



## Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2022 р.

**ЗАВДАННЯ**

## НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Пронюк Олені Дмитрівні  
(прізвище, ім'я, по батькові)1. Тема роботи Моделювання методів класифікації зображень на підставі засобів навчання з учителем

затверджена наказом університету від 16 травня 2022 року № 541Ст

2. Термін подання студентом роботи до екзаменаційної комісії 28 травня 2022 р.

3. Вихідні дані до роботи алгоритми детектування ключових точок зображень та визначення дескрипторів, теоретичні відомості про методи класифікації зображень, теоретичні відомості про засоби навчання систем розпізнавання, тестові зображення.

4. Перелік питань, що потрібно опрацювати в роботі

1. Огляд методів класифікації зображень.2. Аналіз методу ORB для знаходження дескрипторів ключових точок зображення.3. Аналіз мереж Кохонена задля навчання систем розпізнавання зображення.4. Застосування моделі конкурентного навчання Кохонена для класифікації зображень.5. Тестування розробленої моделі застосування навчання з учителем у методах класифікації зображень.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми класифікації зображень, постановка задачі, формування дескрипторів ключових точок, суть конкурентного навчання Кохонена, тестові зображення, аналіз результатів моделювання, висновки.

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

| Найменування розділу                               | Консультант<br>(посада, прізвище, ім'я, по батькові) | Позначка консультанта про виконання розділу |      |
|--|--|---|------|
|  |  | підпис                                      | дата |
| Консультант з дотримання діючих стандартів та норм | доцент Белова Н.В.                                   |   |      |

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів роботи                         | Терміни виконання етапів роботи | Примітка |
|-------|---|---------------------------------|----------|
| 1     | Отримання завдання на кваліфікаційну роботу | 18.04.2022                      |          |
| 2     | Аналіз завдання, підбір літератури          | 18.04.22-21.04.22               |          |
| 3     | Аналіз літератури з досліджуваної проблеми  | 22.04.22-25.04.22               |          |
| 4     | Аналіз технічних засобів                    | 26.04.22-30.04.22               |          |
| 5     | Розробка методу                             | 01.05.22-14.05.22               |          |
| 6     | Програмна реалізація                        | 15.05.22-23.05.22               |          |
| 7     | Оформлення пояснювальної записки            | 24.05.22-26.05.22               |          |
| 8     | Перевірка на плагіат                        | 29.05.22                        |          |
| 9     | Рецензування                                | 30.05.22                        |          |
| 10    | Підготовка презентації та доповіді          | 30.05.22-4.06.22                |          |
| 11    | Занесення роботи в електронний архів        | 31.05.22                        |          |
| 12    | Попередній захист кваліфікаційної роботи    | 06.06.22                        |          |

Дата видачі завдання 18 квітня 2022 р.

Студент \_\_\_\_\_  
(підпис)

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

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 66 с., 7 табл., 16 рис., 2 дод., 44 джерела.

КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, КЛЮЧОВА ТОЧКА, ДЕСКРИПТОР, МЕРЕЖІ КОХОНЕНА, НАВЧАННЯ З УЧИТЕЛЕМ, КОНКУРЕНТНЕ НАВЧАННЯ.

Об'єктом роботи є методи класифікації зображень у системах комп'ютерного зору.

Метою роботи є підвищення результативності методів класифікації зображень шляхом впровадження засобів навчання з учителем на підставі мережі Кохонена.

Проведено моделювання методів класифікації зображень із використанням опису у вигляді множини дескрипторів ORB. Класифікація виконана з використанням прикладної бази зображень брендів. Впроваджено модель конкурентного навчання Кохонена у формі навчання з учителем, розроблено алгоритм і програму для методу класифікації.

У результаті роботи здійснена програмна реалізація і експериментальне дослідження методу класифікації зображень із використанням засобів навчання з вчителем.

CLASSIFICATION OF IMAGES, FEATURES, DESCRIPTOR, KOHONEN NETWORKS, SUPERVISED LEARNING, COMPETITIVE LEARNING.

The object of the work is the methods of image classification in computer vision systems.

The aim of the work is to increase the effectiveness of image classification methods by introducing supervised learning based on the Kohonen network.

Modeling of image classification methods using the description in the form of a set of ORB descriptors is developed. The classification is performed using the applied image database of brands. Kohonen's model of competitive learning in the form of supervised learning was introduced, an algorithm and a program for the classification method was developed.

As a result of work the program realization and experimental research of a method of images classification with use of means of supervised learning is carried out.

## ЗМІСТ

|  |    |
|--|----|
| Перелік умовних позначень, символів, одиниць, скорочень і термінів ..... | 6  |
| Вступ.....   | 7  |
| 1 Аналіз методів класифікації зображень.....                             | 10 |
| 1.1 Основні поняття проблеми розпізнавання зображень .....               | 10 |
| 1.2 Класифікація та кластеризація у системах розпізнавання.....          | 14 |
| 1.3 Визначення ключових точок зображення .....                           | 15 |
| 1.4 Постановка задачі .....  | 18 |
| 2 Засоби навчання з учителем для класифікації зображень .....            | 20 |
| 2.1 Аналіз властивостей детекторів ключових точок .....                  | 20 |
| 2.2 Аналіз методів машинного навчання.....                               | 28 |
| 2.3 Моделі навчання з учителем.....                                      | 31 |
| 3 Програмне моделювання методу класифікації зображень .....              | 42 |
| 3.1 Особливості програмного моделювання методу ORB.....                  | 42 |
| 3.2 Модель модифікації мережі Кохонена для навчання з учителем..         | 44 |
| 3.3 Програмна реалізація моделі класифікації та її тестування .....      | 46 |
| Висновки .....   | 51 |
| Перелік джерел посилання .....   | 52 |
| Додаток А Тестові зображення.....  | 57 |
| Додаток Б Лістинг програмного додатку .....                              | 58 |

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

AKAZE – Accelerated KAZE (прискорений KAZE)

BRIEF – Binary Robust Independent Elementary Features

BRISK – Binary Robust Invariant Scalable Keypoints

DoG – Difference of Gaussians (різниця Гаусових фільтрів)

FAST – Features from Accelerated Segment Test

FED – Fast Explicit Diffusion

M-LDB – Modified-Local Difference Binary

ORB – Oriented FAST and Rotated BRIEF

SIFT – Scale Invariant Feature Transform (масштабно-інваріантна  
трансформація ознак)

SURF – Speeded Up Robust Features (стійкі прискоренні ознаки)

## ВСТУП

Для сучасності характерним є швидкий розвиток методів та засобів отримання, обробки, передачі та розпізнавання об'ємних потоків інформації в реальному часі з необхідною точністю та продуктивністю. При цьому значна увага приділяється обробці та аналізу зображень, адже саме візуальні образи є носієм найбільшого об'єму інформації, яка може бути використана в різних галузях діяльності людини. Зростає актуальність розробки систем, моделей, методів та засобів обробки зображень, які давали б можливість автоматизації та підвищення точності і швидкодії вказаних процесів. На даний момент основними задачами систем комп'ютерного зору є [1–5]:

- ідентифікація об'єкту;
- перевірка наявності об'єкта на зображенні;
- локалізація об'єктів;
- відстеження об'єкта у відеопотоці;
- розпізнавання інформації, що містить об'єкт на зображенні;
- комбінація або суперпозиція задач для комплексних систем.

Вирішення задач розпізнавання образів відкриває нові перспективи для автоматизації багатьох процесів, які до нещодавнього часу могла виконувати тільки людина. Наприклад, процеси комунікації в інтелектуальних системах, діяльність з контролю і управління складними об'єктами, що змінюються в реальному часі. Тому продовжується пошук і реалізація ефективних принципів передачі розпізнавальної функції людини комп'ютеризованими системами.

Теорія розпізнавання образів – галузь знань, яка досліджує методи опису та класифікації об'єктів: явищ, процесів, предметів, сигналів та ситуацій. Під класифікацією розуміють процедуру віднесення сигналів до певних сукупностей, образи яких мають загальні властивості [6–9]. Такі сукупності називаються класами, а пристрої, за допомогою яких виконується

віднесення сигналів – класифікаторами. Можливість розпізнавання спирається на схожість однотипних об'єктів. Звідси виникає поняття класифікації – розбиття всієї множини об'єктів на підмножини-класи, елементи яких мають деякі схожі властивості, що відрізняють їх від елементів інших класів. Таким чином, завданням розпізнавання є віднесення розглянутих об'єктів або явищ по їх опису до потрібних класів.

Загалом виділяють такі підходи до групування об'єктів у класи: класифікація та кластеризація. Для процесу класифікації система має знати кількість класів, до яких потенційно можуть відноситися об'єкти, а також відомими є ознаки, на основі яких можна віднести конкретний об'єкт до того чи іншого класу [3, 5, 8]. Кластеризація – розділення об'єктів на змістовні групи (кластери) при відсутності апріорної інформації про них, а лише базуючись на тій інформації, яка міститься в самих даних про об'єкти. При виконанні кластеризації кількість класів заздалегідь невідома, визначається вона різними математичними методами на основі аналізу параметрів об'єктів або тих даних, які є в розпорядженні технічної системи.

Для того, щоб комп'ютерна система розпізнала зображення, її треба навчити. Існують декілька типів навчання систем розпізнавання образів: навчання з учителем, в якому вхідний образ вважається членом певного класу; навчання без учителя, в якому образ вважається членом невідомого класу [5, 10].

Дана робота присвячена дослідженню та реалізації засобів навчання з учителем, а саме конкурентного навчання Кохонена, у системах класифікації зображень

Один із перспективних напрямків вирішення задачі розпізнавання образів ґрунтується на застосуванні штучних нейронних мереж, як найбільш ефективних по відношенню до класу задач розпізнавання образів [6, 8]. Найбільш складною є задача розпізнавання зображень, які зазнали будь-яких змін (наприклад, зсув, зашумлення образу, поворот, зміна масштабу). Для вирішення даної проблеми необхідно правильно обрати конфігурацію мережі

та спосіб її навчання. Проведений аналіз робіт [2, 3, 11] свідчить, що не існує жодної моделі, яка б була інваріантна до будь-яких спотворень зображень. Тому актуальною є розробка моделі, що дозволяє з великою долею ймовірності віднести об'єкт розпізнавання до визначеного класу об'єктів при малій навчальній вибірці.

Актуальність бакалаврської роботи полягає у тому, що розроблена модель класифікації зображень шляхом використання знань на підставі навчання з учителем підвищує показник ефективності розпізнавання. Застосування навчання з учителем для побудови класифікатора засновано на даних, приналежність яких до заданої системи класів вважається відомою, й це дозволяє ефективніше класифікувати зображення. У задачі класифікації еталонні зображення вважаються заданими [6, 8].

# 1 АНАЛІЗ МЕТОДІВ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

## 1.1 Основні поняття проблеми розпізнавання зображень

У сучасному житті постійно зростають обсяги візуальної інформації, яка потребує формального опису і представлення для її використання в інтелектуальних системах. Одним з ключових понять в автоматичній обробці є поняття розпізнавання об'єктів, що призводить до прискорення роботи систем обробки даних і підвищення їх ефективності. Системи комп'ютерного зору, в яких розпізнавання образів є основною операцією, широко застосовуються при створенні багатьох інтелектуальних систем: діагностичні медичні системи, криміналістична експертиза, інтелектуальний пошук інформації та інтелектуальний аналіз даних тощо. Задача класифікації зображень виникає в таких областях як розпізнавання рукописного тексту, дорожніх знаків, номерів автомобілів тощо. Відмінною особливістю даного завдання є величезна розмірність вхідного простору, що веде до ускладнення розпізнавання і обчислювальних труднощів.

Незважаючи на значний успіх, розпізнавання зображень тільки в деяких галузях здатне за якістю результату порівнятися або перевершити когнітивну функцію сприйняття у людини.

Задача розпізнавання зображень складається з навчання системи та саме розпізнавання образів. Під навчанням системи розпізнавання образів розуміється формування в її пам'яті набору ознак, необхідних для розпізнавання класів вхідних об'єктів. Під час навчання системі показують деякі об'єкти із зазначенням їх приналежності певному класу [4, 6, 12]. У результаті такого процесу система має реагувати певними реакціями на всі об'єкти одного образу та іншими реакціями – на всі об'єкти інших образів. Навчальною вибіркою об'єктів можуть бути будь-які візуальні зображення

(картинки, літери, цифри тощо). Як тільки інтелектуальна система пройшла процес навчання, вона приступає до розпізнавання нових об'єктів.

