

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ МЕТОДУ КЛАСИФІКАЦІЇ ЗА МНОЖИНОЮ
НЕЗАЛЕЖНИХ КЛАСТЕРІВ ДЛЯ ДАНИХ СТРУКТУРНОГО ОПИСУ
ЗОБРАЖЕННЯ
(тема)

Виконав:
студент 2 курсу, групи ІНФМ-19-2

Запорожченко А.П.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

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)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 20__ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУстудентові Запорожченку Антону Петровичу
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження методу класифікації за множиною незалежних кластерів для даних структурного опису зображення

затверджена наказом по університету від «23» жовтня 2020 року № 1428Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 30 листопада 2020 р.3. Вихідні дані до роботи Алгоритми пошуку характерних ознак зображень, теоретичні відомості про методи класифікації зображень, тестові зображення, перелік використовуваних програмних засобів: Java, OpenCV, середовище розробки IntelliJ IDEA 2020

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Огляд методів пошуку характерних ознак на зображенні та аналіз дескрипторів2. Математична модель класифікації зображень за множиною незалежних кластерів3. Параметри підрахунку відстані при співставленні центрів кластерів4. Комп'ютерне моделювання алгоритмів кластеризації та пошуку відповідного еталону5. Аналіз отриманих результатів.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Актуальність проблеми кластеризації зображень, постановка задачі, тестові зображення, математична модель структурного опису зображень, алгоритм кластеризації k -середніх, стратегія голосування за еталони, угорський метод, результати експериментів, висновки

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

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	23.10.2020	
2	Аналіз завдання, підбір літератури	24.10.20-27.10.20	
3	Аналіз літератури з досліджуваної проблеми	27.10.20-02.11.20	
4	Аналіз технічних засобів	03.11.20-04.11.20	
5	Розробка методу	04.11.20-10.11.20	
6	Програмна реалізація	10.11.20-17.11.20	
7	Оформлення пояснювальної записки	17.11.20-29.11.20	
8	Перевірка на плагіат	30.11.20	
9	Рецензування	01.12.20	
10	Підготовка презентації та доповіді	02.12.20	
11	Занесення роботи в електронний архів	09.12.20	
12	Попередній захист атестаційної роботи	09.12.20	

Дата видачі завдання 23 жовтня 2020 р.

Студент _____
(підпис)

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

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до атестаційної роботи: 62 с., 9 табл., 13 рис., 2 дод., 44 джерела.

КОМП'ЮТЕРНИЙ ЗІР, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, КЛЮЧОВА ТОЧКА, КЛАСТЕРИЗАЦІЯ, АЛГОРИТМ K-СЕРЕДНІХ, СТРУКТУРНИЙ ОПИС, ДЕТЕКТОР ORB, ДЕСКРИПТОР.

Дослідження присвячено розробленню ефективних методів класифікації зображень на підставі даних структурного опису як множини дескрипторів ключових точок.

Об'єктом дослідження є метод класифікації зображень з поданням опису як множини кластерів.

Розроблено метод класифікації з використанням попередньої обробки даних опису шляхом кластеризації у просторі ознак ключових точок із застосуванням дескриптора ORB, здійснене порівняння працездатності та ефективності класифікації з використанням моделей голосування та угорського методу. Розвивається ідея стиснення подання зображень у вигляді множин характерних ознак за допомогою кластеризації задля посилення показників результативності (точність, швидкодія класифікації).

Розроблена програмна модель класифікації зображень у базі еталонів методом голосуванням та угорським методом.

COMPUTER VISION, IMAGE CLASSIFICATION, KEY POINT, CLUSTERING, K-MEANS ALGORITHM, STRUCTURAL DESCRIPTION, ORB DETECTOR, DESCRIPTOR.

The aim of the research is to enhance the existing methods of image classification based on their structural description.

The object of the research is the classification of images that are represented in a space of features.

Methods of classification of image samples with preliminary processing and clustering in the feature space are developed. ORB descriptors are defined and used. The comparison of correctness and efficiency of classification with use of voting models and Hungarian method is carried out. The research advocates the idea of compressing of the representation of image as a feature set by means of clustering to achieve its optimal practical use.

As a result, the software that classifies sample images based on the predefined database with voting and Hungarian algorithm is implemented.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ.....	7
1 Аналіз методів формування ключових точок зображень	9
1.1 Основні поняття розпізнавання за множиною КТ.....	9
1.2 Детектори кутів	11
1.3 Формування та опис КТ	13
1.4 Сучасні методи визначення та опису КТ	18
1.5 Постановка завдання дослідження.....	24
2 Класифікація зображень за множиною незалежних кластерів	25
2.1 Формалізація задачі структурної класифікації	25
2.2 Використання угорського методу	29
2.3 Кластеризація у просторі дескрипторів КТ.....	37
3 Результати дослідження на підставі комп'ютерного моделювання	45
3.1 Обґрунтування вибору середовища програмної реалізації	45
3.2 Інструкція користувача	45
3.3 Результати моделювання та експериментального дослідження	46
Висновки	54
Перелік джерел посилання	55
Додаток А Тестові зображення – еталони	60
Додаток Б Тестові зображення – зразки	61

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

КТ – ключова точка

ЛГ – лапласіан гаусіану

УГ – угорський метод

ХО – характерна ознака

BRIEF – Binary Robust Independent Elementary Features

FAST – Features from Accelerated Segment Test

ORB – Oriented FAST and Rotated BRIEF

SIFT – Scale-Invariant Feature Transform

SURF – Speeded-Up Robust Features

ВСТУП

Розвиток сучасних систем комп'ютерного зору безпосередньо пов'язаний як зі створенням нових моделей ознак для представлення зображень, так і з синтезом новітніх обчислювальних методів класифікації, що базуються на таких моделях. Фундаментом розпізнавання зображень об'єктів, що описуються множинами ключових точок (КТ), є метричні відносини, що відображають подібність як окремих КТ, так і візуальних об'єктів в цілому, і фактично встановлюють деяку топологію уявлень візуальних образів у просторі ознак [1-4]. Відповідно до структурного підходу, з КТ як з окремих цеглинок збираються образи об'єктів, що становлять сцену зображення. Переміщення сформованих КТ в полі зору адекватно відображає фізичний рух об'єкта. Топологія множини дескрипторів КТ суттєво відображає і фізичну топологію розпізнаваних об'єктів у побудованому просторі образів [3].

Перспективною в плані обсягу обчислювальних витрат для класифікації є ідея побудови деякої множини опорних точок у просторі КТ, які формують довільний структурний опис [5-8]. Комбінацією кількостей ХО, близьких до опорних точок у просторі їх значень (дескрипторів КТ), можна універсально отримати опис довільного образу об'єкта. Схожі можливості забезпечує апарат мультимножин.

Розвитком ідеї кластерного уявлення для структурних описів прикладної бази зображень є безпосередня побудова правила класифікації, яке за значенням дескриптора КТ відносить її до фіксованого класу, в результаті аналізу накопичується певна кількість голосів, за якими згодом визначається клас для підслідного об'єкта. Для близьких за змістом зображень таке правило із заданою ефективністю важко реалізувати. Однак дослідження показують, що для різноманітності баз зображень синтез класифікаційного правила може дати непогані результати [7]. Викликає інтерес також застосування моделі обчислення подібності описів на основі

принципу формування оптимальних групових відповідностей для визначення максимальної кількості пар ознак об'єкт-еталон у вигляді інваріантних просторових структур на основі угорського методу, яка сприяє підвищенню достовірності розпізнавання за рахунок однозначного і більш точного обліку схожості підмножин описів і забезпечує мінімум критерію сумарного відстані між описами об'єкта й еталона.

Мета роботи – побудова методу класифікації зображень на основі голосування дескрипторів ключових точок з використанням апарату бітової обробки, а також порівняння цього методу з іншим ефективним способом структурного аналізу даних – угорським методом, що інакше називається комбінаторною оптимізацією обмежених множин.

У рамках комп'ютерного моделювання створена програма, що реалізує алгоритм кластеризації k -середніх на просторі характерних ознак зображень, проводить голосування простим підрахунком голосів, а також реалізує пошук відповідного еталону угорським методом. Програма має ряд параметрів для управління, серед яких, наприклад, алгоритм пошуку дескрипторів та кількість КТ для пошуку на зображенні.

1 АНАЛІЗ МЕТОДІВ ФОРМУВАННЯ КЛЮЧОВИХ ТОЧОК ЗОБРАЖЕНЬ

1.1 Основні поняття розпізнавання за множиною КТ

Задача розпізнавання образів – це віднесення вихідних даних до певного класу за допомогою виділення істотних ознак, що характеризують ці дані, із загальної маси несуттєвих даних [1]. Для її розв’язку корисним буде розглянути, як ці істотні ознаки виділяє людина під час безперервного процесу аналізу візуальної інформації. Виявляється, що задля запам’ятовування та упізнання візуальних даних людина підсвідомо шукає унікальні шаблони, або ознаки, які легко можуть бути відстежені та порівняні. Ця здатність невід’ємна від процесу людського пізнання, проте для дослідників та науковців завжди було цікаво, чи можна навчити комп’ютер упізнавати об’єкти за тим же принципом.

На рисунку 1.1 зображений прямокутник з відміченими різними кольорами ділянками:

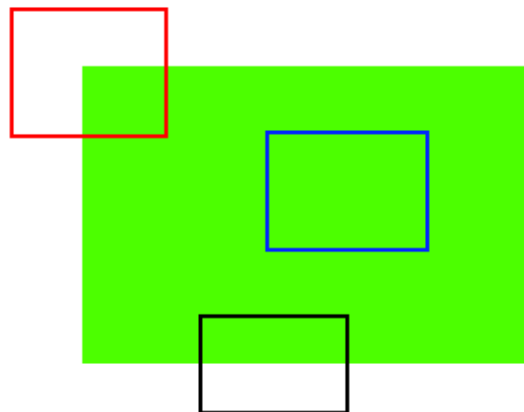


Рисунок 1.1 – Картинка з відміченими ділянками

Ділянку з синім прямокутником важко розпізнати, бо схожих, і навіть ідентичних ділянок багато. Чорний прямокутник – край – знайти легше,

проти визначити його точне розташування все ще не вдасться. Червоний прямокутник – це кут. На даному зображенні такий кут тільки один, тому з визначенням його розташування не буде проблем. Це дає змогу припустити, що кути на зображенні складають «мапу», завдяки якій це зображення можна ідентифікувати. І це припущення виявилось правильним [9].

Для пошуку кутів на зображенні використовуються алгоритми, що мають назву детектори [10, 11]. У їх основу покладена проста ідея, яка полягає у тому, що не уся інформація, закодована у зображенні, має однакову цінність задля його розпізнавання. Результатом роботи детекторів є набір ділянок або навіть точок, які найбільш точно характеризують вихідне зображення, але для аналізу яких знадобиться набагато менше обчислювальних ресурсів. Такі ділянки називаються характерні ознаки, або ключові точки чи точки інтересу. Ці поняття будемо вважати еквівалентними.

Для подальшої роботи з КТ їх потрібно описати, так же, як людина би описала ділянки зображення, проте на мові комп'ютерів. Для цього існує перелік алгоритмів-дескрипторів, більшість з яких представляють КТ у формі числового або бінарного вектору [12]. Такий варіант виявився найбільш прийнятним задля співставлення чи будь-якої іншої машинної обробки.

Останнім та найбільш цікавим етапом у розпізнаванні є співставлення КТ. Так як КТ зазвичай описуються як числові вектори, моделі їх співставлення базуються на порівнянні у лінійному нормованому просторів із застосуванням функцій схожості або метрик [13]. Детальний опис найбільш поширених методів наведений у другому розділі.

Бажаною властивістю системи розпізнавання була б інваріантність до геометричних перетворень та зміни масштабу. Відображення $I:W \rightarrow \Lambda$ вважається інваріантним відносно групи G , якщо виконується

$$I(gB) = I(B) \forall g \in G, \forall B \in W.$$

На практиці це означає, що після застосування будь-яких перетворень групи G система повинна виявити той же самий набір КТ, який описує нормоване, тобто вихідне, зображення. Одним зі шляхів досягнення інваріантності є вживання кореляційних принципів, які базуються на аналізі усіх можливих значень параметру (наприклад, усіх можливих зміщень чи масштабів). Детально механізми забезпечення інваріантності розглядаються разом з дескрипторами КТ.

