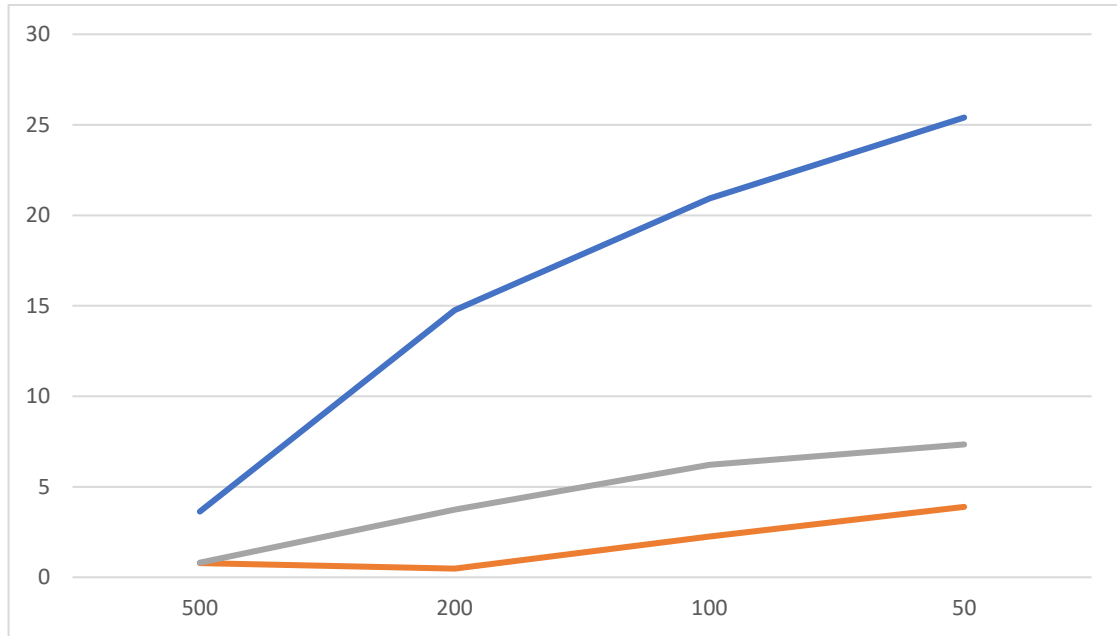


Рисунок 3.8 – Залежність Δ від числа дескрипторів

Подальше зменшення кількості дескрипторів не має сенсу, так як призводить до втрат інформативності вхідних даних. При цьому варто зазначити, що стиснення множини КТ не змінює значення медіани.

Для знаходження найбільш оптимального алгоритму, було проведено ще декілька експериментів. Один із них ґрунтувався на випадковому відборі 100 дескрипторів. Заздалегідь, це ризикований спосіб, так як для одних і тих же об'єктів класифікації кожного разу використовується унікальний ансамбль дескрипторів, що може значно змінити кінцевий результат при повторному використанні алгоритму. Також, для нової множини КТ, необхідно знаходити новий центр класу.

Результати декількох запусків програми показали, що значення відстаней між новими медіанам не покращуються. У середньому вони дорівнюють 20%...40% від максимальної відстані при раніше знайдених 16%...46,5% для 500 КТ. Щодо значень розподілу, то алгоритм взагалі

показав свою непрацездатність. Отримані вектори сум для першого запуску – (33,73; 33,11; 33,16), (31,43; 34,78; 33,79), (31,2; 34,13; 34,67). Числа майже не відрізняються, що показує значну схожість еталонів між собою у даному просторі ознак. Під час повторного використання, ситуація не змінилася.

Ще один перевірений спосіб стиснення полягав у знаходженні 100 дескрипторів за певним алгоритмом відбору, але він, як і попередній, не показав поліпшення результатів, а навіть навпаки, погіршив їх. Значення Δ варіювалося у межах 0%...2%, та і сам результат класифікації не завжди вірний.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи бакалавра було проведено дослідження, під час якого розроблено модель побудови класифікатору на основі інтегрального подання статистичних розподілів компонентів даних. Для реалізації цього методу використаний один із новітніх детекторів ключових точок ORB. Проведено аналіз отриманих результатів та впроваджені спроби по вдосконаленню даного підходу.

Загалом розроблений метод побудови класифікатору показав свою працездатність та отримав право на життя, але все рівно потребує подальшого удосконалення для підвищення якості класифікації під час роботи з більшою базою еталонних зображень, щоб уникнути схожості образів у даному просторі ознак. Навіть при умові використання трьох класів спостерігалась певна подібність між двома зображеннями.

Проведені додаткові дослідження на предмет стійкості даного алгоритму, шляхом імітації дії шумів «солі та перцю» та Гаусса. Виявлено, що на дану модель класифікації та використання медіани як центру класу мають досить сильний відгук на різноманітні пошкодження. Це вирішується попередньою обробкою зображень, але з'являється додаткова відповідальність у користувача алгоритму, яка доволі сильно впливає на сам результат класифікації.

Розглянуто різні варіанти стиснення множини дескрипторів з метою відбору найбільш значущих. Найкращий результат показав метод, у якому виділялася фіксована кількість КТ, найближчих до центру класу (медіани). Перевагою даного способу є те, що сама медіана залишається незмінною.

Варто зазначити, що використання медіани як центру класу має свої перспективи, так як порівняння їх між собою показало, що вони у достатній мірі відрізняються один від одного. Це можна пояснити тим, що вони цілком залежать від усього ансамблю знайдених дескрипторів. Їх використання може значно пришвидшити процес класифікації.