Всі зображення, які поступають до входу системи розпізнавання, треба представити у певному формалізованому вигляді, щоб їх було зручно аналізувати та потім розпізнавати. Тобто кожен об'єкт, який класифікується або кластеризується системою, представляється у вигляді сукупності його ознак – це так званий образ. Ознака – це числові данні, які є параметром конкретного об'єкта та характеризує ступінь вираження у нього певної властивості. Отже, об'єкт представлений в системі не безпосередньо, а опосередковано – вектором ознак. Таким чином, ознаки об'єкта – це все, що надається автоматичній системі для роботи [3, 5, 13].

Задача розпізнавання образів зводиться до віднесення вхідних даних до певного класу за допомогою виділення їх найбільш інформативних ознак із загальної маси несуттєвих даних. Дану задачу можна записати у вигляді відношення:

$$Z \subseteq I^{\text{in}} \times I^{\text{out}}, \quad (1.1)$$

де  $I^{\text{in}}$  – інформація про вхідну множину об'єктів, яка була розбита на кінцеву кількість класів;

$I^{\text{out}}$  – мітки/індекси класів.

Процес виділення інформативних ознак зображення буде ефективним, якщо користуватися наступними правилами [5, 14]:

- обирати величини з мінімальною дисперсією всередині класу для того, щоб їх розподіли не накладалися один на інший;

- обирати величини з максимальними відстанями між класами (навіть якщо їх дисперсії однакові, то класифікація буде ефективною).

Загалом система розпізнавання образів працює за схемою, що зображена на рисунку 1.1.



Рисунок 1.1 – Функціональна схема системи розпізнавання образів

Задача розпізнавання образів вирішується наступним чином [2, 5]:

Крок 1. Генерування ознак (feature generation) – отримання числових ознак, що характеризують об'єкт, шляхом їх виміру або обчислення.

Крок 2. Вибір ознак (feature selection) – визначення найбільш інформативних ознак, за якими буде здійснюватися класифікація (це може бути як первинні ознаки, так й функції від них).

Крок 3. Побудова класифікатора (classifier construction) – знаходження вирішального правила, а саме функції, яка ставить у відповідність вектору ознак клас даного образу. Кожен з класів – множина об'єктів, які мають спільні властивості – має свою ідентифікуючу мітку. Процес призначення міток класу його об'єктам називається класифікацією.

Крок 4. Оцінка якості класифікації (classifier estimation) – обчислення показників коректності класифікації (точність, помилки першого та другого роду, чутливість, специфічність тощо).

Представимо математичну постановку задачі класифікації зображень [5, 7, 9]. Нехай  $\Omega$  – простір образів;  $\omega \in \Omega$  – образ;  $M = \{1, 2, \dots, m\}$  – номери класів  $\Omega_1, \Omega_2, \dots, \Omega_m$ , таких що  $\Omega_i \cap \Omega_j = \emptyset$ , якщо  $i \neq j$  і  $\bigcup_{i=1}^m \Omega_i = \Omega$ .

Тоді  $g: \Omega \rightarrow M$  – індикаторна функція, що є невідомою.

Також припустимо  $X$  – простір ознак, тобто векторний простір, точками якого є вектори ознак образів. Знайдемо функцію, яка ставить у відповідність образу  $\omega$  його вектор ознак  $x(\omega)$ , тобто  $x: \Omega \rightarrow X$ .

$K_1, K_2, \dots, K_m$  – підмножини простору  $X$ , такі що  $K_i \cap K_j = \emptyset$ , якщо  $i \neq j$  і  $\bigcup_{i=1}^m K_i = X$ .

Тоді функція  $\hat{g}: X \rightarrow M$  – є вирішальним правилом, яке ставить у відповідність вектору ознак номер класу, якому належить даний образ.

Для коректного розпізнавання (класифікації/кластеризації) образів інтелектуальна система має генерувати певні реакції на групи зовнішніх ідентичних сигналів. Навчання системи класифікації виконується трьома основними методами [1–6].

Навчання з учителем (supervised learning) – існує тренувальна вибірка вірно класифікованих об'єктів (які класифікував експерт-вчитель), на основі якої треба створити правило класифікації.

Задача класифікації з учителем полягає у тому, щоб на підставі множини прецедентів  $(g_j, x_j)$ ,  $j = 1, \dots, N$ , яка називається навчальною вибіркою, побудувати вирішальне правило  $\hat{g}$ . Прецедент – це апіорна інформація про клас, до якого належить об'єкт. Прецедентом може бути, наприклад, набір ознак метелика, з яким слід порівнювати набір ознак інших комах, щоб знайти між ними усіх метеликів. Вчителем вважається або сама навчальна вибірка, або той, хто указав значення  $g_j$ .

Навчання без вчителя (unsupervised learning) – необхідно сформулювати правило, яке буде використано для групування об'єктів в кластери. Задача класифікації без вчителя часто називається кластеризацією або кластерним аналізом [6, 12]. В цій задачі вибірка образів  $x_j$ ,  $j = 1, \dots, N$  розбивається на підмножини (кластери), що не перетинаються, та складаються із схожих об'єктів. При цьому необхідно, щоб об'єкти із різних кластерів істотно відрізнялися один від одного.

Навчання з підкріпленням (reinforcement learning) – комп'ютерна система взаємодіє із динамічним навколишнім середовищем, в якому вона

повинна досягти певної мети (наприклад, система автоматичного керування автомобілем).

## 1.2 Класифікація та кластеризація у системах розпізнавання

Процес розпізнавання зображень проходить, по-перше, етап ідентифікації образів [2–5, 7], тобто система має відрізнити певний об'єкт у множині схожих на нього (наприклад, впізнати свою дитину серед інших у дитячому садку).

На другому етапі відбувається класифікація об'єкта, що є найбільш поширеною задачею штучного інтелекту. В результаті рішення задачі класифікації визначаються ознаки, які найліпшим способом характеризують класи об'єктів набору даних, що досліджується. Це може бути, наприклад, задача розпізнавання літер або прийняття рішення про наявність дефекту у деякій технічній деталі.

На сьогодні існує багато методів рішення задачі класифікації зображень: байесовський класифікатор, штучні нейронні мережі, лінійні класифікатори, дерево рішень тощо.

Іноді замість класифікації проводять кластерний аналіз, який полягає в розділенні заданого набору об'єктів на кластери – групи об'єктів, схожі між собою за тим чи іншим критерієм. При цьому класи досліджуваного набору даних заздалегідь не визначені і саме в цьому й полягає відмінність кластеризації від класифікації. Таким чином кластеризація призначена для розбиття послідовності об'єктів на однорідні групи – кластери. Якщо дані вибірки уявити як точки в просторі ознак, то задача кластеризації зводиться до визначення «згущень точок». Кластеризація є описовою процедурою, вона не робить ніяких статистичних висновків, але дає можливість провести розвідувальний аналіз і вивчити структуру даних.

Застосування кластеризації має декілька цілей. Ця операція виявляє кластерну структуру, що сприяє розумінню даних. Крім того, розбиття великого набору даних на групи подібних об'єктів спрощує подальшу обробку виборки та прийняття рішень, застосовуючи до кожного кластера різний метод аналізу (стратегія «розділяй і володарюй»).

Також кластеризація сприяє стисненню даних: при досить великій вихідній вибірці, її можна скоротити, залишивши по одному найбільш типовому представнику від кожного кластера.

Крім того, кластеризація сприяє виявленню новизни, що заключається в виділенні нетипових об'єктів, які не приєднуються до жодного з кластерів.

Вирішити завдання кластеризації можна ймовірностним підходом (наприклад, метод  $k$ -середніх), застосуванням нейронних мереж (наприклад, мереж Кохонена) та ін.

У таблиці 1.1 наведено характерні риси параметрів класифікації та кластеризації.

Таблиця 1.1 – Порівняння класифікації та кластеризації

| Характеристика      | Класифікація  | Кластеризація                                 |
|---------------------|---|---|
| Тип навчання        | навчання з учителем   | навчання без учителя                          |
| Мітки класу         | послідовність образів супроводжується міткою, яка вказує клас, до якого Вони належать | мітки класу невідомі                          |
| Основа класифікації | образи класифікуються, базуючись на навчальну виборку                                 | існує безліч даних для встановлення кластерів |

### 1.3 Визначення ключових точок зображення

Стандартний підхід, яким користуються у системах комп'ютерного зору для представлення зображення, заснований на використанні лише тих пікселів, внесок яких в загальну характеристику буде значним. Тобто

зображення замінюється на набір ключових точок. Ключова або особлива точка – це точка, розміщення якої виділяється на фоні будь-якої іншої точки. Незалежно від того, як зображення змінюється, які афінні або проектні перетворення виконуються з ним, все одно можна знайти ті ж самі ключові точки в цьому зміненому зображенні.

Ключові точки можуть представлятися кутами, ребрами або навіть контуром об'єкта, тому, як правило, обчислення виконуються для деякого околу. Ключові точки повинні мати наступні особливості [5, 15]:

- відмінність – кожна ключова точка має бути унікальною у своєму околу, тобто суттєво відрізнятися від інших;
- інваріантність – детектування ключової точки не має залежити від будь-яких афінних перетворень;
- стабільність – виділення таких особливостей має бути інваріантним до шумів та модифікацій;
- інтерпретація – ключові точки мають виділятися у такий спосіб, щоб їх можна було використовувати для подальшого аналізу відповідностей і отримання на їх основі необхідної інформації.

При цьому процес порівняння зображень розділяється на три етапи [5, 8, 16]. На першому етапі знаходяться множини ключових точок за допомогою детекторів. Детектор – це метод визначення ключової точки, що виділяє її на фоні зображення, а дескриптори повинні забезпечувати інваріантність знаходження відповідності між ключовими точками щодо перетворень зображень. Результатом роботи детектора буде множина координат ключових точок, які на кожному зображенні різні.

На другому етапі відбувається побудова дескрипторів, виходом якого є набір векторів ознак для вихідного набору ключових точок. Дескриптор – це метод, котрий дає змогу вилучати ключові точки обох зображень та порівнювати їх між собою.

Третій етап полягає в порівнянні дескрипторів і пошуку точок, що співпадають на обох зображеннях.

Існує ряд методів виділення дескрипторів ключових точок, такі як BRISK (Binary Robust Invariant Scalable Keypoints), SIFT (Scale Invariant Feature Transform), SURF (Speeded Up Robust Features), ORB (Oriented FAST and Rotated BRIEF), KAZE, AKAZE (Accelerated KAZE). Розрізняють дескриптори границь (Corners) та областей (Blobs) [17–19]. У таблиці 1.2 представлено інформацію про тип та розмір найбільш популярних дескрипторів [20].

Таблиця 1.2 – Характеристики типу та розміру основних дескрипторів

| Назва дескриптора  | Тип     | Розмір            |
|--------------------|---------|-------------------|
| SIFT               | Blobs   | 128 Floats        |
| SURF (128 або 64D) | Blobs   | 128 або 64 Floats |
| ORB                | Corners | 32 Bytes          |
| KAZE               | Blobs   | 128 Floats        |
| AKAZE              | Blobs   | 61 Bytes          |
| BRISK              | Corners | 64 Bytes          |

Дескриптори Corners досліджувану область описують формою її границі, шляхом завдання її характеристик. Це може бути цепні коди (подання границь у виді послідовності відрізків прямих ліній заданої довжини та напрямку), сигнатури (подання границі у виді одномірної функції) та апроксимацію багатокутниками.

Застосування дескрипторів областей доцільно, коли для опису досліджуваних об'єктів достатньо кількох основних дескрипторів, що значно полегшує обчислення. Площа області визначається як число пікселів, які розташовані в ній. Цей дескриптор корисний при зборі інформації про взаємне розташування та форму об'єктів, зображених в одному масштабі.