1.2 Детектори кутів

У пошуку найбільш інформативних ділянок зображень велику роль відіграють кути. Кути самі по собі вважаються ключовими ділянками, тобто дозволяють унікально ідентифікувати зображення. Розглянемо декілька методів пошуку кутів на зображенні.

Відомо, що кути – це ділянки з максимальною різницею в інтенсивності у будь-якому напрямку. Цю ідею використали Кріс Гарріс та Майк Стефенс у своїй роботі [10]. В результаті вони вивели функцію (1.1), максимізація якої дозволить визначити кути на зображенні:

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x, y)]^2. \quad (1.1)$$

Перший множник – це прямокутна або гаусова віконна функція, яка надає пікселям вагу. Другий – різниця між інтенсивностями точки та її зміщення. Саме цю різницю треба максимізувати. Після розкладу в ряд Тейлора та деяких математичних перетворень отримуємо остаточне рівняння.

$$E(u, v) \approx [u \quad v] M \begin{bmatrix} u \\ v \end{bmatrix},$$

де

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}.$$

I_x та I_y – похідні зображення за змінними x та y відповідно. Після цього треба порахувати показник (1.2), який продемонструє, є у вікні кут чи немає:

$$R = \det(M) - k(\text{trace}(M))^2, \quad (1.2)$$

де $\det(M) = \lambda_1 \lambda_2$;

$\text{trace}(M) = \lambda_1 + \lambda_2$;

λ_1 та λ_2 – власні числа M .

Значення цих власних чисел показують, чи є ділянка кутом, краєм або рівною площиною:

- якщо значення $|R|$ невелике, ділянка рівна;
- якщо $R < 0$, що трапляється коли $\lambda_1 \gg \lambda_2$, ділянка є краєм;
- якщо значення R велике, ділянка є кутом.

У 1994 році Д. Ші та К. Томаші запропонували модифікацію детектора Гарріса та описали її у роботі [11]. Вони замінили функцію оцінки (1.2) на

$$R = \min(\lambda_1, \lambda_2).$$

Тут кутом вважається ділянка, для якої значення R більше деякого порогу. Якщо побудувати залежність $\lambda_1 - \lambda_2$, вона буде мати вигляд як на рисунку 1.2, зображеному нижче.

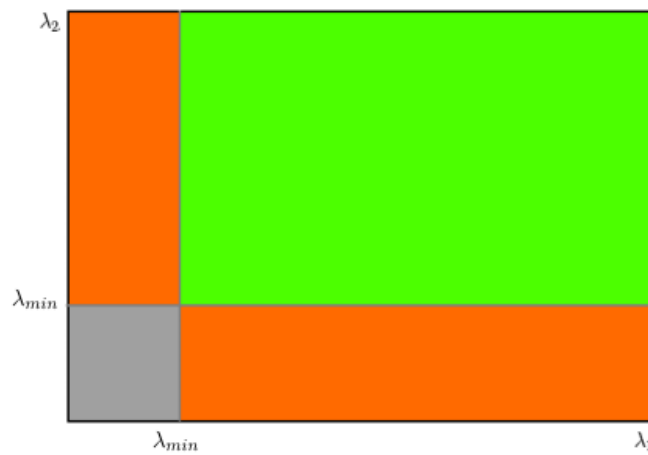


Рисунок 1.2 – Залежність $\lambda_1 - \lambda_2$ у двовимірному просторі

З цього видно, що тільки коли λ_1 та λ_2 більші за λ_{min} , ділянка вважається кутом.

1.3 Формування та опис КТ

Розглянуті методи пошуку кутів інваріантні до поворотів, проте через фіксований розмір вікна вони не здатні правильно розпізнавати кути зі зміною масштабу. Для прикладу, детектор Гарріса розпізнає кут на лівій частині рисунку 1.3, але не знайде його на правій частині, якщо використовувати вікно того ж розміру.

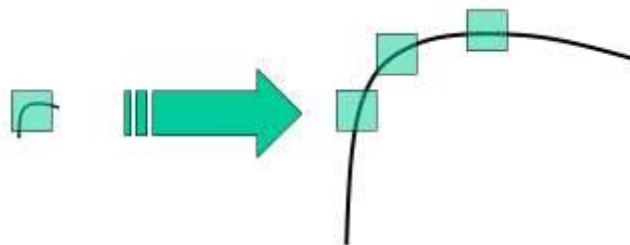


Рисунок 1.3 – Проблема знаходження кутів з різними масштабами

Саме тому у 2004 році Девід Лоу розробив новий алгоритм пошуку ключових точок, який назвав SIFT (Scale-Invariant Feature Transform) [14]. Цей алгоритм складається з чотирьох кроків, описи яких наведені далі.

1. Пошук екстремумів у масштабованому просторі. Для забезпечення інваріантності до зміни масштабу потрібно використовувати вікна змінного розміру. Для цього застосовують фільтрацію у масштабованому просторі. Для зображення рахують лапласіан гаусіану з різними значеннями σ . Саме зміна σ дозволяє маніпулювати розміром вікна під час пошуку кутів. Далі змінюючи площину та масштаб шукають локальні максимуми, тобто комбінації (x, y, σ) , які можуть вказувати на потенційну КТ.

Операція підрахунку ЛГ досить затратна, тому в алгоритмі SIFT використовується різниця гаусіанів, яка є апроксимацією ЛГ. Різниця гаусіанів рахується як різниця гаусового розмиття зображення з різним коефіцієнтом масштабування σ , наприклад σ_1 та $k\sigma_1$. Ця різниця шукається для різних октав зображення у гаусовій піраміді. Далі серед отриманих зображень шукається локальний екстремум, який обирається як потенційна КТ. Цей процес зображений на рисунку 1.4.

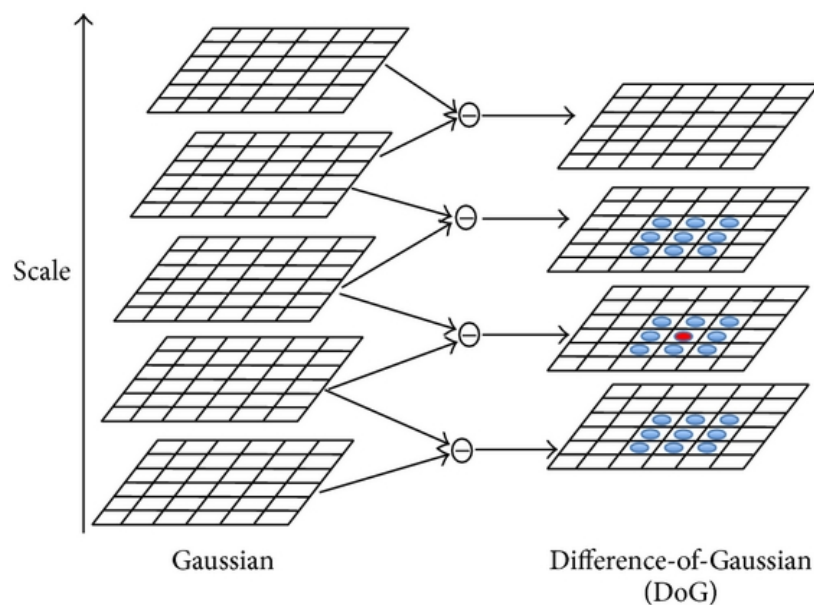


Рисунок 1.4 – Формування різниці гаусіанів та пошук екстремуму

2. Локалізація КТ. Як тільки знайдені потенційні КТ, їх положення треба уточнити для більш точних результатів. В алгоритмі SIFT для локалізації екстремума масштабований простір розкладають у ряд Тейлора. Якщо інтенсивність менша за деякий поріг (0,03 у джерелі), екстремум відкидається. Різниця гаусіанів добре визначає не тільки кути, а й краї. Для відкидання країв автори використали матрицю Гессе H розміром 2×2 , завдяки якій обчислили головну кривизну:

$$H = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix}.$$

З алгоритму пошуку кутів Гарріса ми знаємо, що для краю одне власне значення більше іншого. Нехай менше значення β відноситься до α як $\alpha = r\beta$. Тоді КТ приймається тоді і тільки тоді, коли виконується

$$\frac{\text{trace}(H)^2}{\det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} < \frac{(r+1)^2}{r},$$

де $r = 10$ у [10].

3. Визначення орієнтації. Щоб досягти інваріантності щодо повороту зображення, потрібно додати до кожної обраної КТ орієнтацію. У залежності від масштабу для КТ обирається оточення у 4 піксела, далі у цій ділянці рахується величина градієнту та напрямок. Формується гістограма орієнтації з 36 групами, які покривають 360 градусів. Кожне значення зважене за величиною градієнта та за гаусовим циркулярним вікном з σ , що в 1,5 рази більше за масштаб КТ. Обирається найвищий пік гістограми, а також усі інші, розмір яких не менш 80% від нього. Далі для кожного піка створюються КТ з однаковими координатами та масштабом, але різними напрямками. Це сприяє більш точному співставленню.

4. Опис КТ. Для новостворених КТ обирається ділянка навколо розміром 16×16 . Вона поділяється на 16 підділянок 4×4 . Для кожного такого блоку формується гістограма орієнтації з 8 групами, тобто усього 128 груп. Вони представляються як вектор – дескриптор КТ. Додатково проводяться деякі виміри для досягнення стійкості до змін освітленості, поворотів і т.д.

Алгоритм SIFT виявився надто повільним, тому виникла необхідність більш швидкої версії. У 2006 році була опублікована робота [15], яка і описувала пришвидшену версію SIFT.

У SIFT ЛГ апроксимується як різниця гаусіанів. SURF (Speeded-Up Robust Features) [15] йде далі та апроксимує ЛГ за допомогою розмиття за рамкою. Рисунок 1.5 демонструє роботу цього розмиття. У такого підходу є велика перевага: згортка із таким розмиттям може бути легко розрахована із застосуванням інтегрального образу, паралельно для різних масштабів. Також SURF використовує детермінант матриці Гессе і для масштабу, і для координат.

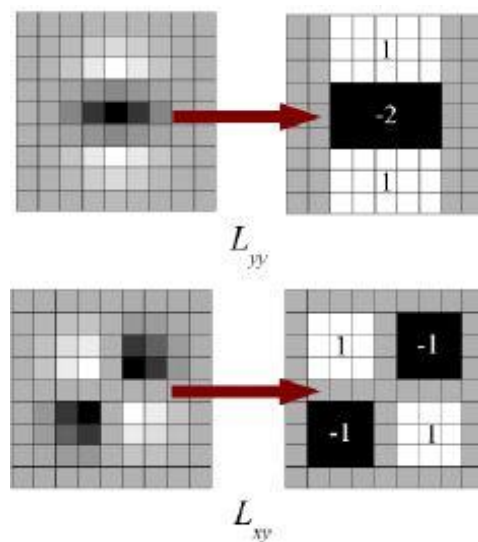


Рисунок 1.5 – Апроксимація часткових гаусових похідних 2-го порядку із застосуванням розмиття за рамкою

Для визначення орієнтації у SURF застосовуються вейвлет-характеристики у горизонтальному і вертикальному напрямку для оточення з

6 пікселів. Застосовуються також відповідні гаусові ваги. Далі вони вирисовуються на площині так, як показано на рисунку 1.6. Домінуюча орієнтація визначається після підрахунку суми усіх характеристик у діапазоні змінного вікна орієнтації з кутом 60 градусів. Цікаво те, що вейвлет-характеристики можуть дуже легко бути підраховані для різного масштабу із застосуванням інтегральних образів. Для багатьох застосувань інваріантність до поворотів не потрібна, тому у пошуку орієнтації немає потреби і процес пришвидшується. SURF має модифікацію Upright-SURF (U-SURF), що не шукає орієнтацію. Вона швидша та надійна з відхиленням до $\pm 15^\circ$.