Новизна дослідження складає розроблення ефективного методу класифікації зображення на основі використання статистичного апарату розподілів на множині класів, що сприяє поглибленому аналізу даних та забезпечує результативність класифікації. Зіставлення значень агрегації отриманих розподілів для кожного класу зводить роботу методу до використання кількох простих арифметичних операцій, що значно спрощує саме розуміння даного підходу та дозволяє обробляти повний набір вхідних даних, не втрачаючи інформативність візуальних образів.

Практична значущість роботи полягає у розробленні програмної реалізації моделі класифікації у заданому просторі ознак, підтвердженні працездатності з використанням множини еталонних зображень та перевірці впроваджених способів удосконалення з подальшим порівнянням отриманих результатів.

Дослідження викладені у одній науковій статті, що опублікована у посібнику «Сучасні інформаційні системи» Том 5, № 1 (2021) [17] та у тезах в рамках 25-го Міжнародного молодіжного форуму [30].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Dharampal, V. M. (2015). Methods of image edge detection: A review. *J Electr Electron Syst*, 4(2).
2. Koka, S., Anada, K., Nomaki, K., Sugita, K., Tsuchida, K., & Yaku, T. (2011). Ridge detection with the steepest ascent method. *Procedia Computer Science*, 4, 216-221.
3. Trajković, M., & Hedley, M. (1998). Fast corner detection. *Image and vision computing*, 16(2), 75-87.
4. Grycuk, R., Scherer, R., & Gabryel, M. (2015). New image descriptor from edge detector and blob extractor. *Journal of Applied Mathematics and Computational Mechanics*, 14(4), 31-39.
5. Krig, S. (2016). Interest point detector and feature descriptor survey. In *Computer vision metrics* (pp. 187-246). Springer, Cham.
6. Rodehorst, V., & Koschan, A. (2006, March). Comparison and evaluation of feature point detectors. In *5th International Symposium Turkish-German Joint Geodetic Days*.
7. Tuytelaars, T., & Mikolajczyk, K. (2008). *Local invariant feature detectors: a survey*. Now Publishers Inc.
8. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110.
9. Işık, Ş. (2014). A comparative evaluation of well-known feature detectors and descriptors. *International Journal of Applied Mathematics Electronics and Computers*, 3(1), 1-6.
10. 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.
11. Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010, September). Brief: Binary robust independent elementary features. In *European conference on computer vision* (pp. 778-792). Springer, Berlin, Heidelberg.

12. Краснобаев, Е. А. (2019). Сравнение бинарных дескрипторов особых точек изображений в условиях искажений. Компьютерная оптика, 43(3).

13. Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). Image classification method modification based on model of logic processing of bit description weights vector. Telecommunications and Radio Engineering, 79(1).

14. Гороховатский, В. А., & Передрий, Е. О. (2009). Корреляционные методы распознавания изображений путем голосования систем фрагментов. Радіоелектроніка, інформатика, управління, (1 (20)).

15. Daradkeh, Y. I., Tvoroshenko, I., Gorokhovatskyi, V., Latiff, L. A., & Ahmad, N. (2021). Development of Effective Methods for Structural Image Recognition Using the Principles of Data Granulation and Apparatus of Fuzzy Logic. IEEE Access, 9, 13417-13428.

16. Гороховатський, В. О., Пупченко, Д. В., & Солодченко, К. Г. (2018). Аналіз властивостей, характеристик та результатів застосування новітніх детекторів для визначення особливих точок зображення.

17. Гороховатський, В. О., Гадецька, С. В., Жадан, О. В., & Хвостенко, О. О. (2021). Дослідження результативності класифікаторів зображень за статистичними розподілами для компонентів структурного опису.

18. 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.

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

20. 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).

21. Gorokhovatskyi, V., Gadetska, S., & Ponomarenko, R. (2019, May). Recognition of visual objects based on statistical distributions for blocks of

structural description of image. In International Scientific Conference “Intellectual Systems of Decision Making and Problem of Computational Intelligence” (pp. 501-512). Springer, Cham.

22. Gorokhovatskyi, V. O., & 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).

23. Медиана в статистике. URL: <https://statanaliz.info/statistica/opisanie-dannykh/mediana-v-statistike/> (дата звернення: 01.05.2021).

24. V.A. Gorokhovatskyi. *Structural Analysis and Intellectual Data Processing in Computer Vision*, 2014.

25. Gorokhovatskyi, O., Gorokhovatskyi, V., & Peredrii, O. (2018). Analysis of Application of Cluster Descriptions in Space of Characteristic Image Features. *Data*, 3(4), 52.

26. Гороховатский, В. А., Путятин, Е. П., & Столяров, В. С. (2017). Исследование результативности структурных методов классификации изображений с применением кластерной модели данных. *Радіоелектроніка, інформатика, управління*, (3), 78-85.

27. Gorokhovatskiy, V. A., & Zamula, A. A. (2016). Employment of Intelligent Technologies in Multiparametric Control Systems. *Telecommunications and Radio Engineering*, 75(19).

28. Гороховатский, В. А. (2003). *Распознавание изображений в условиях неполной информации*. Харьков: ХНУРЭ.

29. Selami, A. M. A. A., & Fadhil, A. F. A Study of the Effects of Gaussian Noise on Image.

30. Жадан, О. В. (2021). Метод стиснення множини ключових точок зображення.