Дескриптори мають вигляд числових або бінарних векторів. Числовий вектор може описувати структуру або текстуру області особливої точки,

наприклад у методах SIFT та SURF для розрахунку дескрипторів будуються гістограми градієнтів областей навколо ключової точки. При формуванні бінарних дескрипторів описується область навколо особливої точки двійковим рядком, який отриманий як попарне порівняння яскравостей пікселів заданої області, наприклад методи BRISK, ORB.

Результатом роботи детекторів є множина особливих точок, для яких необхідно побудувати математичний опис. Вхідними даними для складання дескриптора є набір особливих точок зображення, які були визначені. Виходом дескриптора є множина векторів ознак для множини особливих точок. Існують окремо детектори особливих точок та дескриптори, які вирішують одночасно два завдання – пошук ключових точок і побудова векторів-описів для них.

Пошук ключових точок зображення здійснюється достатньо просто та швидко (у порівнянні з іншими використовуваними характерними ознаками), тому цей підхід є розповсюдженим. Крім того, на зображеннях не завжди вдається виділити інші характерні риси (наприклад, чіткі контури або області), в той час як локальні особливості в переважній більшості випадків виділити можна.

#### 1.4 Постановка задачі

Незважаючи на досягнуті успіхи розпізнавання зображень у системах комп'ютерного зору, досі не існує ідеального алгоритму, який би гарантував ідентифікацію об'єктів з ефективністю людини. Пошук оптимального виділення ключових точок, їх дескрипторів, а також методів розпізнавання зображень, як і раніше є основним завданням в області обробки зображень. Класифікація зображень з використанням навчання з вчителем на базі нейронних мереж є актуальним завданням в задачах комп'ютерного зору.

У роботі поставлено завдання розробки моделі класифікації зображень на основі деякої вибірки, використовуючи апарат виявлення ключових точок і обчислення дескрипторів методом ORB та подальшої класифікації зображень за допомогою мережі Кохонена. Навчання з учителем дає можливість підвищити результативність методів класифікації зображень, а також надає економію часу класифікації об'єктів.

Об'єктом роботи є методи класифікації зображень у системах комп'ютерного зору.

Метою роботи є підвищення результативності методів класифікації зображень шляхом впровадження засобів навчання з учителем на підставі мережі Кохонена.

Для вирішення поставленого завдання необхідно виконати такі задачі:

- проаналізувати існуючі підходи до класифікації зображень та методи навчання інтелектуальних систем, зокрема навчання з учителем;
- проаналізувати методи отримання дескрипторів ключових точок зображення;
- вивчити алгоритм роботи конкурентних структур (мереж Кохонена) та визначити їх ефективність для класифікації множини дескрипторів;
- реалізувати та проаналізувати результативність комп'ютерної моделі для бази зображень.

## 2 ЗАСОБИ НАВЧАННЯ З УЧИТЕЛЕМ ДЛЯ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

### 2.1 Аналіз властивостей детекторів ключових точок

Найчастіше зображення подається до системи класифікації як набір дескрипторів ключових точок [1, 4], тому що ключові точки є унікальними ознаками, що дозволяють ідентифікувати об'єкт сам з собою або зі схожими групами об'єктів. Існує багато способів, що дозволяють детектувати такі точки. Деякі способи виділяють ключові точки в сусідніх кадрах, деякі через великий проміжок часу і при зміні освітлення, деякі дозволяють знайти ключові точки, які залишаються такими навіть при поворотах об'єкта.

Перший клас. Ключові точки, які є незмінними протягом короткого періоду часу, декількох секунд. Такі точки використовуються для розпізнавання відео, коли потрібно вести об'єкт між сусідніми кадрами або камерами. Це може бути локальні максимуми зображення, кути, точки з максимальною дисперсією, певні градієнти тощо [2, 15, 17].

Другий клас. Ключові точки, які є інваріантними до змін освітлення та незначних рухів об'єкта, наприклад, класифікатор пішохода або класифікатор особи. У якості ключових точок може використовуватися примітиви Хаара, відблиски зображення, інші специфічні функції [18, 19]. До таких точок відносяться точки, знайдені методом гістограм спрямованих градієнтів

Третій клас. Повністю стабільні точки. Існує методи, що забезпечуть повну інваріантність ключових точок – це дескриптори SURF і SIFT. Вони дозволяють знаходити ключові точки навіть при повороті зображення. Розрахунок таких точок здійснюється довше в порівнянні з іншими методами, але досить обмежений час [20, 21].

Базовим методом визначення ключових точок є детектор FAST (Features from Accelerated Segment Test), який відноситься до детекторів кутів, тобто в цьому методі вимірюють зміну яскравості пікселя точки. Метод використовує коло Брезенхема радіуса 3 [5, 12], щоб дізнатися, чи є обрана точка кутом. Кожному пікселю в колі за годинниковою стрілкою присвоюється число від 1 до 16 (рис. 2.1). Якщо набір суміжних пікселів всередині кола яскравіше або темніше, ніж піксель-кандидат, то він класифікується як кут. Таким чином, виконується наступна послідовність кроків.

Крок 1. Обирається точка зображення  $p$ , для якої буде вирішуватися, чи є вона ключовою. Нехай  $I_p$  – яскравість точки. Навколо досліджуваної точки обирається окружність, найчастіше з 16 пікселів.

Крок 2. Точка  $p$  вважається кутом, якщо серед усіх пікселів окружності існує  $n$  пікселів, кожен з яких яскравіший, ніж  $I_p + t$ , або темніший, ніж  $I_p - t$ . Значення порога  $t$  обирається, виходячи із прикладної задачі.

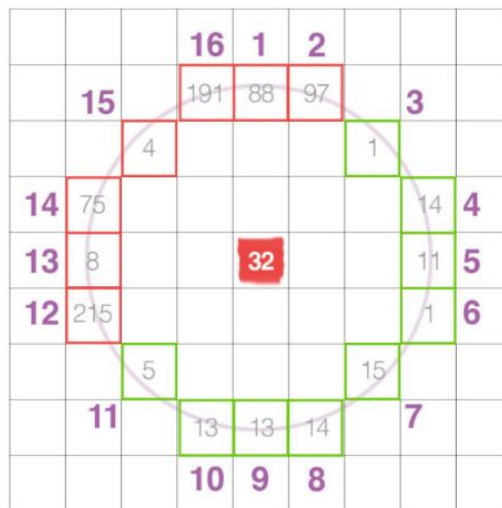


Рисунок 2.1 – Приклад обрання зони навколо ключової точки

Крок 3. Перевіряється інтенсивність точок 1, 5, 9 і 13 з окружності. Якщо хоча б для трьох з них виконується умова Кроку 2, тоді проводиться перевірка решти з 12-ти пікселів. У іншому разі обирається наступна точка і алгоритм повторюється для неї. Представлений алгоритм забезпечує високу швидкодію оброблення даних.

Важливість детектора FAST полягає в його простоті та ефективності обчислення, також він працює швидкіше, ніж багато відомих методів виділення інформативних ознак (наприклад, DoG, SIFT, Harris). FAST вважається високоякісним детектором особливих точок, але все ж не стійкий до шуму і залежить від порогового значення  $t$ .

Метою створення метода BRISK (Binary Robust Invariant Scalable Keypoints) було досягнення високої швидкості роботи, порівнянної з FAST та інваріантності до масштабу [5, 22, 23].

У методі BRISK багатомасштабний простір зображення складається з  $n$  октав  $c_i$  і  $n$  внутрішніх октав  $d_i$ ,  $i = \{0, 1, \dots, n - 1\}$ . Кількість октав  $n$  зазвичай вибирається рівним 4. Октави знаходяться шляхом зменшення масштабу вихідного зображення в два рази (рис. 2.2). Внутрішні октави розташовані між  $c_i$  та  $c_{i+1}$  та представлені у вигляді стиснутого вхідного зображення у  $2^i$  рази.

Пошук спеціальних точок у октавах здійснюється детектором FAST. Нехай  $\epsilon$  вибірка з  $N$  особливих точок. Область навколо кожної особливої точки розбивається на 60 ділянок  $p_i$  (рис. 2.3) та створює множину  $A$ :

$$A = \{(p_i, p_j) \in R^2 \times R^2 \mid i < N \wedge j < i \wedge i, j \in \mathbb{N}\}. \quad (2.1)$$

Множина  $A$  у свою чергу розбивається на підмножини:

$$A_1 = \{(p_i, p_j) \in A \mid \|p_j - p_i\| < \delta_{max}\} \subseteq A, \quad (2.2)$$

$$A_2 = \{(p_i, p_j) \in A \mid \|p_j - p_i\| > \delta_{min}\} \subseteq A, \quad (2.3)$$

де  $\delta_{min} = 13.67b$ ,  $\delta_{max} = 9.75b$ ,  $b$  – розмір особливої точки.

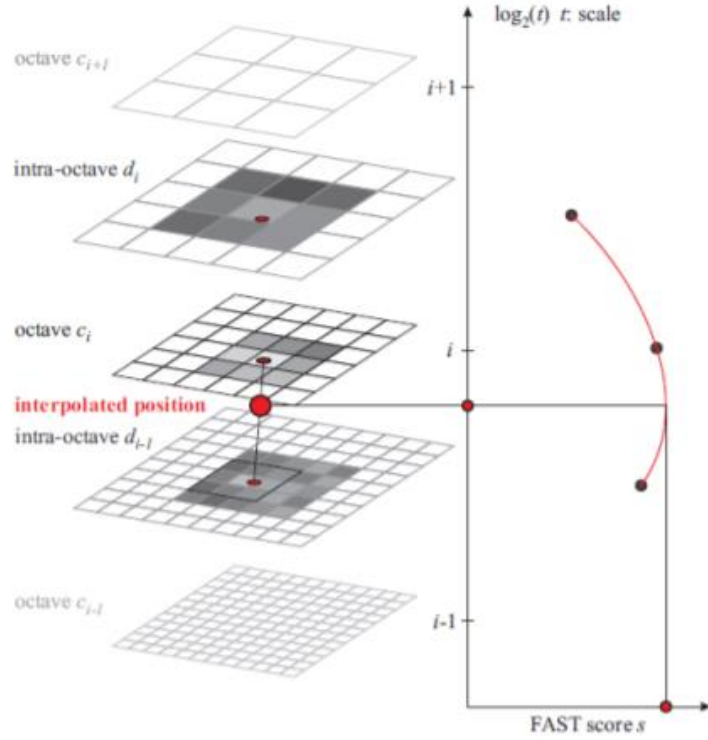


Рисунок 2.2 – Приклад пошуку особливих точок з максимальним значенням  $S$

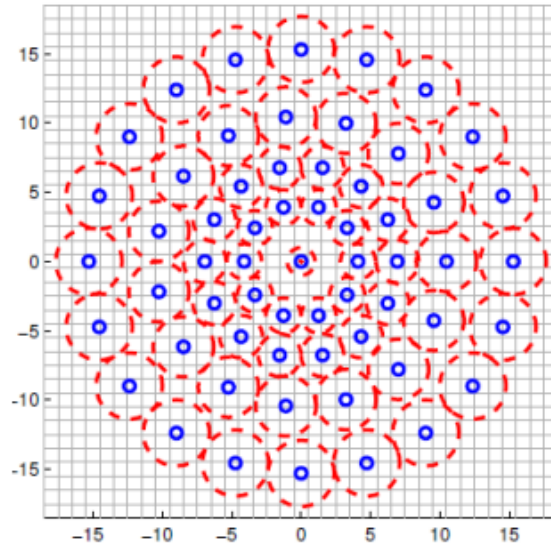


Рисунок 2.3 – Область обчислення дескриптора

Обчислюється середнє значення градієнта множини  $A_2$ :

$$g = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{|A_2|} * \sum_{(p_i, p_j) \in L} \left[ (p_j - p_i) \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\|p_j - p_i\|^2} \right], \quad (2.4)$$

де  $I(p_i, \sigma_i)$  – інтенсивність околу радіусу  $\sigma_i$  точки  $p_i$ .