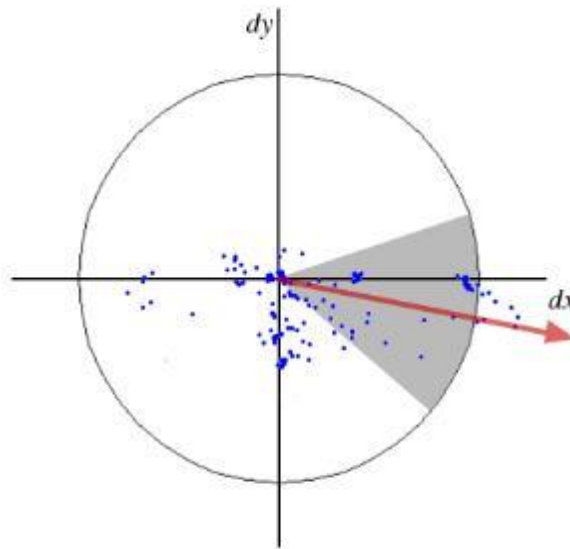


Рисунок 1.6 – Вейвлет-характеристики використовуються для визначення орієнтації

Для опису КТ SURF використовує вейвлет-характеристики у горизонтальному і вертикальному напрямку (тут знову на допомогу приходять інтегральні образи). Навколо КТ обирається ділянка розміром $20s \times 20s$, де s – розмір. Вона поділяється на підділянки 4×4 . Для кожної рахуються вертикальні та горизонтальні вейвлет-характеристики. Тоді дескриптор формується як вектор виду

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|).$$

Стандартний дескриптор має розмірність 64, але є й подовжена версія розмірністю 128. Суми d_x та $|d_x|$ рахуються окремо для $d_y < 0$ та $d_y \geq 0$. Так само суми для d_y та $|d_y|$ розділяються відповідно до знаку d_x , подвоюючи кількість ознак. Це не додає багато складності до обчислень, проте додає ознакам виразність.

Ще одне істотне покращення полягає у використанні знаку лапсіана (з матриці Гессе). Його значення рахується під час визначення точок інтересу. Знак дозволяє відділяти яскраві ділянки на темному фоні від протилежної ситуації. На етапі співставлення ми порівнюємо КТ, тільки якщо їх типи контрасту співпадають. Ця мінімальна одиниця інформації дозволить пришвидшити співставлення без втрат у виразності дескриптивів.

Як висновок, особливості SURF дозволяють пришвидшити процес на кожному шагу. Порівняння із SIFT демонструє пришвидшення у 3 рази зі співставною ефективністю. SURF добре справляється з розмитими або повернутими зображеннями, але не дуже добре розпізнає при зміні точки зору або освітлення.

1.4 Сучасні методи визначення та опису КТ

Описані детектори особливих точок працюють добре, проте з точки зору застосунку, що працює у режимі реального часу, їх швидкість незадовільна. Прикладом такого застосунку є задача SLAM – одночасна локалізація і картографування – де обчислювальні ресурси зазвичай обмежені. Як рішення цієї проблеми Едвардом Ростеном та Томом Драммондом був запропонований алгоритм FAST (Features from Accelerated Segment Test) [16]. Наведемо принцип роботи цього алгоритму.

Спочатку на зображенні обирається піксел p з інтенсивністю I_p – він буде ідентифікуватись як точка інтересу. Обирається також поріг t .

Далі беруться 16 пікселів навколо підслідного піксела, як показано на рисунку 1.7.

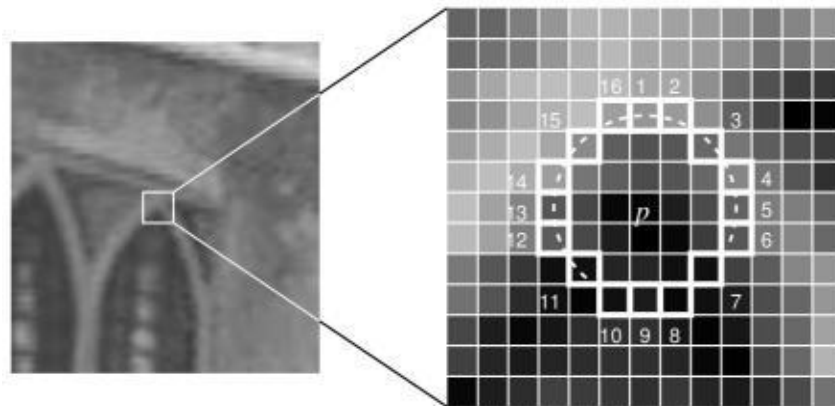


Рисунок 1.7 – Приклад знаходження пікселів-сусідів в алгоритмі FAST

Піксел p вважається кутом, якщо є n послідовних пікселів у колі з 16 пікселів, яскравість яких більша за $I_p + t$ або менша за $I_p - t$. Нехай n буде 12. Для виключення великої кількості точок, що не є кутами, застосовується високошвидкісний тест. Цей тест перевіряє лише 4 піксела – перший, дев'ятий, п'ятий та тринадцятий (спочатку перевіряються 1 та 9, далі 5 і 13). Якщо p – кут, тоді хоча б 3 з цих пікселів повинні бути яскравіші за $I_p + t$ або темніші за $I_p - t$. У іншому випадку p не може бути кутом. Перевірка усіх 16 пікселів може бути застосована у разі успішного проходження швидкісного тесту. Цей детектор забезпечує високу ефективність, проте має ряд недоліків:

- він не відкидає достатню кількість кандидатів для $n < 12$;
- вибір пікселів не оптимальний, бо залежить від порядку перевірки та розподілу появи кутів;
- результати високошвидкісного тесту не зберігаються;
- знаходяться декілька особливих точок, що прилягають одна до одної.

FAST зарекомендував себе як детектор кутів, що у декілька разів швидший за інші. Але він не дуже стійкий до шуму високого рівня. До того ж він дуже залежний від порогу.

SIFT використовує дескриптори розмірністю 128. Так як вони складаються з чисел з рухомою комою, один дескриптор займає 512 байтів. Для дескрипторів SURF потребується не менше ніж 256 байтів (розмірність 64). Для векторів з тисяч таких дескрипторів знадобиться багато пам'яті, що не є прийнятним для застосувань з обмеженими ресурсами, наприклад для вбудованих систем. Чим більше пам'яті зайнято, тим повільніше буде проходити процес співставлення.

Але не усі розмірності будуть потрібні для самого співставлення. Їх можна стиснути такими методами, як PCA або LDA. Для дескрипторів SIFT інколи навіть застосовують LSH (Locality Sensitive hashing), який конвертує числа з рухомою комою у бінарні строки. Ці бінарні строки можуть бути порівняні із використанням відстані Хемінга. Це дає перевагу у швидкості, бо знаходження відстані Хемінга полягає у операції XOR та підрахунку бітів. На сучасних CPU з інструкціями SSE (Streaming SIMD Extensions) ці обчислення проводяться дуже швидко. Але перед хешуванням потрібно знайти дескриптори, тому проблема з пам'яттю ще не вирішена.

Алгоритм BRIEF (Binary Robust Independent Elementary Features) стає у нагоді – він дозволяє знаходити бінарні строки без знаходження дескрипторів [17]. Він бере згладжену ділянку зображення і обирає n_d унікальних пар координат (x, y) . Далі серед деяких пар проводяться порівняння яскравості. Нехай є пари координат p та q . Якщо $I(p) < I(q)$, результат 1, інакше 0. Це застосовується до усіх пар, щоб отримати бітову строку розміром $n_d \cdot n_d$ може бути 128, 256 або 512. В залежності від цього довжина дескриптора буде 16, 32 або 64 байти. Далі для співставлення рахують відстань Геммінга.

BRIEF – це алгоритм для опису КТ. Для знаходження координат КТ можна обрати SIFT, SURF або CenSurE. Останній рекомендований авторами

[17], бо з ним BRIEF працює дещо краще. Підсумовуючи, BRIEF достатньо швидкий метод для опису та співставлення КТ з високим рівнем розпізнавання.

ORB (Oriented FAST and Rotated BRIEF) був описаний чотирма вченими у їх роботі [18]. ORB – гарна альтернатива SIFT та SURF у швидкості обчислення та використанні ресурсів, та на відміну від наведених алгоритмів він не запатентований, тому його використання не тягне додаткових витрат.

ORB – це поєднання детектора FAST та дескриптора BRIEF з багатьма модифікаціями для збільшення ефективності. Спочатку він використовує FAST для знаходження КТ, далі за допомогою міри кутів Гарріса знаходить топ N точок серед них. Використання піраміди дозволяє шукати КТ у різних масштабах. Єдина проблема полягає у тому, що FAST не визначає орієнтацію. Автори знайшли метод, як можна зробити це та зберегти інваріантність до поворотів.

Алгоритм бере ділянку з центром у знайденому куті та обчислює центроїд зважений за інтенсивністю. Напрямок вектору з кутової точки до центроїду задає орієнтацію. Для покращення рівня інваріантності до поворотів обчислюються моменти за x та y , які повинні знаходитись всередині кола з радіусом r , де r – розмір ділянки.

Для опису дескрипторів ORB використовує BRIEF. Проте цей алгоритм не дуже добре працює з повернутими зображеннями, тому ORB «направляє» BRIEF згідно з орієнтацією КТ. Для будь-якої множини КТ з n бінарних тестів у локації (x_i, y_i) будується матриця S розмірністю $2 \times n$, яка містить координати цих пікселів. Далі з використанням орієнтації ділянки – θ – знаходиться матриця повороту та обернена версія $S - S_\theta$.

ORB дискретизує кути до множників $2\pi/30$ та створює пошукову таблицю з підрахованими значеннями BRIEF. Якщо орієнтація КТ θ

зберігається між образами зображення, для обчислення дескриптору буде використовуватись коректна множина точок S_θ .

BRIEF має важливу особливість: кожний дескриптор має велику дисперсію та середнє значення близько 0,5. Але якщо він направлений згідно з орієнтацією КТ, він втрачає цю особливість і стає більш розподіленим. Велика дисперсія робить дескриптори більш виразними. Ще одна бажана особливість – позбутися кореляції між тестами, так, щоб кожний тест впливав на результат. Для вирішення цих питань ORB запускає жадібний пошук серед усіх можливих бінарних тестів та шукає ті, в яких найбільша дисперсія та середнє значення близько 0,5, а також відсутня кореляція. Результат називається rBRIEF.

У роботі [19] говориться, що ORB набагато швидший за SURF або SIFT, а дескриптори ORB працюють краще.

Дескриптор BRISK відрізняється від BRIEF та ORB у тому, що ділянка навколо КТ обирається згідно із заздалегідь визначеним шаблоном [19]. Шаблон BRISK складається з концентричних кіл, як показано на рисунку 1.8.

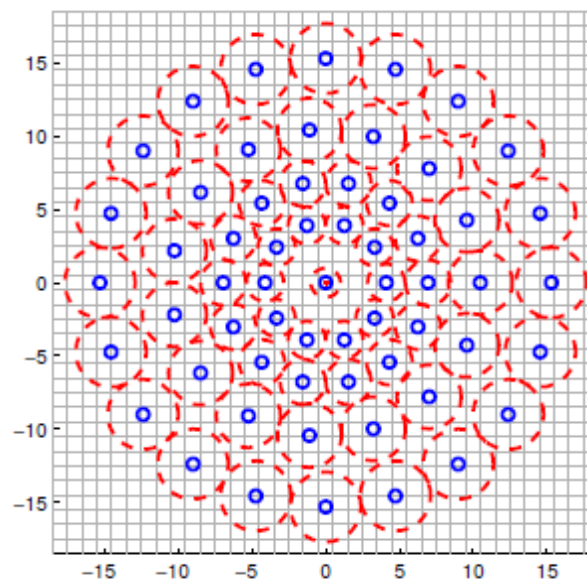


Рисунок 1.8 – Шаблон знаходження точок у ділянці навколо КТ

При розгляді точки обирається невелика ділянка навколо, до якої застосовується згладжування за Гаусом. Червоне коло на рисунку 1.8 ілюструє розмір середнього квадратичного відхилення гаусового фільтра, застосованого до точки інтересу.

З таким шаблоном розрізняють короткі та довгі пари. Короткі пари – це такі пари піддослідних точок, відстань між якими менш, ніж деякий поріг d_{\max} , а довгі пари – це такі, відстань між якими більша іншого порогу d_{\min} , $d_{\min} > d_{\max}$. Довгі пари у BRISK потрібні для визначення орієнтації, а короткі застосовують у порівняння інтенсивності при формуванні дескриптора, так же, як у BRIEF та ORB.

Для обчислення орієнтації КТ BRISK використовує локальні градієнти між піддослідними парами, які визначаються як

$$g(p_i, p_j) = (p_j - p_i) * \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{p_j - p_i^2},$$

де $g(p_i, p_j)$ – локальний градієнт між парами p_i та p_j ;

I – згладжена інтенсивність за Гаусом у відповідній точці інтересу з відповідним середнім квадратичним відхиленням.

Для обчислення орієнтації підсумовуються усі локальні градієнти між усіма довгими парами та рахується $\arctg(g_y / g_x)$, де g_y та g_x – компоненти градієнта. Це дає кут КТ. Тепер треба обернути короткі пари на цей кут щоб зробити дескриптор більш інваріантним до поворотів. BRISK використовує лише довгі пари спираючись на припущення, що локальні градієнти нівелюють один одного та не будуть впливати на визначення глобального градієнту.

BRIEF демонструє кращі результати за BRISK (та ORB) при фотометричних змінах – розмитті, змінах освітлення та JPEG-стисканні,

проте BRISK виграє у BRIEF при зміні точки зору. Загальна продуктивність методу співставна з ORB [19].

1.5 Постановка завдання дослідження

Метою даної роботи є розроблення та дослідження ефективних методів класифікації зображень на підставі даних структурного опису як множини дескрипторів ключових точок із використанням кластеризації на просторі характерних ознак.

Об'єктом дослідження є метод класифікації зображень з поданням опису як множини кластерів.

Для досягнення мети роботи необхідно виконати наступні завдання:

- розробити математичну модель класифікації зображень за множиною незалежних кластерів;
- програмно реалізувати та порівняти алгоритми класифікації зображень на основі голосування та угорського методу із використанням кластеризації k -середніх у фіксованій базі еталонів;
- дослідити вплив параметрів підрахунку відстані (наприклад, кількість КТ) при співставленні центрів кластерів;
- провести аналіз результатів.

Як тестовий матеріал у роботі обрані кольорові зображення істот з уявного світу покемонів, розміром 476×476 пікселів [20]. Приклади тестових зображень наведено в додатках А, Б.

2 КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ ЗА МНОЖИНОЮ НЕЗАЛЕЖНИХ КЛАСТЕРІВ

2.1 Формалізація задачі структурної класифікації

Сформулюємо математичну постановку задачі класифікації візуального об'єкта, представленого описом в просторі дескрипторів КТ. Маємо два стрижневих простори даних: простір сигналів, в якому формуються розпізнаються образи – зображення об'єктів, і простір ознак, де реалізується прийняття рішення про клас образів. Класи об'єктів задані в просторі сигналів, а визначаються і розрізняються в просторі ознак.

Нехай Ω – деякий простір образів (зображень); $\omega \in \Omega$ – образ (візуальний об'єкт, зображення); $M[J] = \{1, 2, \dots, J\}$ – множина номерів для класів образів $\Omega_1, \Omega_2, \dots, \Omega_J$ таких, що $\Omega = \cup_{i=1}^J \Omega_i$, $\Omega_i \cap \Omega_k = \emptyset$, \emptyset – порожня множина; $g: \Omega \rightarrow M$ – невідома індикаторна функція, яка ставить у відповідність кожному образу $\omega \in \Omega$ деякий номер $j \in M$ його класу.

Розглянемо тепер $Z = \{z / z \in R^n\}$ – простір описів образів, де опис – це кінцева множина числових векторів (дескрипторів КТ). Задамо відображення $\Theta: \Omega \rightarrow Z$ з простору Ω в простір Z . Кожному $\omega \in \Omega$ відображення Θ ставить у відповідність його опис $z(\omega) = \{z_i\}_{i=1}^s$. Опис $z(\omega)$ в просторі Z – це кінцева множина: $z(\omega) = \{z_i\}_{i=1}^s$, $s = \text{card } z(\omega)$ – потужність $z(\omega)$; Z_1, Z_2, \dots, Z_J – підмножини в просторі Z , такі, що $Z_i \cap Z_k = \emptyset$ і $Z = \cup_{i=1}^J Z_i$. Передбачається, що ці підмножини отримані дією відображення $\Theta: \Omega_k \rightarrow Z_k$.

Тепер розглянемо $g: Z \rightarrow M$ – вирішальне правило, яке множині $z(\omega)$ ознак образу ставить у відповідність номер класу образу. Процес класифікації – це ланцюжок перетворень даних з простору образів у множину класів, має вигляд

$$\Omega \rightarrow \Theta(\Omega) \rightarrow Z \rightarrow g(Z) \rightarrow M. \quad (2.1)$$

Головним завданням процесу (2.1) є побудова правила g , базуючись на представленні в просторі Z , отриманому застосуванням Θ . Вибір Z і $z(\omega)$ прямо впливає на ефективність правила g . При цьому критерієм якості класифікації виступає результативність (правильність) прийняття рішення щодо елементів вихідного простору образів Ω .

У задачі розпізнавання, де описи представлені сукупностями КТ зображення, множини описів $z(\omega)$ формуються детекторами побудови КТ: SIFT, SURF, ORB, BRISK і ін. [12, 14-19]. Для методу SURF, наприклад, $z(\omega) = \{z_i / z_i \in R^{64}\}_{i=1}^s$. В окремому випадку множини $z(\omega)$ представимо у вигляді єдиного вектора, тоді Z трактується як простір векторів [21, 22].

Обмеження традиційної постановки не враховують, що образи у просторі Ω , як і їх описи у просторі Z , можуть мати непорожній перетин. Цей нетривіальний аналіз забезпечують сучасні методи, засновані, зокрема, на імовірнісному або нечіткому представленнях [23, 24].

На попередньому етапі здійснюємо формування бази еталонних образів і побудова для кожного з них структурних описів Z_1, Z_2, \dots, Z_J у просторі Z .

Базовий метод класифікації полягає в наступному [22].

1. Побудова опису $O \in Z$ для об'єкта, що розпізнається.
2. Обчислення значень релевантності $d_j(Z_j, O)$ для опису O на множині Z_1, Z_2, \dots, Z_J описів еталонів.

3. Оптимізація на множині значень $d_j(Z_j, O)$, $j = 1, \dots, J$, і визначення номера v класу еталона $v = \arg \underset{M}{\text{opt}} d_j(Z_j, O)$ з найкращим значенням релевантності (максимум для подібності або мінімум для відстані).

Побудуємо класифікаційне правило $g: Z \rightarrow M$ шляхом формування центрів класів $\{m_j\}_{j=1}^J$, $m_j \in Z$ на навчальній множині $Z(J) = \{Z_1, Z_2, \dots, Z_J\}$.

Квантування за допомогою множини опорних точок визначає дискретну модель простору ознак. Класифікацію будь-якого $x \in Z$ здійснюватимемо конкурентним способом

$$v = \arg \min_{j=1, \dots, J} \rho(x, m_j), \quad (2.2)$$

де v – обчислений клас КТ;

$\rho(x, m_j)$ – відстань між дескрипторами КТ.

Зрозуміло, що при розпізнаванні на правило (2.2) на практиці потрібно доповнити логічним аналізом із застосуванням параметра порогу: віднесення дескриптора КТ до класу v здійснюється тільки в разі, якщо для вектора $x \in Z$ буде також виконано умову $\rho(x, m_j) \leq \delta$, де δ – заданий поріг еквівалентності елементів [3, 22]. При відсутності такого обмеження в клас може бути віднесений зовсім не схожий елемент, хоча мінімум в (2.2) завжди буде досягнутий [25].

В результаті застосування (2.2) до фіксованого опису $z(\omega)$ отримаємо вектор $h[z(\omega)] = (h_1, h_2, \dots, h_J)$, де h_j – цілі числа. Цей вектор відповідає розподілу елементів вхідного безлічі дескрипторів по еталонним класам. Аналогічні представлення $h[Z_j]$ можуть бути отримані і для кожного з еталонів Z_j . набір векторів $h[Z_j]$ утворює матрицю еталонних описів бази.

Вектор $h[Z_j]$ може виступити також критерієм якості класифікації на навчальній вибірці. Ідеальним класифікатором є той, для якого виконується умова: $\arg \max_j \{h_j[Z_k]\} = k$, тобто при класифікації еталона з номером k максимальне значення в векторі h досягається для компоненти рядки з номером k . Чисельним критерієм за умови правильної класифікації є ставлення числа дескрипторів, які за результатами обробки віднесені до «правильному» класу, до загальної кількості s дескрипторів бази

$$\beta = \sum_{k=1}^J h_k [Z_k] / s. \quad (2.3)$$

Критерій (2.3) відображає частку правильно класифікованих дескрипторів навчальної вибірки. Ідеальним варіантом є значення $\beta = 1$ [26].

Запропоноване класифікаційне правило g фактично ототожнює клас об'єкта і клас дескриптора КТ, що виключає додаткові витрати на обробку, викликані необхідністю переходу з кластерного простору в безліч класів візуальних об'єктів, як запропоновано у [22].

Виходячи з уявлення простору ознак у вигляді множини бінарних векторів, розглянемо наступний підхід до формування кортежу центрів класів. Визначимо центр класу еталона на основі логічної продукції

$$m_j(a) = \begin{cases} 1, & f(Z_j, a) \geq s(j) / 2, \\ 0, & \text{інакше,} \end{cases} \quad (2.4)$$

де $m_j(a)$ – значення біта на позиції з номером a для центра m_j ;

$s(j) = \text{card } Z_j$ – число елементів еталона;

$f(Z_j, a)$ – функція, яка підраховує кількість одиничних бітів в позиції з номером a в безлічі дескрипторів еталона Z_j , $a = 1, \dots, n$. Функцію $f(Z_j, a)$ можна обчислити безпосереднім складанням бітів як

$$f(Z^j, a) = \sum_{d=1}^{s(j)} x_d(a), \quad x_d \in Z^j, \quad (2.5)$$

де $x_d(a)$ – біт з номером a для дескриптора з номером d в описі зразка.

У виразі (2.5) значення кожного з бітів центру m_j для j -го класу визначається переважною більшістю значень відповідних розрядів усіх дескрипторів КТ, що належать еталону Z_j . Підрахунок числа одиничних

бітів в (2.5) можна здійснити також застосуванням відповідних бінарних функцій.

Аналізуємо елементи $x \in Z$ змісту структурних описів бази еталонів (навчальна вибірка) шляхом віднесення їх до відповідного класу з використанням конкуренції (2.2). В якості ρ можна застосувати метрику Хеммінга

$$\rho(x, m_j) = \sum_{a=1}^n |x(a) - m_j(a)|, \quad (2.6)$$

яка визначає кількість бітів, що відрізняються, для бінарних послідовностей однакової довжини.

2.2 Використання угорського методу

Одним з ефективних способів структурного аналізу даних, пов'язаних з обчисленням оптимального подібності кінцевих множин, є угорська метод, що також називають комбінаторної оптимізацією обмежених множин. Його можна розглядати як рішення задачі про призначення для найбільш загального випадку пошуку максимального паросполучення в дводольному зваженому графі [7]. Принцип оптимального зіставлення структурних описів в порівнянні з голосуванням КТ дозволяє реалізувати схему побудови однозначних відповідностей КТ з порівнюваних описів, що в цілому сприяє поліпшенню достовірності розпізнавання.