Дескриптор складається з бінарного рядка довжиною 512 бит, заповненого результатами проведених тестів у множині  $A_1$ :

$$d = \begin{cases} 1, & I(p_j^\alpha, \sigma_j) > I(p_i^\alpha, \sigma_i); \\ 0, & \text{інакше} \end{cases}; \quad \forall (p_i^\alpha, p_j^\alpha) \in A_1, \quad (2.5)$$

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

Детектор SIFT (Scale Invariant Feature Transform) визначає ключові точки за допомогою побудови піраміди гауссіанів та різниць гауссіанів (Difference of Gaussian – DoG) [3, 24] у різних масштабах зображень предмета. Гауссіаном (або зображенням, розмитим гаусовим фільтром) є зображення:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \quad (2.6)$$

де  $L$  – значення гауссіана у точці з координатами  $(x, y)$ ;

$\sigma$  – радіус розмиття;

$G$  – гаусове ядро;

$I(x, y)$  – значення вихідного зображення;

$*$  – операція згортки.

Детектор виділяє область розміром  $16 \times 16$  навколо кожної особливої точки та далі сегментує її на підблоки, відображаючи загалом 128 бінарних значень. Незважаючи на досить велику розмірність дескриптора, обчислювальні витрати залишаються низькими. При цьому дескриптори з меншою розмірністю векторів є не настільки ефективними. У той же час збільшення розмірності дескриптора давала б ще кращі результати, але не суттєво, та й небезпека збільшення чутливості до спотворень і перешкод також може зростати [17, 22].

Наступне рівняння показує згортку різниці двох гауссіанів (розрахованих у різних масштабах) із деяким зображенням  $I(x, y)$ :

$$D(x, y, \sigma) = (G(x, y, \sigma, k\sigma) - G(x, y, \sigma)) * I(x, y). \quad (2.7)$$

SIFT інваріантний щодо зрушень, обертань, масштабу зображень, що не зміщує локальні екстремуми. Однак, основним недоліком методу є висока обчислювальна вартість.

Метод SURF (Speeded up Robust Features) [25] зазвичай застосовується для вирішення двох завдань: пошуку особливих точок на зображенні та створення їх дескрипторів. Крім того, сам пошук ключових точок має володіти інваріантністю, тобто повернутий об'єкт повинен володіти тим же набором особливих точок, що і зразок. Для знаходження ключових точок використовується матриця Гессе. Детермінант матриці Гессе (так званий Гессіан) досягає екстремуму в точках максимальної зміни градієнта яскравості. Для двовимірної функції її детермінант визначається так:

$$H(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}, \quad (2.8)$$

$$\det(H) = \frac{\partial^2 f}{\partial x^2} \frac{\partial^2 f}{\partial y^2} - \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2, \quad (2.9)$$

де  $H$  – матриця Гессе;

$f(x, y)$  — функція зміни градієнта яскравості.

Потім обчислюються напрями найбільшої зміни яскравості та перебираються масштаби матриці Гессе за октавами. Після знаходження ключових точок метод SURF формує їх дескриптори. Дескриптором є набір з 64 (або 128) чисел для кожної ключової точки. Кожна особлива точка є

максимумом гесіану, що гарантує факт того, що в околі цієї точки будуть ділянки з різними градієнтами. Градієнт у точці обчислюється за допомогою фільтрів Хаара. Таким чином, забезпечується відмінність дескрипторів для різних спеціальних точок, унаслідок чого утворюється інваріантність дескриптора щодо обертання.

Таким чином, цей дескриптор інваріантний до перетворень зображення, таких як зміна масштабу і освітленості, на додаток до невеликих змін в точці огляду. Він використовує проміжне представлення для зображення і містить суму значень пікселів зображення в градаціях сірого. Основною перевагою SURF перед SIFT є низька обчислювальна вартість.

В основі детектору ORB (Oriented FAST and Rotated BRIEF) лежить комбінація таких алгоритмів, як детектор FAST та дескриптор BRIEF з деякими покращеннями [4, 7, 18]. Структура алгоритму ORB показує, що він менш вимогливий до обчислювальних ресурсів у порівнянні з іншими алгоритмами. Виграш у швидкості обчислень пояснюється простою процедурою побудови дескрипторів.

Згідно з FAST для кожної точки зображення формується коло деякого радіуса і підраховується число пікселів, що лежать на ньому і мають значення яскравості менше або більше її центру. Ця точка, тобто центр кола, вважається особливою, якщо таких пікселів буде більше 75% від їх кількості. В ORB використовується модифікація FAST-9 (радіус кола дорівнює 9 пікселям), оскільки вона виявилася найбільш ефективною з погляду продуктивності. Для побудови дескриптора отриманих точок використовується модифікація BRIEF, яка інваріантна до повороту за допомогою додаткових перетворень.

Тобто ORB відноситься до локального детектору ключових точок і дескриптор, який надає вектор ознак у вигляді бінарного рядку. Метод залежить від відносно невеликої кількості підрахунків різниці в інтенсивності, щоб представити ділянку зображення у вигляді бінарного вектору. Ідентифікація і дослідження цієї локальної особливості відбувається швидко і з високою

ефективністю. Метод ORB вважається найефективнішим, поки не потрібна інваріантність до великих поворотів, також не була належним чином вирішена проблема нестабільності дескрипторів до змін масштабу.

Детектор KAZE заснований на нормалізованому детермінанті матриці Гессе, який обчислюється на кількох рівнях масштабу. Максимуми реакції детектора вибираються як ознаки за допомогою рухомого вікна.

Дескриптори KAZE стійкі до поворотів зображення, змін масштабу, вони мають більшу відмінність у різних масштабах, несуттєво збільшуючи при цьому час обчислень. Опис об'єкта має властивість інваріантності до обертання шляхом знаходження домінуючої орієнтації в околі навколо кожного виявленого об'єкта [2, 26]. У наступному рівнянні показано стандартну нелінійну формулу дифузії:

$$\frac{dL}{dt} = \text{div} (c(x, y, t), \nabla L), \quad (2.10)$$

де  $c$  – функція провідності;

$\nabla$  – градієнтний оператор;

$L$  – яскравість зображення.

При розробці методу AKAZE – Accelerated-KAZE, намагалися досягти високої швидкодії роботи як детектора, так і дескриптора. При цьому знайдені особливі точки та їхні дескриптори повинні були задовольняти високі показники точності при порівнянні зображень.

Алгоритм AKAZE базується на нелінійній фільтрації дифузії, як KAZE, але його нелінійні масштабні простори будуються за допомогою обчислювальної ефективної структури під назвою Fast Explicit Diffusion (FED). Застосування алгоритму FED на пірамідальній схемі дає можливість побудувати нелінійну багатомасштабну піраміду [2, 4].

Обчислення ключових точок детектором AKAZE засновано на детермінанті матриці Гессе. Властивість інваріантності до поворотів

зображення покращено за допомогою фільтрів Шарра. Максимуми відгуків детектора в просторових місцях вибираються як характерні точки.

У таблиці 2.1 представлено порівняння алгоритмів по ряду параметрів за шкалою від 1 до 5, де 1 – погано, 5 – відмінно [17, 26].

Таблиця 2.1 – Порівняння детекторів

| Метод | Тип     | Інваріантність |         |     |                  | Швидкість роботи           |                            | Якість знайдених точок |
|-------|---------|----------------|---------|-----|------------------|----------------------------|----------------------------|------------------------|
|       |         | Масштаб        | Поворот | Шум | Зміна яскравості | Низька роздільна здатність | Висока роздільна здатність |                        |
| Fast  | Corners | 3              | 5       | 2   | 4                | 5                          | 5                          | 3                      |
| Sift  | Blobs   | 4              | 3       | 2   | 5                | 1                          | 1                          | 4                      |
| Surf  | Blobs   | 5              | 4       | 2   | 4                | 4                          | 3                          | 5                      |
| Brisk | Corners | 4              | 4       | 2   | 5                | 3                          | 5                          | 5                      |

## 2.2 Аналіз методів машинного навчання

У процесі навчання система розпізнавання має отримати здатність реагувати потрібними діями на певні сукупності зовнішніх даних.

Найбільш розповсюдженими у машинному навчанні є алгоритми навчання системи з учителем і без учителя. При першому засобі навчання існує вчитель-експерт, який надає вхідні образи системі, потім порівнює реакції системи з необхідними відповідями та налаштовує ваги мережі, щоб мінімізувати відмінності. Тобто під час навчання з учителем для кожного вхідного набору даних є цільовий вектор-функція, який представляє собою потрібну вихідну реакцію [6]. Така сукупність представляє собою навчальну пару. Загалом система навчається на деякому наборі подібних навчальних пар. Таким чином, системі пропонується вихідний вектор, аналізується вихід

мережі і порівнюється з відповідним цільовим вектором. Відмінність, тобто помилка, знов надається до мережі і ваги переналаштовуються відповідно до заданого алгоритму, який мінімізує похибку. Такий процес навчання має ітераційний характер: навчальні вектори надаються до системи послідовно, обчислюються помилки і ваги перераховуються до тих пір, поки помилка розпізнавання не досягне визначеного рівня.

Навчання без учителя у системах розпізнавання зображень досить часто формулюється як задача кластеризації. Вхідна послідовність об'єктів розбивається на кластери (множини, що не мають однакових образів) так, що кожний кластер містить схожі об'єкти, а різні кластери суттєво відрізняються один від одного. Кластеризація часто використовується в якості допоміжного засобу розв'язування задач класифікації та регресійного аналізу.

У машинному навчанні без учителя є лише вхідні дані і алгоритму необхідно витягти знання з цих даних. Головна проблема машинного навчання без учителя – оцінка корисності інформації, отриманої алгоритмом. Алгоритми машинного навчання без вчителя, як правило, застосовуються до даних, які не містять ніяких міток, таким чином, невідомо, якою має бути правильна відповідь. Тому дуже важко судити про якість роботи моделі. Як наслідок, алгоритми машинного навчання без вчителя часто використовуються в розвідувальних цілях, коли фахівець хоче краще вивчити самі дані.

Деякі алгоритми розв'язування задач класифікації комбінують навчання з учителем та навчання без учителя (наприклад, навчання мереж векторного квантування).

Одним з розповсюджених та простих методів навчання без вчителя є метод кластеризації  $k$ -середніх [27, 28].

Спочатку вибирається число кластерів  $k$ . Після вибору значення  $k$  алгоритм  $k$ -середніх відбирає точки, які будуть представляти центри кластерів. Потім для кожної точки даних обчислюється його евклідова відстань до кожного центру кластера. Кожна точка призначається

найближчому центру кластера. Алгоритм обчислює центроїди – центри важкості кластерів. Кожен центр ваги – це вектор, елементи якого являють собою середні значення характеристик, обчислені по всіх точках кластера. Центр кластера зміщується в його центр ваги. Точки заново призначаються найближчому центру кластера. Етапи зміни центрів кластерів і перепризначення точок ітеративно повторюються до тих пір, поки границі кластерів і розташування центроїдів не перестануть змінюватися, тобто на кожній ітерації в кожен кластер будуть потрапляти одні і ті ж точки даних.

Навіть якщо є відомою правильна кількість кластерів для конкретного набору даних, алгоритм *k*-середніх не завжди може виділити їх. Кожен кластер визначається виключно його центром, це означає, що кожен кластер має опуклу форму. В результаті цього алгоритм *k*-середніх може описати відносно прості форми [14, 27]. Крім того, алгоритм *k*-середніх передбачає, що всі кластери в певному сенсі мають однаковий «діаметр», він завжди проводить кордон між кластерами так, щоб він проходив точно посередині між центрами кластерів.

Одним з розповсюджених класифікаторів є нейронні мережі, одиничний елемент яких, нейрон, емулює роботу біологічного нейрона. Класичні нейронні мережі є рішенням для завдання розпізнавання з учителем [10–12, 29]. Незважаючи на складність отримання знань з нейрон-мережевої системи, вони успішно застосовуються для численних завдань класифікації, управління, прогнозування.

Для вирішення задач класифікації/кластеризації перспективним є застосування штучних нейронних мереж. Можна виділити ряд переваг використання нейронних мереж, у тому числі мереж Кохонена [6, 15, 29]:

- побудова ефективної моделі розпізнавання зображень на підставі нейронних мереж навіть в умовах неповноти даних;
- штучні нейронні мережі легко працюють в розподілених системах з великою паралелізацією, що обумовлено їх природою;

– вибір найбільш інформативних ознак об'єктів є оптимальним, тому що штучні нейронні мережі підлаштовують свої вагові коефіцієнти в залежності від вихідних даних.