Поширена в прикладних дослідженнях класична транспортна задача полягає в мінімізації при деяких обмеженнях лінійної функції виду

$$C = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \text{ що характеризує сумарні витрати при виконанні умов}$$

повного задоволення попиту і рівності кількості продукту, що вивозиться,

запасам його в пунктах відправлення [27]. До завдань транспортного типу зводяться багато інших завдань лінійного програмування – про призначення, мережеві, календарного планування тощо.

Рішення транспортної задачі традиційним симплекс-методом часто не враховує усієї специфіки умов і тому виявляється занадто громіздким. Структура конкретних прикладних обмежень враховується в ряді спеціальних обчислювальних методів її рішення. Одним з найбільш поширених з практичної точки зору обчислювальних підходів вважається Угорський метод (УМ) [7, 28, 29].

Суть УМ складається в послідовному формуванні оптимального рішення за скінченне число кроків. Процедура на основі УМ для вирішення найбільш загальної транспортної задачі складається з попереднього етапу і кінцевого числа однотипних ітерацій. На підготовчому етапі будується матриця $X_0 = (x_{ij}[0])_{m,n}$, елементи якої невід’ємні і задовольняють нерівностям:

$$\sum_{j=1}^n x_{ij}[0] \leq a_i, i = \overline{1, m}, \quad \sum_{i=1}^m x_{ij}[0] \leq b_j, j = \overline{1, n},$$

де m, n – число пунктів відправлення і призначення;

a_i, b_j – запаси і потреби продукту.

Якщо ці умови є рівностями, то матриця X_0 вважається рішенням завдання. Якщо серед цих умов є нерівності, то здійснюється перехід до першої ітерації. На k -ій ітерації будується матриця $X_k = (x_{ij}[k])_{m,n}$. Близькість цієї матриці до вирішення завдання характеризує число Δ_k – сумарна нев’язка матриці X_k :

$$\Delta_k = \sum_{i=1}^m a_i + \sum_{j=1}^n b_j - 2 \sum_{i=1}^m \sum_{j=1}^n x_{ij}[k].$$

В результаті здійснення першої ітерації будується матриця X_1 , що складається з невід'ємних елементів. При цьому виконується умова $\Delta_1 < \Delta_0$, де Δ_0 – сумарна нев'язка матриці X_0 на підготовчому етапі. Якщо $\Delta_1 = 0$, то X_1 вважається оптимальним рішенням. Якщо $\Delta_1 > 0$, то переходять до наступної ітерації. Ітерації проводяться до тих пір, поки Δ_k при деякому значенні k не стане рівним 0. Відповідна матриця X_k і вважається рішенням транспортної задачі.

УМ найбільш ефективний при вирішенні транспортних задач з цілочисельними обсягами виробництва і споживання. У цьому випадку оцінюється число ітерацій не перевищує величини $\Delta_0 / 2$.

Перевагою УМ є можливість в процесі його застосування оцінювати близькість результату кожної з ітерацій до оптимального плану. Це дозволяє контролювати процес обчислень і при необхідності (з метою скорочення витрат часу) припиняти його при досягненні заданої точності. Така властивість особливо важливо для задач великої розмірності, до яких відносяться практично всі завдання обробки зображень, наприклад, число векторів ХО в структурних описах об'єктів досягає двох-трьох сотень.

Важливим з прикладної точки зору окремим випадком транспортної задачі є задача про призначення, яка формулюється наступним чином (рис. 2.1). Є m робіт і n кандидатів для їх виконання. Витрати i -го кандидата на виконання j -ої роботи рівні c_{ij} , $i = \overline{1, m}$, $j = \overline{1, n}$. Кожен кандидат може бути призначений тільки на одну роботу, і кожна робота може бути виконана тільки одним кандидатом. Ознака $x_{ij} = 1$ якщо i -й кандидат виконує j -у роботу, $x_{ij} = 0$ в іншому випадку. Потрібно знайти призначення кандидатів на роботи, при якому сумарні витрати на виконання робіт мінімальні.

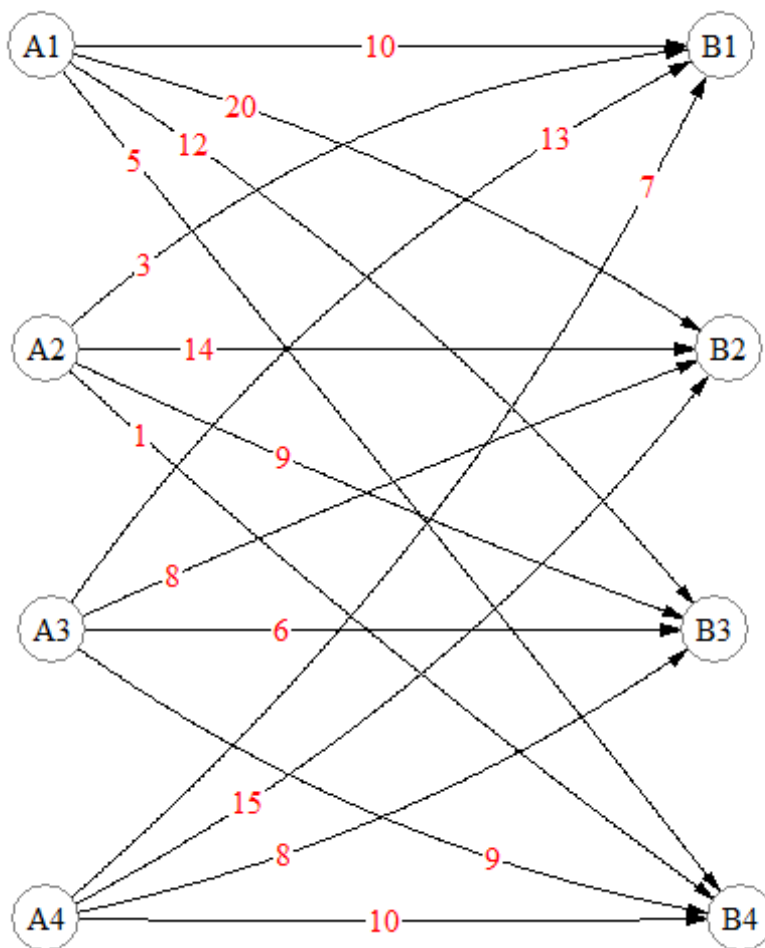


Рисунок 2.1 – Демонстрація задачі про призначення

Візьмемо в якості моделі обчислення оптимального подібності завдання про призначення, де значення x_{ij} будемо використовувати як наявність або відсутність відповідності між елементами порівнюваних описів з номерами i, j . Математична модель задачі про призначення представлена у вигляді:

$$C = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min, \quad (2.7)$$

$$\left\{ \begin{array}{l} \sum_{j=1}^n x_{ij} = 1, \forall i = \overline{1, m}, \\ \sum_{i=1}^m x_{ij} = 1, \forall j = \overline{1, n}, \\ x_{ij} \in \{0, 1\}, i = \overline{1, m}, j = \overline{1, n}. \end{array} \right. \quad (2.8)$$

Задача (2.7) – (2.8) є задачею лінійного програмування, в якій праві частини обмежень рівні 1, а змінні x_{ij} можуть приймати тільки два значення, і тому вона може бути ефективно вирішена за допомогою УМ. Класичний варіант задачі про призначення передбачає мінімізацію цільової функції C для квадратної матриці при $m = n$: $C = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$. Перехід до пошуку

максимуму $C = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \max$ здійснюється шляхом заміни в (2.7) – (2.8):

$$c_{ij} \leftarrow c^* - c_{ij}, \quad c^* = \max_{i, j} \{c_{ij}\}, \quad i, j = \overline{1, n}. \quad (2.9)$$

Практичний інтерес для зіставлення структурних описів довільній розмірності представляють завдання з прямокутною матрицею, вони можуть розглядатися як завдання при $m = n$ після формального доповнення матриці $\{c_{ij}\}$ рядками або стовпцями з нульовими елементами. Сканування нульових рядків або стовпців буде марним, але необхідним етапом реалізації УМ. Задача про призначення вирішує проблему встановлення величини подібності структурних описів об'єктів в умовах невідповідності порядку їх слідування або порушення цілісності опису через перешкоди.

Представимо опис візуального об'єкта Z у вигляді кінцевої множини $Z = \{z_i\}_{i=1}^m$ елементів z_i , де m – кількість ХО в описі, z_i – ХО, яка може представляти як дескриптор (значення ознаки), так і групу дескрипторів, об'єднаних за якимось алгоритмом, тобто кластер. Критерієм при зіставленні елементів є значення деякої метрики $\rho(z_i, z_j)$, яка оцінює відмінності z_i, z_j , взятих з різних описів $z_i \in Z_1, z_j \in Z_2$. Методи з використанням голосування визначають суму голосів елементів, виходячи з величин $\rho(z_i, z_j)$ шляхом оптимізації на кінцевій множині або через оцінку за допомогою деякого порога [30]. Оптимальні методи на основі $\rho(z_i, z_j)$ формують рішення, яке, наприклад, мінімізує суму відстаней між описами Z_1, Z_2 з урахуванням однозначності відповідностей елементів.

Застосуємо УМ для оптимального встановлення величини відповідності між двома структурними описами з урахуванням можливого впливу просторових перешкод, що призводять до спотворень окремих елементів. Результатом виконання УМ є формування максимального паросполучення для елементів двох множин з мінімізацією загальної вартості (ваг), яку можна оцінити у вигляді суми відстаней між парами ХО з порівнюваних описів. Зіставлення Z_1, Z_2 на основі УМ формально зведемо до вирішення оптимізаційної задачі:

$$R(x) = \sum_{i=1}^m \sum_{j=1}^n \rho(z_i, z_j) x_{ij} \rightarrow \min, \quad (2.10)$$

де $z_i \in Z_1, z_j \in Z_2$;

m і n – кількості точок в описах;

x_{ij} – бінарна ознака, $x_{ij} \in \{0,1\}$, що відображає відповідність i -го і j -го елементів з Z_1, Z_2 .

Розв'язок задачі (2.10) при обмеженні на однозначність відповідності ознак із множин, що зіставляються

$$\begin{cases} \sum_{j=1}^m x_{ij} = 1 \quad \forall i = \overline{1, n}, \\ \sum_{i=1}^n x_{ij} = 1 \quad \forall j = \overline{1, m}, \end{cases} \quad (2.11)$$

мінімізує відстань між Z_1, Z_2 .

Як оптимальний, так і традиційний (заснований на голосуванні) підходи використовують в якості базової інформації матрицю відстаней

$$P = \begin{bmatrix} \rho_{11} & \rho_{12} & \dots & \rho_{1n} \\ \rho_{21} & \rho_{22} & \dots & \rho_{2n} \\ \dots & \dots & \dots & \dots \\ \rho_{m1} & \rho_{m2} & \dots & \rho_{mn} \end{bmatrix}$$

між будь-якими парами ХО $z_i \in Z_1, z_j \in Z_2$.

Реалізація принципу однозначного голосування елементів z_i пов'язана з аналізом окремих рядків матриці P на предмет формування голосу, рівного 0 або 1. За отриманим набором відповідей (голосів) можна обчислити значення критерію подібності двох описів у вигляді суми $S_1 = \sum_i L(i)$, де $L(i)$ – предикат, що дорівнює 1, якщо голос формується (відповідність встановлено) і нулю в іншому випадку.

Реалізація УМ передбачає цілеспрямований вибір такої послідовності з $\{\rho_{ij}\}$, сума значень якої $S_2 = \sum \rho_{ij}$ буде мінімальна на безлічі можливих відповідей елементів описів, і при цьому з кожного рядка і стовпця матриці P буде обраний тільки один елемент, що в результаті забезпечує

вибір оптимальної за критерієм S_2 однозначної відповідності множин елементів описів. Реалізація УМ здійснюється шляхом цілеспрямованого еквівалентного перетворення матриці P з послідовним аналізом рядків і стовпців для отримання матриці з невід'ємними елементами і системою m незалежних нулів, з яких ніякі два не належать одному й тому ж рядку або одному і тому ж стовпцю. При досягненні ситуації з m незалежними нулями проблема вибору вважається вирішеною, оптимальний варіант призначень визначається позиціями незалежних нулів в перетвореній матриці [7]. Приклад паросполучення показаний на рисунку 2.2.

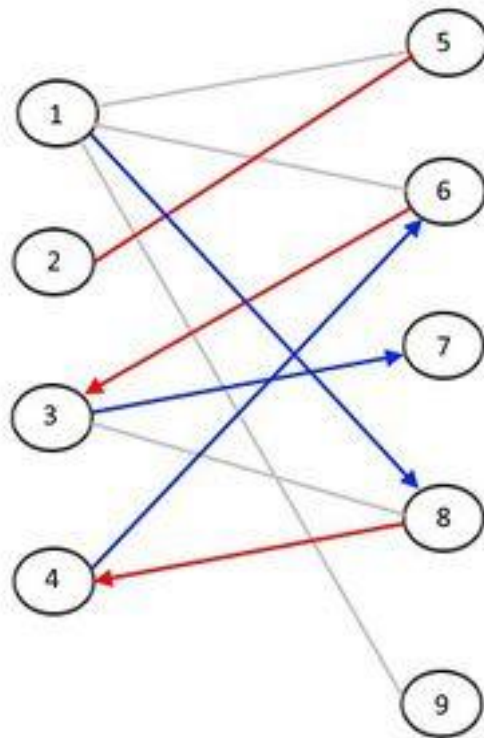


Рисунок 2.2 – Паросполучення для множини чисел

Відомо, що задача пошуку паросполучень з використанням УМ вирішується за час, пропорційний величині d^3 , де d – це розмір порівнюваних описів. Бачимо, що необхідні витрати обчислень суттєво зростають зі збільшенням d . У той же час для методу з голосуванням

витрати прямо пропорційні d , що не так критично до зростання обсягу описів.

При застосуванні УМ для зіставлення описів Z_1, Z_2 їх розмірності m і n в загальному випадку можуть відрізнятися. В цьому випадку в традиційному УМ розмір матриці збільшують до максимального серед m та n , а нові елементи заповнюють нулями [5]. Якщо вихідна матриця не є квадратною, то додатково вводять необхідну кількість рядків (або стовпців), а їх елементам надають значення, які визначаються умовами задачі. Для задачі визначення подібності описів, що отримані за рахунок розширення елементів, необхідно призначити деяке максимальне значення серед можливих значень $\rho(z_i, z_j)$.

Для завдання зіставлення не зовсім доцільно усікати вагову матрицю до квадратної традиційним способом у вигляді $\min(m, n)$, тому що при цьому може бути втрачена цінна інформація про зображення. З іншого боку, подальше ускладнення обробки за рахунок застосування УМ вимагає мінімізації розмірів матриці на етапі попередньої обробки. Це досягається за допомогою застосування спеціальних методів стиснення структурних описів [31-33]. Іншим способом зниження витрат є відбір в деякому сенсі «значущих» рядків і стовпців, наприклад таких, що містять великі за значенням відстані.

2.3 Кластеризація у просторі дескрипторів КТ

Кластерне представлення є результатом попереднього навчання системи розпізнавання на визначеній множині еталонів, які складають базис для розпізнавання. Векторну трансформацію структурного опису – множини можна описати як апроксимацію простору ознак системою кластерів. У результаті такого представлення стає можливим більш продуктивно обчислювати релевантність на структурах даних типу вектор-множина і вектор-вектор.

На попередньому етапі обробки множина еталонів $Z = \{Z^j\}_{j=1}^J$ ХО, у тому числі усі зразки для розпізнавання (Z_j – еталон, J – число еталонних класів), розбивається на кінцеве число k кластерів $M = \{M_i\}_{i=1}^k$, так що $M_i \cap M_d = \emptyset$, $M = Z$, кластери задані множиною їх центроїдів $m = \{m_i\}_{i=1}^k$. Кластеризація відбувається відображенням кінцевої множини ХО в саму себе $Z \rightarrow Z$, причому кожна ХО належить одному з кластерів. У загальному випадку в результаті кластеризації може виявитися, що $m_i \notin M_i$. Множини Z і Z^j є мультимножинами, де близькі між собою ХО вважаються еквівалентними [3, 34]. Після завершення кластеризації еталонної множини здійснюють «просівання» множин ХО кожного з еталонів, в результаті чого опис j -го еталону набуває вигляду цілочисленого вектора

$$H[Z^j] = (h_1, h_2, \dots, h_i, \dots, h_k)^j, \quad (2.12)$$

де $h = \text{card}\{z \mid z \in Z^j \ \& \ z \in M_i\}$;

$h \in C$ – число елементів еталону Z^j , віднесених до кластера M ;

C – множина цілих чисел.

Подання (2.12) – це образ еталона $Z_j \subseteq Z$ в кластерному відображенні.

Обчислення релевантності для пари образів – числових векторів виду (2.12) в обчислювальному плані значно (в десятки разів [22, 34]) менш витратне, ніж зіставлення опису вхідного зображення з множинами $\{Z_j\}$. Найбільш практичні способи – обчислення відстані або коефіцієнта кореляції в просторі векторів [3].

При визначенні релевантності може бути застосований також нормалізований кластерний опис об'єкта $O = \{o_b\}$ у вигляді вектора

$$H[O] = (h_1^n, h_2^n, \dots, h_k^n)^o, \quad h_{ii}^n = h / s^o, \quad (2.13)$$

де $s^o = \text{card}(O) = \sum_{i=1}^k h_i$ – потужність безлічі O .

Нормалізація (2.13) необхідна через різні в загальному випадку кількості елементів в еталонах та об'єктах, що розпізнаються. Однак в результаті нормування (2.13) описи, в яких значення h_n^i близькі, можуть бути визнані ідентичними [35].

Таким чином в результаті здійснення кластерізації ми маємо два різні представлення еталонної множини Z : у вигляді набору еталонних описів $Z = \{Z^j\}_{j=1}^J$ і у вигляді кінцевого числа кластерів $Z = M = \{M\}^k$. Метою розпізнавання є ідентифікація об'єкта з одним з еталонів, а кластерне уявлення застосовується для спрощення обробки і підвищення швидкодії.

Створимо тепер метод розпізнавання для віднесення опису $O = \{o_b\}$ невідомого візуального об'єкта до одного з класів $\{Z^j\}_{j=1}^J$ на основі кластерного опису M бази еталонів. Метод здійснює відображення з множини описів об'єктів O в кінцеву множину номерів класів $\{1, \dots, J\}$ і може бути визначений послідовністю кроків.

Крок 1. Зарахувати кожен елемент $o_b \in O$ об'єкта, що розпізнається, до одного з кластерів $M_i \subseteq M$ відповідно до правила оптимальності

$$o_i \rightarrow M_i \mid \arg \min_d \rho(o_b, m_d) = i, \quad (2.14)$$

де $\rho(o_b, m_d)$ – метрика на множині ХО.

Правило (2.14) виконує конкурентне віднесення ХО o_b за принципом оптимальної близькості в системі кластерів. Характеристикою кластера виступає його центр $m_i \in M_i$. Тут же реалізують фільтрацію несхожих (помилкових) елементів. Для цього після (2.14) додатково виконують порогову верифікацію встановленого мінімуму $m_i: \rho(o_b, m_i) \leq \delta$, де δ – деякий поріг значимості для мінімуму відстані. Якщо умова не виконана,

елемент o_b об'єкта, що розпізнається, не відносять до жодного з кластерів і вважають помилковим.

Крок 2. Реалізуємо крок 1 для усіх $o_b \in O$, в результаті формуються описи виду (2.12), (2.13) об'єкта як $O = (h_1, h_2, \dots, h_k)^o$, де h_i – число елементів, віднесених до кластеру M_i .

Крок 3. Обчислимо ступінь r_j релевантності між нормалізованими кластерними описами (2.13) об'єкта і кожного з еталонів як відстань $r_j = \beta(H[O_n], H[Z_n^j])$ у векторному просторі R^k .

Крок 4. Об'єкт O віднесемо до класу $d \in \{1, \dots, J\}$ за правилом $d = \arg \min r_j$.

Відстані $\rho(\dots)^j$ для елементів і $\beta(\dots)$ для векторних описів у загальному випадку різні, замість них застосовуються також міри подібності, наприклад, коефіцієнт кореляції, а поріг δ визначається рівнем розрізнення в конкретній базі еталонів і допустимим рівнем завад. На етапі 4 також може бути застосована перевірка значущості отриманого мінімуму релевантності $r_d : r_d \leq \varepsilon$, де ε – поріг. При незадоволенні нерівності клас об'єкта вважаємо невизначеним, тому що відсутня значима відповідність в наявному просторі еталонів.

Розглянутий метод не враховує важливі для результативного розпізнавання співвідношення характеристик побудованих кластерів із властивостями набору взаємно розрізнених еталонів у складі бази зразків. Для урахування цієї ключової обставини застосуємо в процесі розпізнавання векторний показник значущості класів для кожної з ознак $o_b \in B$ [36]. Значущість для ХО o_b визначимо у вигляді вектора α^i ваг класів, який буде приписаний окремому кластеру. Будемо формувати α^i на етапі попереднього навчання системи розпізнавання, враховуючи особливості конкретної бази еталонів в аспекті її кластерного представлення. Визначимо

α^i для кластера M_i у вигляді вектора $\alpha^i = (\alpha_1^i, \alpha_2^i, \dots, \alpha_J^i)$, компоненти якого задамо співвідношенням

$$\alpha_d^i = c_d^i / s_i, \quad (2.15)$$

де $c_d^i = \text{card}\{z \mid z \in Z^d \ \& \ z \in M_i\}$ – число елементів еталонного класу Z^d , визначених за результатами навчання в кластер M_i ;

$s_i = \text{card}(M_i)$ – число елементів кластера M_i .

Значення (2.15) відповідає нормованому поданню (2.13) для кластерного розкладання еталона по множині класів. З (2.15) видно, що $\sum \alpha_d^i = 1$, так що α_d^i можна вважати оцінкою ймовірності події, що елемент кластера M_i відноситься до класу Z^d . Величини α_d^i є відносні ваги ХО кластера в плані його віднесення до еталонів класів. Зокрема, максимальне значення серед компонентів α^i відповідає найбільш ймовірному класу, до якого може належати ХО. Очевидно, що значення характеристик (2.15) безпосередньо залежать як від змісту бази еталонів, так і від застосовуваного методу кластеризації, який встановлює відображення еталон-кластер [36].

Модифікуємо метод розпізнавання додаванням вектора ваг α^i в процес прийняття рішення. Віднесення елемента o_d до одного з кластерів на кроці 1 тепер супроводжуємо формуванням суми значень апріорних векторів α^i , в результаті чого після аналізу усіх значущих $o_d \in O$ на кроці 2 буде сформований сумарний для елементів об'єкта вектор характеристик класів

$$\sum = \sum_o \alpha^i = (\sum_1, \sum_2, \dots, \sum_J), \quad \sum_d = \sum_i \alpha_d^i. \quad (2.16)$$

Підсумовування в (2.16) здійснюється по усіх значимих відповідностях кластерам для множини ХО об'єкта.

Модифікація кроку 4 методу набуває наступного вигляду: об'єкт відносимо до класу $d \in \{1, \dots, J\}$ за правилом:

$$O \rightarrow d \mid d = \arg \max_j \Sigma_j. \quad (2.17)$$

У модифікованому методі розпізнавання також можуть бути застосовані перевірки на значущість при віднесенні до кластеру (на кроці 1) і на значущість максимуму подібності в (2.17) (крок 4). Зокрема, значимість максимуму в (2.17) може бути підтверджена ступенем переважання глобального максимуму над найближчим з локальних [37].