### 2.3 Моделі навчання з учителем

У процесі навчання з учителем випробувана система примусово навчається за допомогою наявної множини прикладів «стимул-реакція» з метою визначення «реакції» для «стимулів», які не належать до наявної множини прикладів. Між входами та еталонними виходами (стимул-реакція) може існувати деяка залежність, але вона не відома. Відома лише кінцева сукупність прецедентів пар «стимул-реакція», тобто навчальна вибірка. На основі цих даних потрібно відновити залежність, тобто алгоритм, здатний для будь-якого об'єкта видати досить точну відповідь. Для вимірювання точності відповідей може вводитися функціонал якості [6, 15, 30].

Формально задача навчання може бути сформульована наступним чином: існує деяка сукупність стимулів  $x$  і реакцій на стимули  $y$ , необхідно визначити залежність між  $y$  та  $x$  таку, що в межах припустимої помилки  $E$  буде справедливим

$$f(x) - E \leq y \leq f(x) + E. \quad (2.11)$$

У випадку навчання нейронної мережі з вчителем на вхід подаються дані з деякого навчального набору. Потім сигнали підсумовуються і активують приховані нейрони. Цей процес повторюється шар за шаром, поки не буде досягнутий вихідний шар. Сигнали нейронів можна трактувати як відповідь на якесь питання, наприклад, про приналежність зображення якогось класу. Якщо це відповідь правильна, то переходять до наступного зразка, в іншому випадку відбувається процес зворотного поширення

помилки. Це можна трактувати так: учитель поставив свою оцінку за цю відповідь, і потрібно вивчити нове правило. Якщо параметри нейронної мережі підбрані вірно, то, обробивши достатню кількість навчальних зразків на вході, нейронна мережа стає здатна класифікувати незнайомі об'єкти.

Мережа Кохонена – це одна з різновидів нейронних мереж, які використовують системи розпізнавання зображень для класифікації об'єктів. Вона з'явилася у 1982 р., як спеціальний вид нейронних мереж для обробки зображень і звука. Основною метою мереж Кохонена є перетворення складних багатомірних даних в більш просту структуру малої розмірності. Але виявилось, що вони гарно пристосовані для кластерного аналізу, коли треба виявити приховані закономірності у великих масивах даних. Мережі Кохонена завжди в алгоритм навчання нейронів вводять інформацію про його розміщення у мережі.

Основний принцип роботи мережі Кохонена, як і всіх штучних нейронних мереж, заключається в імітуванні властивостей мозку людини [30–32]. Кора головного мозку людини нагадує звичайний лист, який згорнутий складками. Тобто, можна зробити висновок, що вона характеризується деякими топологічними особливостями (ділянки, відповідальні за близькі частини тіла, примикають один до одного і все зображення людського тіла відображається на цю двовимірну поверхню). У багатьох моделях вирішальну роль відіграють зв'язки між нейронами, які визначаються ваговими коефіцієнтами і вказують місце нейрона в мережі. Однак в біологічних системах, наприклад, у мозку, сусідні нейрони, отримуючи аналогічні входні сигнали, реагують на них подібним чином, тобто групуються, утворюючи деякі області. Оскільки при обробленні багатовимірного входного образу здійснюється його проектування на область меншої розмірності зі збереженням його топології, часто подібні мережі називають картами (self-organizing feature map). У таких мережах суттєвим є облік взаємного розташування нейронів одного шару [30, 33].

Вхідні нейрони утворюють вхідний шар мережі, якій містить по одному нейрону для кожного вхідного поля (рис. 2.4). Як і у звичайній нейронній мережі, вхідні нейрони не беруть участі у процесі навчання. Їх задачею є передача значень вхідних полів початкової вибірки на нейрони вихідного шару. Кожен зв'язок між нейронами має певну вагу, яка в процесі ініціалізації встановлюється випадковим чином в інтервалі  $[0; 1]$ . Процес навчання полягає у настроюванні ваг. На відміну від більшості нейронних мереж [34], мережа Кохонена не має прихованих шарів: дані з вхідного шару передаються безпосередньо на вихідний, нейрони якого впорядковані у одномірну або двохмірну решітку прямокутної або шестикутної форми. Значення кожної ознаки надходять через вхідні нейрони на нейрони вихідного шару.

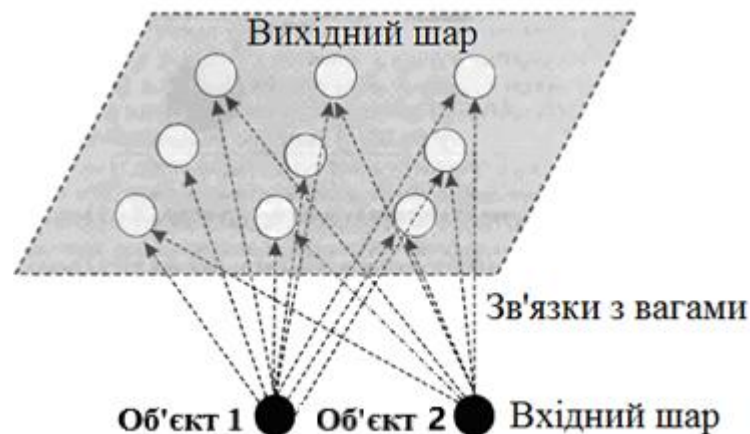


Рисунок 2.4 – Загальна структура мережі Кохонена

У процесі навчання і функціонування мережа Кохонена виконує три процедури [30]:

Крок 1. Конкуренція (competition). Вихідні нейрони конкурують між собою за те, щоб вектори їх ваг виявились як можна ближче до вектора ознак об'єкта. Вихідний нейрон, вектор ваг якого має найменшу відстань до вектора ознак об'єкта, оголошується переможцем.

Крок 2. Об'єднання (cooperation). Нейрон-переможець стає центром деякої групи сусідніх нейронів. У мережі Кохонена усі нейрони такого сусідства називають нагородженими правом підстроювання ваг. Тобто, не дивлячись на те, що нейрони у вихідному шарі не з'єднуються безпосередньо, вони мають схожі набори ваг завдяки сусідству з нейроном-переможцем.

Крок 3. Підстроювання ваг (adaptation). Нейрони, що є сусідами нейрона-переможця, беруть участь у підстроюванні ваг, тобто у навчанні.

Розглянемо набір з  $m$  значень полів  $n$ -ого запису початкової вибірки, який є вхідним вектором  $X_n = (x_{n1}, x_{n2}, \dots, x_{nm})^T$  і поточний вектор ваг  $j$ -го вихідного нейрону  $W_j = (w_{1j}, w_{2j}, \dots, w_{mj})^T$ .

У навчанні по Кохонену нейрони, які є сусідами нейрону-переможця підстроюють свої ваги, використовуючи комбінацію вхідних векторів і поточних векторів ваг:

$$w_{ij}^{\text{нове}} = w_{ij}^{\text{поточне}} + \alpha (x_{ni} - w_{ij}^{\text{поточне}}), \quad (2.12)$$

де  $0 < \alpha \leq 1$  – коефіцієнт швидкості навчання.

Швидкість навчання мережі має бути функцією від кількості ітерацій, яка зменшується. Тому процес навчання мережі можна розділити на дві фази: грубого та точного підстроювання [6]. При ініціалізації мережі початкові ваги нейронів призначаються випадково, якщо відсутні апріорні знання про характер розподілу у початковій вибірці. Також, при ініціалізації задаються початкова швидкість навчання  $\alpha$  і радіус навчання  $R$ , тобто кількість нейронів, які будуть вважатись сусідами для нейрона-переможця і підстроювати свої ваги разом із ним. Радіус є максимальним на початку навчання і зменшується по мірі навчання.

Алгоритм Кохонена складається з таких кроків [30].

Крок 1. *Ініціалізація*. Для нейронів мережі встановлюються початкові ваги, а також задаються початкові швидкість навчання  $\alpha$  і радіус навчання  $R$ .

Крок 2. *Збудження*. На вхідний шар подається вектор впливу  $X_n$ , що містить значення вхідних полів запису навчальної вибірки.

Крок 3. *Конкуренція*. Для кожного вихідного нейрону обчислюється відстань  $D(W_j, X_n)$  між векторами ваг усіх нейронів вихідного шару і вектором вхідного впливу. Наприклад, для евклідової відстані отримаємо:

$$D(W_j, X_n) = \sqrt{\sum_i (w_{ij} - x_{ni})^2}. \quad (2.13)$$

Переможцем стає нейрон  $j$ , для якого така відстань виявиться найменшою.

Крок 4. *Об'єднання*. Визначаються усі нейрони, розташовані в межах радіуса навчання відносно нейрона-переможця.

Крок 5. *Підстроювання*. Здійснюється підстроювання ваг нейронів в межах радіуса навчання відповідно до формули (2.12).

При цьому ваги нейронів-сусідів нейрона-переможця, підстроюються у бік його вектора ваг, як це наведено на рисунку 2.5, де координати вхідного вектору помічені знаком «+», а вигляд мережі після модифікації штриховими лініями.

Крок 6. *Корекція*. Змінюється радіус і параметр швидкості навчання відповідно із заданим законом.

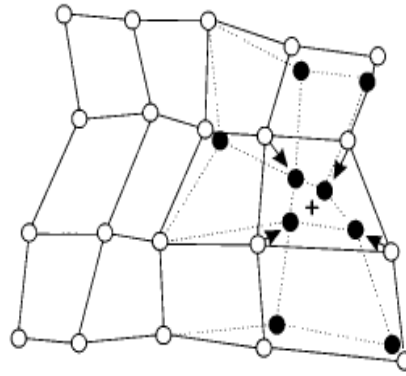


Рисунок 2.5 – Підстроювання ваг нейронів в межах радіуса навчання

Припустимо, що кожному з класів значень  $x$  поставлено у відповідність кілька кодуючих векторів. Тоді  $x$  буде вважатися таким, що відноситься до того ж класу, що  $i$  найближчий до нього кодуючий вектор  $m_j$ . Нехай співвідношенням

$$v = \arg \min_{j=1, \dots, J} \rho(x, m_j) \quad (2.14)$$

визначається індекс  $v$  для вектора  $m_j$ , найближчого до вектора  $x$ . Зауважимо, що індекс переможця  $v$  залежить від  $x$  і від списку векторів  $m_j$ .

Модель (2.14) називають конкурентним навчанням Кохонена [30].

Значення для  $m_j$ , які мінімізують частку помилкових класифікацій, можуть бути знайдені в ході базового процесу

$$m_v(t+1) = m_v(t) + \gamma \alpha(t) [x(t) - m_v(t)], \quad (2.15)$$

$$m_j(t+1) = m_j(t) \quad \forall j \neq v, \quad (2.16)$$

де  $t$  – дискретний момент часу навчання,  $t = 0, 1, 2, \dots$ ;

$\alpha(t)$  – коефіцієнт швидкості навчання,  $0 < \alpha(t) < 1$ ;

$\gamma = +1$ , якщо  $x$  та  $m_v$  належать одному класу;

$\gamma = -1$ , якщо  $x$  та  $m_v$  належать різним класам.

Рекомендується, щоб на початковому етапі процесу навчання значення  $\alpha(t)$  було досить малим, наприклад, меншим 0,1. Але загалом вид залежності  $\alpha(t)$  немає вирішального значення.

Карти Кохонена або Карти ознак, що самоорганізуються (self organizing map), будуються на основі нейронних мереж Кохонена і призначені для візуалізації багатовимірних об'єктів на двохвимірній карті, де відстані між об'єктами відповідають відстаням між їх векторами, а самі значення ознак відображаються різними кольорами і відтінками [30]. Можна провести аналогію карт Кохонена зі звичайною географічною картою, яка лініями відображає кордони країн (кластерів), а різними кольорами – рельєф поверхні (значення параметрів).

Якщо розмірність простору ознак набору даних дорівнює 2, то і вектори ваг нейронів будуть двохкомпонентними і відображення результатів кластеризації на двохвимірній карті не складе проблем [35]. Відобразити багатовимірний простір на площину можна за допомогою проєкцій, але при цьому неминуче порушується топологічна подібність (рис. 2.6 а): значно відділені один від іншого об'єкти у початковому просторі ознак, можуть розташуватись поряд один з іншим на карті, як це видно з наведеного рисунку.