Як бачимо, перший метод базується на процедурі голосування (2.14), в основі якої лежить конкурентний вибір центру кластера для кожної ХО об'єкта і аналіз подібності кластерного опису об'єкта на множині еталонів. Тут виконано розкладання у сенсі «еталон за кластерами». Ідея ж модифікованого методу заснована на побудові інтегрального ймовірнісного розподілу кластерів на множині ХО об'єкта в наявному діапазоні класів. Тут інформаційною основою є розкладання «кластер за еталонами». Це має забезпечити більш глибоке узгодження зі значеннями ХО еталонних зразків, що в цілому поліпшить результативність розпізнавання. Модифікований метод також можна вважати голосуванням ХО об'єкта, зваженим розподілом наявних класів бази зображень по побудованій множині кластерів.

У прикладних дослідженнях при наявності завад доцільно рекомендувати схему аналізу на кроці 1 розглянутих методів, коли кожен кластер бази еталонів «відбирає» відповідні йому елементи об'єкта, тобто пошук відповідності для ХО здійснюється «від еталону», а не «від об'єкта». Нормалізація при цьому може бути виконана на основі числа відібраних ознак об'єкта, що аналізується.

Узагальнюючи, вихідними даними для розпізнавання є цілочисельна матриця кластерного уявлення H розмірністю $J \times k$, де число рядків

дорівнює кількості еталонів, а число стовпців – кількості кластерів. Рядок матриці – це розкладання наявних даних типу «еталон за кластерами», а стовпець – типу «кластер за еталонами». Кожен зі зрізів (рядки, стовпці) допускає проведення додаткової обробки з метою поліпшення властивостей розпізнавання. Однак при цьому потрібно вводити зміни і в метод розпізнавання. Іншим напрямком удосконалення є зміна результатів кластеризації за рахунок застосування інших методів.

Матриця кластерного представлення H є основою процедури розпізнавання. Проаналізуємо тепер можливості вивчення її якісних властивостей з метою поліпшення розпізнавання, а також можливість її додаткової логічної обробки. Зауважимо, що зміна матриці вплине і на вагові коефіцієнти α^i . Таку обробку здійснюють на попередньому етапі, тому час розпізнавання не збільшується. Додаткова логічна обробка може бути необхідна в цілях подальшого стиснення і уточнення трансформованих описів, виключення кластерів незначного обсягу, тобто для формування конструктивної системи кластерів, в рамках якої можна реалізувати якісне розпізнавання. Іншим методом обробки може бути виключення (занулення) в стовпцях матриці незначних у порівнянні з іншими елементів.

Для аналізу або логічної обробки значень h_i стовпців матриці H , що характеризують кластер M_i , застосуємо наступний критерій [3]

$$\gamma_i = \sum_a^J = \sum_{b>a} |h_i[a] - h_i[b]|, \quad (2.18)$$

де $h_i[a]$ – компонента i -го стовпця.

Величина (2.18) дорівнює нулю в разі повного співпадіння значень в стовпці і зростає зі збільшенням їх відмінності між собою. Замість (2.18) можна застосувати також дисперсійні характеристики значень стовпців. Обробка може складатися в обчисленні величини (2.18) для стовпців матриці H і відкиданні як непридатних для розпізнавання стовпців з найменшими

значеннями критерію. В результаті отримаємо усічену по числу стовпців матрицю H . Зауважимо, що введення логічної обробки призводить до трансформації наявних даних і до необхідності у зв'язку з цим деяких змін алгоритму розпізнавання. Остаточна ефективність таких евристичних обробок може бути оцінена лише на основі експериментів по розпізнаванню в прикладній задачі. У той же час, величина компонент вектора γ_i може виступити критерієм якості кластерного представлення: чим більш віддалені значення (2.18) від нуля, тим краще якість кластеризації в аспекті розпізнавання по сформованій системі кластерів.

Варто зазначити, що в реальних умовах структурного розподілу ситуації з низькими значеннями міри (2.18) виникають достатньо рідко [38], тому що структурні описи зображень все-таки помітно відрізняються між собою. У той же час перевага окремих значень у стовпцях матриці H є властивістю ХО зображень [39].

3 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ НА ПІДСТАВІ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ

3.1 Обґрунтування вибору середовища програмної реалізації

Для реалізації алгоритмів визначення релевантності зображень розроблений програмний продукт з використанням мови Java (Oracle/HotSpot™ v.13.0.1).

Процес розробки вівся у інтегрованому середовищі IntelliJ Idea 2020 від компанії JetBrains [40]. Реалізація алгоритмів пошуку та опису КТ, а також малювання КТ на зображенні були взяті з бібліотеки OpenCV 4.1.0 [41]. У якості інструмента для керування залежностями та збірки проекту був використаний Maven [42]. Тестування програми проводилось на комп'ютері з ОС Windows 10 Pro 20H2 з процесором Intel(R) Core(TM) i7-9750H 2.60 ГГц та 16 Гб оперативної пам'яті.

Для запуску застосування достатньо комп'ютера або ноутбука з процесором серії Intel Core i3 першого покоління чи вище та мінімум 1 Гб оперативної пам'яті. На комп'ютері повинно бути встановлено Java Runtime Environment від Oracle/HotSpot версії 8 чи вище.

3.2 Інструкція користувача

Програма являє собою Java-застосунок «ImageClassifier», який зібраний у JAR та запускається з командного рядка командою

```
java -jar ImageClassifierEntryPoint.jar [parameters list].
```

Параметри (parameters list) подаються до програми у вигляді

```
--paramName1=paramValue1 --paramName2=paramValue2.
```

Програма має обов'язкові та не обов'язкові параметри:

- `modelFilesLocation` – обов'язковий параметр. Повний або відносний шлях до директорії з зображеннями, які будуть використані як еталони;
- `sampleFile` – обов'язковий параметр. Повний або відносний шлях до тестового зображення;
- `outFile` – не обов'язковий параметр. Повний або відносний шлях до файлу, куди запишеться тестове зображення з КТ розділеними на кластери, що відмічені різними кольорами;
- `nfeatures` – не обов'язковий параметр. Кількість ключових точок, які алгоритм буде шукати на зображеннях. Початкове значення – 500;
- `algorithm` – не обов'язковий параметр. Алгоритм для пошуку та опису КТ. Допустимі значення – ORB та BRISK, початкове значення – ORB.

3.3 Результати моделювання та експериментального дослідження

Для прикладу класифікації зображень та оцінювання ефективності запропонованих підходів будемо використовувати десять еталонних зображень (додаток А) [20], проводити кластеризацію у просторі КТ та згодом голосування КТ за кластери (у першому варіанті) або знаходити відповідності між центрами кластерів за допомогою УМ (у другому варіанті). Кластеризація у просторі КТ виконується за реалізованим алгоритмом k -середніх з манхеттенською метрикою у якості відстані.

В якості експерименту будемо класифікувати три зображення-зразка, що наведені у додатку Б, двома способами та нотувати відстані до усіх кластерів, утворених з зображень-еталонів.

На рисунку 3.1 наведений приклад кластеризації КТ для деяких зображень з фіксованою кількістю кластерів ($k = 3$). Самі центри кластерів ніяк не відмічені, бо, за запропонованою реалізацією алгоритму k -середніх,

точки, що представляють ці центри, можуть не належати до переліку знайдених КТ зображень.



Рисунок 3.1 – Зображення з КТ, що розбиті на кластери

Розпочнемо дослідження методів з голосування. У таблицях 3.1 – 3.4 наведені числові (у %) долі дескрипторів КТ, що голосували за той чи інший еталон, для трьох зображень-зразків. При цьому для використовувалася детектор і дескриптор ORB, що знаходив 300, 500, 1000 та 1500 КТ відповідно.

Таблиця 3.1 – Результати голосування при N КТ = 300

Еталон Зразок	1	2	3	4	5	6	7	8	9	10
3`	10	9,7	23,3	11,7	9,3	8,3	4	8,6	7	8
5`	9	13,6	5,7	9,3	27,3	8,6	10	4,7	4,7	7
6`	8,3	6,3	2,7	14,6	8,3	36,3	7,3	4,3	2,7	9

Таблиця 3.2 – Результати голосування при N КТ = 500

Еталон Зразок	1	2	3	4	5	6	7	8	9	10
3`	7,4	8,8	23,8	9,6	12,6	7	6,8	9,4	6,6	8
5`	11	9,8	10,2	7,2	29,8	9,2	3,8	5,8	3,2	10
6`	9,6	9,8	8,2	12	9,4	26	6,6	4,8	5,6	8

Таблиця 3.3 – Результати голосування при N КТ = 1000

Еталон Зразок	1	2	3	4	5	6	7	8	9	10
3`	9,3	6,6	26,2	10,6	10,4	6,5	6,1	8,3	10	7
5`	9,5	6,9	9,1	8,8	28,4	7,8	8	5,8	5,2	10,5
6`	8,4	7,4	6,8	10,8	11	25,2	8,1	4,1	9,5	8,5

Таблиця 3.4 – Результати голосування при N КТ = 1500

Еталон Зразок	1	2	3	4	5	6	7	8	9	10
3`	9	6,3	23,8	6,7	9,4	10,4	11,4	9,7	7,5	5,7
5`	8,9	8	9,7	6,5	24,1	9,5	10,3	8,6	5,3	9,1
6`	9,1	9,5	7,3	11,4	5,7	25,3	10,4	3,9	9,5	7,8

На рисунку 3.2 зображений вивід програми для одного із зразків.

```
Cluster center 0 voted for image 5.png
Cluster center 1 voted for image 5.png
Cluster center 2 voted for image 5.png
Голосів за еталон 1.png = 27 (9%)
Голосів за еталон 10.png = 21 (7%)
Голосів за еталон 2.png = 41 (13.67%)
Голосів за еталон 3.png = 17 (5.67%)
Голосів за еталон 4.png = 28 (9.33%)
Голосів за еталон 5.png = 82 (27.33%)
Голосів за еталон 6.png = 26 (8.67%)
Голосів за еталон 7.png = 30 (10%)
Голосів за еталон 8.png = 14 (4.67%)
Голосів за еталон 9.png = 14 (4.67%)
```

Рисунок 3.2 – Голосування за еталони для одного із зразків

Як бачимо, для будь-яких кількостей КТ дескриптори зразків голосували за «правильні» еталони, що доводить працездатність цього методу. Доля дескрипторів не пов'язана будь-яким чином з кількістю КТ, що не підтверджує гіпотезу про залежність ефективності класифікації голосуванням від кількості КТ.

Тепер дослідимо метод класифікації, що полягає у застосуванні угорського методу для пошуку максимальних паросполучень між центрами кластерів еталонів та зразка. Ця процедура може бути реалізована двома шляхами:

- складення зведеної матриці H відстаней між центрами кластерів зразка та центрами кластерів усіх еталонів та знаходження оптимальних паросполучень на ній (розмірність матриці $k \times J * k$, де k – кількість кластерів, $k = 3$, J – кількість еталонів, $J = 10$);

- складення 10 матриць відстаней розмірністю $k \times k$, по одній на кожний еталон, оптимізація кожної з них та підрахунок найкоротшого шляху у кожній з них. Найменше значення довжини серед еталонів визначить результат класифікації.

Для експериментів були реалізовані обидва варіанти, з яких другий виявився чуть швидшим та, що важливіше, таким, що дає більш коректні результати. Вивід програми для одного із зразків зображений на рисунку 3.3.

```

    0    1    2
  115  200  182
  206  165   97
  109  204  216
[0=1, 1=2, 2=0]
406.0
    0    1    2
   48  187  184
  203  136  221
  142  237   80
[0=0, 1=1, 2=2]
264.0
    0    1    2
   75  205  188
  216  170  121
  151  121  218
[0=0, 1=2, 2=1]
317.0

```

Рисунок 3.3 – Матриці відстаней, результат роботи УМ та довжина найкоротшого шляху у матриці

У таблицях 3.5 – 3.8 занесені результати роботи УМ, а саме довжини найкоротших шляхів у матрицях відстаней між центрами кластерів.