Рисунок 2.6 б) показує, що класифікація двох груп об'єктів, що відрізняються шириною і висотою, при вилученні з розгляду довжини, буде некоректною. Це пов'язано з тим, що об'єкти стануть практично нерозрізненими.

У картах Кохонена кількість нейронів набагато перевищує кількість кластерів. Карта складається з комірок прямокутної або шестикутної форми, де відстані між центрами суміжних комірок є однаковими (рис. 2.7). Кожній комірці відповідає нейрон мережі Кохонена [30].

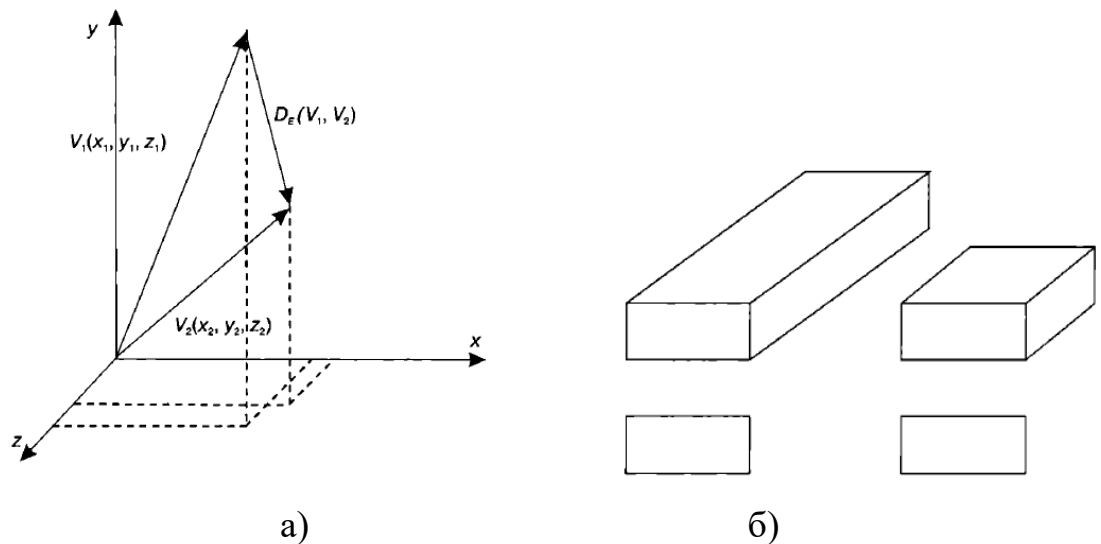


Рисунок 2.6 – Приклад труднощів класифікації деяких об'єктів:

- а) порушення топологічної подібності
- б) схожість об'єктів при вилученні деяких ознак

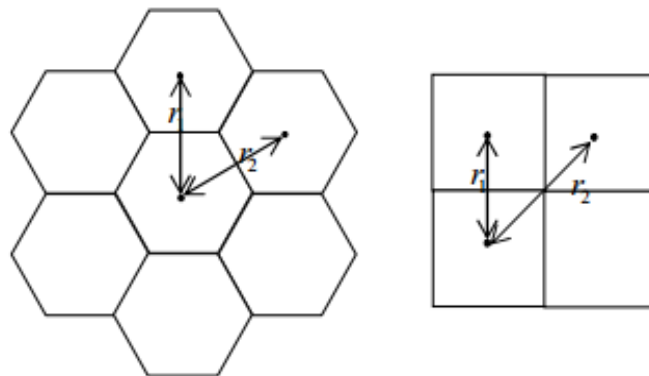


Рисунок 2.7 – Структура карт Кохонена

Навчання нейронів карти здійснюється так само, як і навчання нейронів мережі Кохонена. Об'єкти, вектори ознак яких близькі між собою, потрапляють в одну комірку карти або в комірки, розташовані поруч. Хоча відстань між об'єктами надає уяву про ступень їх подібності або відмінності, але важливою є інформація і про те, у чому саме проявляється ця подібність або відмінність, за якими ознаками об'єкти розрізняються більше, а за якими менше тощо. Відповідь на ці питання і надає спеціальна розкраска, яка виконує функцію третього виміру (рис. 2.8).

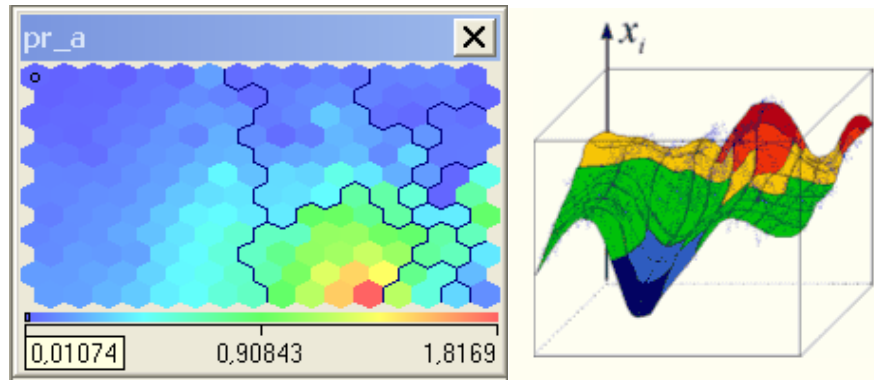


Рисунок 2.8 – Представлення карт Кохонена

На одній карті можна використати розкраску лише за однією ознакою [30]. Для візуалізації кількох ознак треба будувати окремі карти. Наприклад, для об'єктів з двома ознаками, для отримання інформації щодо значень цих ознак, треба будувати дві карти. Отримавши усю сукупність карт для кожної з ознак, можна наочно порівняти значення кожної ознаки даного кластера зі значеннями кожної з ознак інших кластерів. У кожному комірку в загальному випадку потрапляє кілька об'єктів. Тому обчислюється або середнє значення параметра об'єктів кожної комірки або мінімальне чи максимальне значення. Якщо в комірку не потрапив жоден об'єкт (комірці відповідає мертвий нейрон), то за значення комірки береться вага нейрона, відповідного даному параметру.

У результаті роботи алгоритму отримуємо декілька карт.

Карта входів нейронів:

– ваги нейронів підлаштовуються під значення вхідних змінних і відображають їх внутрішню структуру. Для кожного входу формується своя карта, розфарбована у відповідності зі значенням конкретної ваги нейрона. Розглянемо фрагмент карти, що складається з карт трьох входів;

– при аналізі даних використовують кілька карт входів;

– на одній з карт виділяють область певного кольору – це означає, що відповідні вхідні приклади мають приблизно однакове значення відповідного

входу. Кольоровий розподіл нейронів з цієї області аналізується на інших картах для визначення схожих або відмінних характеристик.

Виділимо на першій карті об'єкти (рис. 2.9), розташовані в правому нижньому кутку, які мають найбільші значення. Можна помітити, що ці ж об'єкти мають найбільші значення показника, зображеного на третій карті.

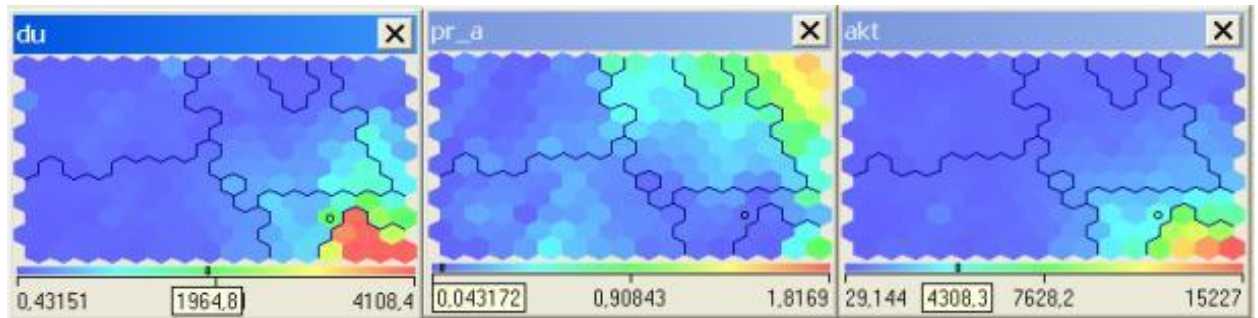


Рисунок 2.9 – Карти входів нейронів

Карта виходів нейронів відображає взаємне розташування досліджуваних вхідних даних. Нейрони з однаковими значеннями виходів утворюють кластери – замкнуті області на карті.

Спеціальні карти – карта кластерів, матриця відстаней, матриця щільності попадання і інші карти, які характеризують кластери, отримані в результаті навчання мережі Кохонена.

Але у карт Кохонена є ряд недоліків і обмежень, що примушують у деяких випадках обережно ставитись до самого процесу побудови карт, і до інтерпретації його результатів [11, 26, 36, 37].

Самі по собі карти задачу кластеризації не вирішують, а лише дозволяють висунути гіпотези про наявність кластерної структури і залежностей в наборі даних. Висунуті гіпотези потрібно підтверджувати іншими методами, тому що вони можуть виявитися помилковими.

Недоліком є і евристичний характер методу. Якщо навчати карту кілька разів, завжди будуть отримані несхожі результати, оскільки встановлювані випадковим чином початкові ваги нейронів будуть різні і

навчання кожен раз буде йти трохи інакше. Крім того, може виникнути така проблема, як «мертві» нейрони – нейрони, ваги яких були ініційовані таким чином, що їх вектори опинилися в тій частині простору ознак, де відсутні або майже відсутні вектори об'єктів. При цьому обробку вхідних даних буде здійснювати надто мала кількість нейронів, яких може не вистачити для якісної кластеризації.

### 3 ПРОГРАМНЕ МОДЕЛЮВАННЯ МЕТОДУ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

#### 3.1 Особливості програмного моделювання методу ORB

Для реалізації запропонованого методу класифікації необхідно всі зображення представити у вигляді масиву дескрипторів. Для детектування ключових точок у роботі використовується метод ORB.

Як було описано у розділі 2, для пошуку кутових точок методом ORB виділяється коло Брезенхема (рис. 3.1). Точка  $p$  з яскравістю  $I_p$  вважається особливою, якщо не менше трьох пікселів виконують умову:

$$I_i < I_p - t \text{ OR } I_i > I_p + t \text{ для } i = 1, \dots, 4. \quad (3.1)$$

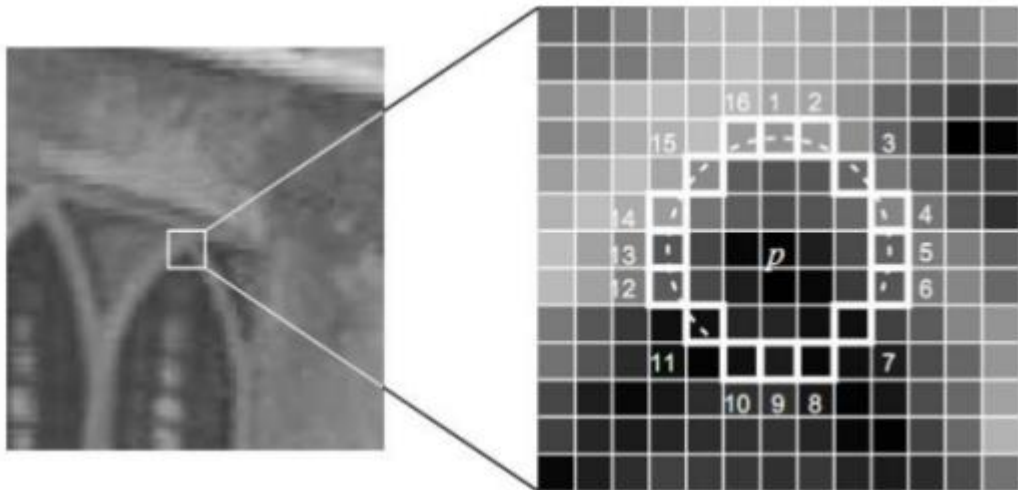


Рисунок 3.1 – Область потенційної особливої точки FAST детектора

За замовчуванням в алгоритмі ORB максимальна кількість точок не більше 500, якщо їх більше, то до них застосовується детектор кутів Харріса [15, 32], щоб виключити найменш значущі. Щоб отримати  $N$  ключових точок, спочатку використовується низький поріг для того, щоб отримати більше  $N$

точок, потім вони впорядковуються за допомогою метрики Харріса і вибираються перші  $N$  точок.

Для інваріантності до масштабування в алгоритмі ORB застосовується піраміда Гаусса [24, 38, 39]. Початкове стиснуте зображення  $c_0$  з заданим кроком і визначає октаву  $c_i$ . Введення параметра кутової орієнтації дозволяє домогтися стійкості детектування при обертанні об'єкта. Він заснований на напрямках градієнта яскравості щодо центру точки, напрямом з найбільшою інтенсивністю призначається орієнтацією особливої точки  $p$ .

BRIEF дескриптор – це вектор довжиною 256, що складається з результатів бінарних тестів навколо особливої точки. Для цього зображення розбивається на патчі (окремі перекриті ділянки) розміром  $31 \times 31$  пікселів. Із патча деяким чином вибирається множина пар пікселів  $(x, y)$ , де  $x, y$  – області  $5 \times 5$  пікселів. Для них будується набір бінарних тестів:

$$\tau(I, x, y) := \begin{cases} 1: I_x < I_y \\ 0: I_x \geq I_y \end{cases} \quad (3.2)$$

де  $I_x$  – середня яскравість обраної області.

Область обчислення дескриптора орієнтується по ключовій точці  $p$  для досягнення інваріантності. Матрицю  $S$  розмірністю  $2 \times n$  формують усі  $n = 256$  наборів  $x_i$  та  $y_i$ . Далі за допомогою матриці повороту  $R_p$  орієнтується відповідно до кута  $p$ , де  $S_p = R_p S$ . А сам вектор дескриптора записується як:

$$g_n(I, p) := f_n(I) \mid (x_i, y_i) \in S_p, \quad (3.3)$$

$$f_n(I) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(I; x_i, y_i). \quad (3.4)$$

Детектор ORB розраховує дескриптори для кожного зображення у вигляді бінарного вектору з 256 значеннями, які і будуть використані у системах розпізнавання.

### 3.2 Модель модифікації мережі Кохонена для навчання з учителем

Розглянемо формальний опис застосування конкурентних структур (мереж Кохонена) в задачах класифікації зображень, використовуючи навчання з вчителем. Візьмемо за основу векторне квантування, коли для кожного класу, що задається еталоном, попередньо синтезується його деякий центр, а класифікація здійснюється шляхом віднесення будь-якого вектору з аналізованого простору до класу з найбільш близьким центром [30, 40, 41].

Нехай  $W = \{x \mid x \in R^n\}$ ,  $W \subseteq R^n$  – простір векторів-дескрипторів зображень. Послідовність дескрипторів еталонів  $Z \subset W$  задана у вигляді множини

$$Z = \{Z^j\}_{j=1}^J, \quad s_j = \text{card } Z^j, \quad s = \text{card } Z = \sum s_j. \quad (3.5)$$

Застосуємо наступний алгоритм навчання мережі.

Крок 1. Побудуємо матрицю  $M = \{m_j\}_{j=1}^J$  ваг, рядками якої є сформовані вектора  $m_j$  – центрів класів так, що  $m_j = x_i^j$ ,  $x_i^j \in Z^j$ , де  $i$  – номер довільного вектора з класу зразків  $Z^j$ .

Крок 2. Сформуємо елементи навчальної множини у вигляді сукупності  $Z = \{Z^j\}_{j=1}^J$  всіх  $s$  елементів бази еталонів.

Крок 3. Виберемо поточний елемент  $z \in Z$ , для кожного  $j \in [1, 2, \dots, J]$  обчислимо відстань до центрів кожного класу і визначимо номер класу  $\nu$  нейрона-переможця за формулою (2.14). Будемо обчислювати  $\rho(z, m_j)$  як евкідову метрику за формулою (2.13).

Крок 4. Обчислимо зміни ваг для нейронів вихідного шару мережі

$$\Delta m_j = h(j, \nu, t) + \alpha(z - m_j), \quad (3.6)$$

де  $\alpha$  – коефіцієнт швидкості навчання, що задається;

$h(j, v, t)$  – значення функції околу для нейрона  $j$  в момент часу навчання  $t$ , зазвичай

$$h(j, v, t) = e^{-\frac{\rho(j,v)}{\sigma(t)}} \quad (3.7)$$

визначають у вигляді гаусової функції, а радіус  $\sigma(t) = \frac{1}{e^{t-2}}$  околу зменшують зі збільшенням параметра  $t$  часу навчання,  $t = 1, \dots, s$ .

Крок 5. Коректуємо матрицю ваг  $M = M + \Delta M$  на кроці  $t$ .

Крок 6. Продовжуємо навчання (Кроки 3-5) до завершення списку  $Z$ .

Крок 7. Перевіряємо виконання критерію зупинки навчання. Це може бути величина помилки кластеризації. При необхідності (невиконання умови зупинки) продовжуємо навчання з Кроку 2.

Ітераційний алгоритм сходиться до локального мінімуму помилки  $E$ . Значення помилки кластеризації оцінимо за сумою квадратів відхилень від центрів сформованих кластерів:

$$E = \sum_{j=1}^J \sum_{v=1}^{s(j)} \rho^2(x_v, m_j), \quad (3.8)$$

де  $s(j)$  – потужність  $j$ -го кластеру.

Мінімізація критерію (3.8) сприяє тому, щоб квантування найкращим чином відповідало аналізованим даним.

Дослідники [8, 30, 42] відзначають, що нормування даних (приведення норми вектора до одиниці) при застосуванні мереж Кохонена прискорює збіжність мережі до локального мінімуму помилки. Для мереж з незначним числом нейронів (2-3) застосування нормування вважається обов'язковим. Якщо обробляються нормовані дані, то нейрони мережі стають нормованими в ході навчання. У даній ситуації ми використовуємо ненормовані дані,

прагнути забезпечити достатню відстань нейронів між собою з метою забезпечення результативного поділу класів.

Відповідно до методу векторного квантування для сформованого набору з  $j$  кодових векторів  $M = \{m_j\}_{j=1}^J$ ,  $m_j \in R^n$  апроксимація довільного вектора  $x \in W$  конкурентним способом означає визначення номера  $v$  найближчого до нього (зазвичай в евклідовій метриці  $\rho(x, m_j)$ ) вектора  $m_v \in M$  в просторі кодуєчих векторів за формулою (2.14). Суть навчання для системи векторного квантування полягає в формуванні комплексу  $M$  на основі навчальної множини  $W$  відповідно до оптимуму деякого критерію.

Навчання з учителем для мережі Кохонена полягає у тому, що для кожного з векторів вхідних зображень знаходиться найближчий кодуєчий вектор («переможець забирає все») [30, 43]. Після цього даний кодуєчий вектор перераховується. Вектори центрів змінюються за формулою:

$$m_v(t+1) = m_v(t) + \alpha(t) [x(t) - m_v(t)], \quad (3.9)$$

де  $\alpha \in [1; 1/t; 2/t; \dots]$ ,  $t$  – порядковий номер кроку навчання, що відповідає порядковому номеру елемента навчальної вибірки.

Переналаштування центрів класів у мережах Кохонена, що використовують навчання з учителем за алгоритмом векторного квантування, відбувається за формулами (2.15), (2.16).

Помилка кластеризації обчислюється за формулою (3.8).

### 3.3 Програмна реалізація моделі класифікації та її тестування

Модель класифікації зображень на підставі навчання з учителем була нами програмно реалізована за допомогою мови Golang. Для тестування роботи програми створили послідовність зображень світових брендів. У якості еталонної вибірки обрали логотипи, які представлені на рисунку 3.2.

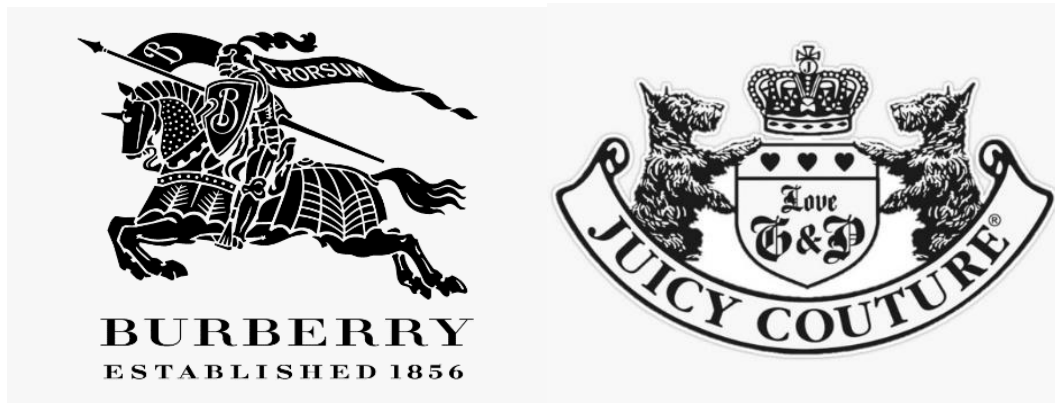


Рисунок 3.2 – Еталонні зображення

Потім знайшли їх ключові точки, що зображені на рисунку 3.3.

Дескриптори знайдених ключових точок мають вигляд бінарного вектора розміром 256, кодуються у вигляді вектора з 32 цілими значеннями (обсяг 8 біт, значення 0 ... 255), а не в бітах.

Для кожного еталонного зображення створений опис у вигляді множини з 500 векторів-дескрипторів методом ORB за допомогою формул (3.3), (3.4). Кожен дескриптор є бінарним 256-бітним вектором, який був представлений у вигляді 32-розмірного вектора з цілими значеннями.



Рисунок 3.3 – Ключові точки зображення

Для еталонних зображень (рис. 3.2) ми розрахували центри кластерів, за якими й будемо навчати мережу Кохонена та робити класифікацію. В результаті обробки для кожного зразка відповідно до процедури Кохонена або з співвідношенням (2.14) отримуємо кластерне уявлення  $[Z^i] = (h_1^i, \dots, h_j^i)$  еталона, де  $h_j^i$  – цілі числа. Воно відповідає розподілу елементів множини  $Z^i$  за класами еталонів. У результаті отримана матриця розподілу ключових точок за класами у вигляді таблиці 3.1, у останньому стовпці представлені результати обчислення помилки за формулою (3.8).

Таблиця 3.1 – Кластеризація еталонної вибірки зображень

| Назва бренду    | 1 клас | 2 клас | 3 клас | E     |
|-----------------|--------|--------|--------|-------|
| Versace_1       | 208    | 116    | 176    | 0.484 |
| Juicy Couture_1 | 139    | 202    | 159    | 0.493 |
| Burberry        | 142    | 159    | 199    | 0.491 |

За результатами роботи програми можна бачити, що обрані зображення були правильно класифіковані з середньою помилкою класифікації 0,489, що є задовільним. Максимум на діагоналі матриці дещо перевищує інші значення у рядку. Також була розрахована манхеттенська відстань між кластер ними поданнями образів для отриманого розбиття за формулою:

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|. \quad (3.10)$$

Ця міра може застосовуватися для обчислення відстані між об'єктами, описаними кількісними, якісними і дихотомічними ознаками. Її використання доцільно, коли ознаки однорідні за смисловим навантаженням і однаково важливі для розв'язуваної задачі. Саме вона найбільш підходить для роботи з даними дескрипторів, так як у цій моделі обробляються дійсні дані.

Результати розрахунку представлені у таблиці 3.2.

Таблиця 3.2 – Манхеттенська відстань між еталонними зображеннями

|                        | <b>Versace_1</b> | <b>Juicy Couture_1</b> | <b>Burberry</b> |
|------------------------|------------------|------------------------|-----------------|
| <b>Versace_1</b>       | 0                | 92                     | 126             |
| <b>Juicy Couture_1</b> | 92               | 0                      | 66              |
| <b>Burberry</b>        | 126              | 66                     | 0               |

Таким чином, ми бачимо, що класифікація дескрипторів зображень ефективна та тільки зображення Juicy Couture\_1 та Burberry мають деяку незначну схожість.