Таблиця 3.5 – Результати використання УМ при N КТ = 300

Эталон Зразок	1	2	3	4	5	6	7	8	9	10
3`	224	243	123	203	247	237	199	218	153	200
5`	217	268	196	246	128	200	215	207	221	213
6`	213	242	199	222	174	138	211	259	148	233

Таблиця 3.6 – Результати використання УМ при N КТ = 500

Эталон Зразок	1	2	3	4	5	6	7	8	9	10
3`	212	230	133	242	201	197	224	193	179	198
5`	248	182	227	276	133	221	238	193	199	214
6`	232	196	227	248	211	49	212	213	183	106

Таблиця 3.7 – Результати використання УМ при N КТ = 1000

Эталон Зразок	1	2	3	4	5	6	7	8	9	10
3`	242	224	111	198	191	207	214	234	237	196
5`	168	251	216	177	92	191	190	170	221	118
6`	149	183	256	177	226	101	187	183	200	176

Таблиця 3.8 – Результати використання УМ при N КТ = 1500

Эталон Зразок	1	2	3	4	5	6	7	8	9	10
3`	406	317	194	429	415	493	302	390	403	264
5`	457	386	335	458	256	470	373	431	506	349
6`	345	312	349	378	332	203	426	343	396	299

Проаналізувавши отримані таблиці, виявилось, що застосування УМ дозволяє на даній базі еталонів проводити класифікацію зразків із задовільною якістю. Кількість КТ, що знаходяться дескрипторним алгоритмом, не впливає на якість розпізнавання та обирається експериментально для кожної конкретної бази еталонів.

Далі проведемо порівняння швидкості роботи двох представлених раніше методів класифікації за множиною незалежних кластерів. Конфігурація системи була описана у 3.1. Тести будемо проводити як процедуру класифікації одного з еталонів, а саме 3`, з різної кількістю КТ. Будемо вимірювати сумарний час роботи алгоритму (з урахуванням знаходження КТ та дескрипторів) t_{total} , а також окремо час виконання кластеризації методом k -середніх з фіксованою кількістю кластерів t_{clast} та час специфічної логіки класифікації t_{logic} . Результатом вважатимемо середнє значення серед трьох викликів алгоритму з однаковими параметрами з попереднім програвом. Заміри будемо заносити у таблицю 3.9.

Таблиця 3.9 – Виміри швидкості роботи алгоритмів

Голосування				Угорський метод			
Кі-сть КТ	t_{clast} , мс	t_{logic} , мс	t_{total} , мс	Кі-сть КТ	t_{clast} , мс	t_{logic} , мс	t_{total} , мс
300	110	4	1426	300	113	2	1157
500	147	7	1192	500	147	2	1258
1000	237	14	1276	1000	260	2	1284
1500	344	21	1313	1500	327	2	1375

Згідно з вимірами, основну частину роботи програми займає пошук та опис КТ за допомогою алгоритму ORB бібліотеки OpenCV [9]. Такий параметр як кількість КТ, як і очікувалось, не впливає на швидкість у випадку використання УМ, але не критично збільшує час голосування. На практиці використання кількості КТ більш, ніж $N=2000$, навряд чи буди

виправдано, тому швидкість голосування буде задовільною у будь-якому випадку. Швидкість реалізації алгоритму k -середніх також можемо назвати прийнятною.

Підсумовуючи, можна сказати, що обидва розглянуті методи досягають прийнятної якості класифікації із будь-якою кількістю КТ для виділення пошуковим алгоритмом ORB. З отриманих числових даних помітно, що алгоритм, що базується на голосуванні, більш впевнено класифікує усі зразки. Перевага підходу з використанням угорського методу у швидкодії та фіксованому часі роботи, бо цей алгоритм працює не з окремими КТ та дескрипторами, а лише з фіксованою кількістю центрів кластерів.

ВИСНОВКИ

У рамках атестаційної роботи були розроблені та програмно реалізовані методи класифікації зображень на основі моделі голосування та з використанням угорського алгоритму. Також була досліджена доцільність використання алгоритмів кластеризації на просторі структурних ознак зображення задля зменшення розмірності даних та збільшення швидкодії.

Розглянуті методи реалізують процедуру розпізнавання візуального об'єкта на основі кластерного уявлення у просторі структурних ознак у формі зрізу «еталон за кластерами». Експерименти показали, що проведення кластеризації на даних структурного опису зображення забезпечує необхідну результативність класифікації, дає можливість суттєво зменшити обсяг обчислень (у порівнянні з класичним голосуванням дескрипторів) та приймати рішення про належність до того чи іншого за окремими представниками множини КТ, що важливо у прикладному сенсі.

Результати даного дослідження апробовано на 23-му Міжнародному молодіжному форумі «Радіoeлектроніка та молодь у ХХІ столітті» [43] та у статті [44].

Подальше дослідження методів на основі кластерного представлення структурного опису може включати:

- вибір іншого алгоритму для пошуку КТ та дескрипторів, наприклад, BRISK;
- вибір іншого алгоритму кластеризації з визначеною або невизначеною кількістю кластерів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Шапиро, Л., & Стокман, Д. (2006). Компьютерное зрение.
2. Путятин, Е. П. (1990). Обработка изображений в робототехнике. Москва.
3. Гороховатский, В. А. (2014). Структурный анализ и интеллектуальная обработка данных в компьютерном зрении.
4. Путятін, Є. П., Гороховатський, В. О., & Матат, О. О. (2006). Методи та алгоритми комп'ютерного зору: навч. посіб. Харків: ТОВ «Компанія СМІТ».
5. Полякова, Т. В., Куликов, Ю. А., & Гороховатский, В. А. (2013). Классификация изображений на основе признаков пространственных структур.
6. Gorokhovatskyi, V.O., Tvoroshenko, I.S., and 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), pp. 59-69. DOI: 10.1615/TelecomRadEng.v79.i1.60. Available online: <http://www.dl.begellhouse.com/ru/journals/0632a9d54950b268,706f31e606bab3dd,2492ae044887304a.html>.
7. Гороховатский В.А., Полякова Т.В. (2018). Применение пространственных структур признаков для классификации изображений в компьютерном зрении (монография). – Харьков: ФОП Панов А.Н., 120 с.
8. Гороховатський В.О., Гадецька С.В. (2020) Статистичне оброблення та аналіз даних у структурних методах класифікації зображень (монографія), Харків, ФОП Панов А.Н., 128 с., DOI: 10.30837/978-617-7859-69-6.
9. OpenCV Open Source Computer Vision [Електронний ресурс]. – Режим доступу: <https://docs.opencv.org/master/index.html>.
10. Harris, C. G., & Stephens, M. (1988, August). A combined corner and edge detector. In Alvey vision conference (Vol. 15, No. 50, pp. 10-5244).

11. Shi, J. (1994, June). Good features to track. In 1994 Proceedings of IEEE conference on computer vision and pattern recognition (pp. 593-600). IEEE.
12. Гороховатський, О. В., Пупченко, Д. В., & Солодченко, К. Г. (2018). Аналіз властивостей, характеристик та результатів застосування новітніх детекторів для визначення особливих точок зображення.
13. Гороховатский, В. А. (2008). Метрики на множествах ключевых точек изображений.
14. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110.
15. Bay, H., Tuytelaars, T., & Van Gool, L. (2006, May). Surf: Speeded up robust features. In *European conference on computer vision* (pp. 404-417). Springer, Berlin, Heidelberg.
16. Rosten, E., & Drummond, T. (2006, May). Machine learning for high-speed corner detection. In *European conference on computer vision* (pp. 430-443). Springer, Berlin, Heidelberg.
17. 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.
18. 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.
19. 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.
20. Бібліотека зображень покемонів [Електронний ресурс]. – Режим доступу: <https://www.pokemon.com/ru/pokedex/>.
21. Гороховатский, В. А., Путятин, Е. П., & Столяров, В. С. (2017). Исследование результативности структурных методов классификации изображений с применением кластерной модели данных. *Радиоелектроніка, інформатика, управління*, (3), 78-85.

22. Gorokhovatskiy, V. A. (2016). Efficient Estimation of Visual Object Relevance during Recognition through their Vector Descriptions. *Telecommunications and Radio Engineering*, 75(14).
23. Duda, R. O., & Hart, P. E. (2006). *Pattern classification*. John Wiley & Sons.
24. Tvoroshenko, I.S., Gorokhovatsky, V.O. (2019). Intelligent classification of biophysical system states using fuzzy interval logic. *Telecommunications and Radio Engineering*, 78 (14), pp. 1303-1315. – doi: 10.1615/TelecomRadEng.v78.i14.80.
25. Гороховатський, В. А., & Куликов, Ю. А. (2010). Модели обработки дескрипторов характерных признаков изображений на основе анализа гистограмм. *Системи обробки інформації*, (9), 145-148.
26. Gorokhovatskyi O., Gorokhovatskyi V., Peredrii O. (2018). Analysis of Application of Cluster Descriptions in Space of Characteristic Image Features. *Data*, 3(4), 52. – doi: 10.3390/data3040052. Available online: <https://www.mdpi.com/2306-5729/3/4/52>.
27. Gorokhovatskiy, V. A., & Putyatin, Y. P. (2009). Image Likelihood Measures on the Basis of the Set of Conformities. *Telecommunications and Radio Engineering*, 68(9).
28. Гороховатский, В. А., & Полякова, Т. В. (2011). Оптимальные методы сопоставления структурных описаний видеообъектов. *Бионика интеллекта*, (3), 77.
29. Гороховатский, В. А., & Полякова, Т. В. (2011). Критерии и модели структурной классификации с применением принципа голосования. *Системи обробки інформації*, (2), 12-16.
30. Гороховатский, В. А., & Путятин, Е. П. (2008). Структурное распознавание изображений на основе моделей голосования признаков характерных точек. *Реєстрація, зберігання і обробка даних*.
31. Gorokhovatskyi, V., Gorokhovatskyi, O., Yevgenyi, P., & Olena, P. (2018, August). Quantization of the Space of Structural Image Features as a Way

to Increase Recognition Performance. In 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP) (pp. 464-467). IEEE.

32. Gorokhovatskiy, V. A. (2011). Compression of descriptions in the structural image recognition. *Telecommunications and Radio Engineering*, 70(15).

33. Lyashenko, V., Kobylin, O., & Ahmad, M. A. (2014). General methodology for implementation of image normalization procedure using its wavelet transform. *International Journal of Science and Research (IJSR)*, 3(11), 2870-2877.

34. Гороховатский, В. А., Дунаевская, М. Д., & Струненко, В. А. (2016). Изучение свойств методов кластеризации применительно к множествам характерных признаков изображений. *Системы обработки информации*, (5), 124-127.

35. Gorokhovatskiy, 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).

36. Гороховатский, В. А., & Столяров, В. С. (2016). Классификация изображений на основе кластерного представления структурных описаний.

37. Гороховатский, В. А. (2008). Иерархия пространственных отношений структурных признаков в задачах сопоставления визуальных объектов. *Системы управління, навігації та зв'язку*. – К.: Центральний наук.-досл. інститут навігації і управління, 85-89.

38. Гороховатский, В. А., Гороховатский, А. В., & Берестовский, А. Е. (2016). Структурное распознавание изображений с применением моделей интеллектуальной обработки и самоорганизации признаков. *Радіоелектроніка, інформатика, управління*, (3 (38)).

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

40. IntelliJ IDEA: The Java IDE for Professional Developers by JetBrains [Електронний ресурс]. – Режим доступу: <https://www.jetbrains.com/idea>.
41. OpenCV 4.1 [Електронний ресурс]. – Режим доступу: <https://opencv.org/opencv-4-1>.
42. Apache Maven Project [Електронний ресурс]. – Режим доступу: <https://maven.apache.org>.
43. Запорожченко А.П. (2019). Визначення релевантності описів зображень за розподілами значень фрагментів. 23-й Міжнародний молодіжний форум «Радіоелектроника та молодь у ХХІ столітті». Збірник матеріалів форуму, (7), 13-14.
44. Gorokhovatskyi, V., Zaporozhchenko, A., Siryk T., & Tarasenko, O. (2020). Дослідження результативності застосування ознак розподілів даних для обчислення релевантності описів зображень. Системи управління, навігації та зв'язку. Збірник наукових праць, 1(59), 68-73.