На наступному етапі застосували алгоритм векторного квантування до послідовності зображень, які представлені у Додатку А. Для кожного зображення були знайдені дескриптори ключових точок та проведена їх класифікація відносно еталонної вибірки за алгоритмом конкурентного навчання Кохонена, користуючись правилом (2.15)-(2.16). Навчання мережі Кохонена протестована на 100 ітераціях. Результати класифікації зображень із застосуванням навчання з учителем представлені у таблиці 3.3.

Таблиця 3.3 – Результати класифікації зображень за допомогою навчання з учителем

| Назва бренду    | 1 клас | 2 клас | 3 клас | Е     |
|-----------------|--------|--------|--------|-------|
| Versace_2       | 268    | 99     | 133    | 0.464 |
| Juicy Couture_2 | 85     | 262    | 153    | 0.473 |
| Armani          | 156    | 133    | 211    | 0.496 |

Аналізуючи отримані результати, видно, що помилка класифікації майже не змінилася (середнє значення 0,477), ефективність розпізнавання образів є задовільною, так як максимум на діагоналі суттєво перевищує інші значення у рядку. Також була розрахована манхеттенська відстань для отриманого розбиття векторів (табл. 3.4). Результати додатково свідчать про високу ефективність класифікації зображень за запропонованою моделлю.

Таблиця 3.4 – Манхеттенська відстань між зображеннями

|                        | <b>Versace_2</b> | <b>Juicy Couture_2</b> | <b>Armani</b> |
|------------------------|------------------|------------------------|---------------|
| <b>Versace_2</b>       | 0                | 366                    | 114           |
| <b>Juicy Couture_2</b> | 366              | 0                      | 258           |
| <b>Armani</b>          | 114              | 258                    | 0             |

Можна зробити висновок, що запропонована модель класифікації зображень з використанням навчання з учителем є більш ефективною у порівнянні із традиційною моделлю без навчання.

Лістинг програми, яка реалізує запропоновану модель класифікації зображень на підставі навчання з учителем, представлена у Додатку Б.

## ВИСНОВКИ

У рамках кваліфікаційної роботи було розроблено ефективний метод класифікації зображень на підставі засобів навчання з учителем за моделлю конкурентного навчання Кохонена.

У роботі проведено огляд методів розпізнавання зображень, розглянуто задачі класифікації та кластеризації образів. Проаналізовані методи виявлення дескрипторів ключових точок зображення. Для визначення ключових точок та побудови дескрипторів використовувався метод ORB, який представляє дескриптор у вигляді 256-бітного бінарного вектору. Даний метод був обраний завдяки своїй невибагливості до обчислювальних ресурсів та швидкодії у обчисленні дескрипторів ключових точок.

Для навчання системи розпізнавання зображень було застосовано навчання з учителем, а саме методику конкурентного навчання Кохонена, що значно прискорює розпізнавання образів. Розроблений програмний застосунок для класифікації зображень на прикладі образів світових брендів. Проаналізовані результати роботи програми на основі розрахованих помилок класифікації та відстаней між отриманими поданнями еталонних класів. Основною перевагою розроблених методів над відомими є вища результативність розпізнавання зображень.

Результати роботи апробовано у вигляді двох тез доповідей на Міжнародному молодіжному форумі «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У ХХІ СТОЛІТТІ» [23, 44].

## ПЕРЕЛІК ДжЕРЕЛ ПОСИЛАННЯ

1. Путьятин, С. П., Гороховатський, В. О., & Матат, О. О. (2006). Методи та алгоритми комп'ютерного зору: навч. посібник.
2. Шапиро Л., Стокман Дж. (2006). Компьютерное зрение.
3. Гороховатський, В. О., & Творошенко, І. С. (2021). Методи інтелектуального аналізу та оброблення даних: навч. посібник.
4. Гороховатский, В.А. & Полякова, Т.В. (2018) Применение пространственных структур признаков для классификации изображений в компьютерном зрении (монография). *Харьков: ФОП Панов А.Н.*
5. Клетте, Рейнхард (2019). Компьютерное зрение, *Springer*.
6. Hal Daumé III, (2015). A course in Machine Learning, *TODO*.
7. Gorokhovatsky, V.A. (2014). Structural analysis and intellectual data processing in computer vision. *Kharkiv: SMIT*.
8. Гороховатский, В.А. (2003) Распознавание изображений в условиях неполной информации, *Харків: ХНУРЕ*.
9. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ. 124 с.
10. Бринк, Хенрик, Ричардс, Джозеф, Феверолф, Марк (2017). Машинное обучение, *Питер*.
11. Аггарвал, Ч. (2020). Нейронные сети и глубокое обучение. Диалектика.
12. Мюллер, Андреас, & Гвидо, Сара. (2017). Введение в машинное обучение с помощью Python.
13. Gorokhovatskyi V.A. (2018) Image Classification Methods in the Space of Descriptions in the Form of a Set of the Key Point Descriptors, *Telecommunications and Radio Engineering*, 77 (9), pp. 787-797.

14. Gorokhovatskiy, V. A. (2016). Efficient Estimation of Visual Object Relevance during Recognition through their Vector Descriptions, *Telecommunications and Radio Engineering*, 75(14).
15. Gorokhovatskiy, V., Vasylychenko, A., Manko, K., & Ponomarenko, R. (2018). Дослідження модифікацій методу встановлення релевантності зображень об'єктів за описами у вигляді множини дескрипторів ключових точок. *Системи управління, навігації та зв'язку*, 5(51), С. 74-78.
16. Gorokhovatskiy V., Gadetska S., Ponomarenko R. (2020). Recognition of Visual Objects Based on Statistical Distributions for Blocks of Structural Description of Image. Proceedings of the XV International Scientific Conference «Intellectual Systems of Decision Making and Problems of Computational Intelligence» (ISDMCI'2019), Ukraine, pp. 501-512.
17. Gorokhovatskiy V.A., Gorokhovatskiy A.V., Peredrii Ye.O. (2018) Hashing of Structural Descriptions at Building of the Class Image Descriptor, Computing of Relevance and Classification of the Visual Objects, *Telecommunications and Radio Engineering*, 77(13), pp. 1159-1168.
18. Gorokhovatskiy O., Gorokhovatskiy V., Peredrii O. (2018) Analysis of Application of Cluster Descriptions in Space of Characteristic Image Features, *MDPI*, pp. 52-62.
19. Гороховатський В.О., Гадецька С.В., Стяглик Н.І. (2019) Вивчення статистичних властивостей моделі блочного подання для множини дескрипторів ключових точок зображень, *Радіоелектроніка, інформатика, управління*, №2, с. 100-107.
20. Бубенчиков, М.А. (2016). Сравнительный анализ методов нахождения особых точек на изображении: выпускная квалификационная работа бакалавра. СПбГУ. URL: <https://nauchkor.ru/pubs/sravnitelnyy-analiz-metodov-nahozhdeniya-osobyh-tochek-na-izobrazhenii-587d363a5f1be77c40d589ec> (дата звернення 23.04.2022).
21. Gorokhovatskiy, V.O. and Gadetska, S.V., (2019) Determination of Relevance of Visual Object Images by Application of Statistical Analysis of

Regarding Fragment Representation of their Descriptions, *Telecommunications and Radio Engineering*, 78 (3), pp. 211-220.

22. S. Leutenegger, M. Chli, R. Siegwart (2011) BRISK: Binary Robust Invariant Scalable Keypoints, *Computer Vision (ICCV)*, pp. 2548-2555.

23. Пронюк, О.Д. (2021). Аналіз властивостей методу BRISK. 25 Міжнародний молодіжний форум Радіоелектроніка та молодь у XXI столітті. 36. Матеріалів форуму. Т.7., Харків: ХНУРЕ, 22-23.

24. Rasmussen, C. E. & Williams, C. K. I. (2006). Gaussian Processes for Machine Learning. *MIT Press*.

25. Bay H., Ess A., Tuytelaars T., Gool L.V. (2008). Speeded-up robust features (SURF), *Computer Vision and Image Understanding*. Academic Press., Vol. 110. No. 3, pp. 346-359.

26. Гороховатський В.О., Пупченко Д.В., Солодченко К.Г. (2018) Аналіз властивостей, характеристик та результатів застосування новітніх детекторів для визначення особливих точок зображення. *Системи управління, навігації та зв'язку*, Вип. 1, С. 93-98.

27. Coates, Adam and Andrew, Y. Ng. (2012). Learning Feature Representations with K-means, Stanford University, URL: [https://cs.stanford.edu/~acoates/papers/coatesng\\_nntot2012.pdf](https://cs.stanford.edu/~acoates/papers/coatesng_nntot2012.pdf) (дата звернення 24.04.2022).

28. В.А. Гороховатский, Ю.Н. Ересько, Е.П. Путятин, В.И. Стрельченко. (1990) Локализация объектов на изображениях визуальных сцен, *Автометрия*, №6. с. 3-7.

29. Руденко, О. Г. & Бодянский, Е. В. (2002). Основы теории искусственных нейронных сетей, *Харьков: ТЕЛЕТЕХ*.

30. Кохонен, Тойво (2017). Самоорганизующиеся карты, *Бином*.

31. Гороховатський В.О., Гадецька С.В., Стяглик Н.І., Власенко Н.В. (2020) Класифікація зображень на підставі ансамблю статистичних розподілів за класами еталонів для компонентів структурного опису, *Радіоелектроніка, інформатика, управління*, №4, с. 85-94.

32. Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski (2011) ORB: an efficient alternative to SIFT or SURF, *Computer Vision (ICCV), IEEE International Conference on IEEE*, pp. 2564-2571.
33. Гороховатский В.А., Путятин Е.П., Столяров В.С. (2017) Исследование результативности структурных методов классификации изображений с применением кластерной модели данных, *Радиоэлектроника, информатика, управление, №3 (42)*, с. 78-85.
34. Гороховатский В.А., Передрий Е.О. (2009) Корреляционные методы распознавания изображений путем голосования систем фрагментов, *Радиоелектроніка. Інформатика. Управління, №1(20)*, с.74-81.
35. Gorokhovatsky V.A., Putyatin Y. P. (2009) Image Likelihood Measures of the Basis of the Set of Conformities, *Telecommunications and Radio Engineering*, 68 (9), pp. 763-778.
36. Gorokhovatskyi, V., Rusakova, N., Tvoroshenko, I. (2020) The application of image analysis methods and predicate logic in applied problems of magnetic monitoring, *Telecommunications and Radio Engineering*, 79 (20), pp. 1801-1811.
37. M. A. Ahmad, V. Gorokhovatskyi, I. Tvoroshenko, N. Vlasenko, S. K. Mustafa (2021) The Research of Image Classification Methods Based on the Introducing Cluster Representation Parameters for the Structural Description, *International Journal of Engineering Trends and Technology*, 69(10), pp. 186-192.
38. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Al-Dhaifallah M., (2022) Classification of Images Based on a System of Hierarchical Features, *Computers, Materials & Continua*, 72(1), pp. 1785-1797.
39. Gorokhovatskyi V., Putyatin Y., Gorokhovatskyi O, Peredrii O. (2018). Quantization of the Space of Structural Image Features as a Way to Increase Recognition Performance. The Second IEEE International Conference on DataStream Mining & Processing, Lviv, Ukraine, pp. 464-467.

40. Gorokhovatskyi V.A., Zamula A.A. (2016) Employment of Intelligent Technologies in Multiparametric Control Systems. Telecommunications and Radio Engineering. Vol. 75, No 19, p. 1775-1785.
41. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, International Journal of Engineering and Information Systems, 6(2), pp. 40-48.
42. Tvoroshenko, I.S., Gorokhovatsky, V.O. (2019) Modification of the branch and bound method to determine the extremes of membership functions in fuzzy intelligent systems. Telecommunications and Radio Engineering, 78 (20), pp. 1857-1868.
43. Gadetska, S.V., Gorokhovatskyi, V. O., Stiahlyk, N. I., Vlasenko, N.V. (2021) Statistical data analysis tools in image classification methods based on the description as a set of binary descriptors of key points. Radio Electronics, Computer Science, Control, №4, pp. 58-68.
44. Пронюк, О.Д. (2022, подано до друку). Застосування мережі Кохонена в задачах розпізнавання зображень. 26 Міжнародний молодіжний форум Радіоелектроніка та молодь у ХХІ столітті. Зб. Матеріалів форуму. – Харків: ХНУРЕ.