

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

МОДЕЛЮВАННЯ МЕТОДУ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ ІЗ
ВИКОРИСТАННЯМ ЗАСОБІВ САМОНАВЧАННЯ
(тема)

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

Клінов В.О.
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

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

Допускається до захисту

Зав. кафедри _____
(підпис)

Кобилін О.А.
(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Клінову В'ячеславу Олексійовичу
(прізвище, ім'я, по батькові)1. Тема роботи Моделювання методу класифікації зображень із використанням засобів самонавчання

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 науково-методична та науково-технічна література, теоретичні відомості про методи знаходження дескрипторів ключових точок, дані інтернет-мережі, бібліотека комп'ютерного зору з відкритим кодом OpenCV, відкритий набір зображень автомобілів Stanford Cars Dataset, інтерактивне середовище розробки PyCharm 2020.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Огляд методів розпізнавання образів та класифікації зображень.2. Аналіз методів пошуку ключових точок та їх дескрипторів на зображенні.3. Математична модель мережі Кохонена для задач кластеризації та класифікації.4. Розроблення застосунку для класифікації зображень автомобілей мережею Кохонена.5. Порівняння якості роботи мережі Кохонена та методу класифікації без навчання.

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, мета роботи, постановка задачі, тестові зображення, перспективи подальшої роботи.

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

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

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів роботи | Терміни виконання етапів роботи | Примітка |
|-------|---|---------------------------------|----------|
| 1 | Отримання завдання на кваліфікаційну роботу | 18.04.2022 | |
| 2 | Аналіз завдання, підбір літератури | 18.04.22-21.04.22 | |
| 3 | Аналіз літератури з досліджуваної проблеми | 22.04.22-25.04.22 | |
| 4 | Аналіз технічних засобів | 26.04.22-30.04.22 | |
| 5 | Розробка методу | 01.05.22-14.05.22 | |
| 6 | Програмна реалізація | 15.05.22-23.05.22 | |
| 7 | Оформлення пояснювальної записки | 24.05.22-26.05.22 | |
| 8 | Перевірка на плагіат | 27.05.22 | |
| 9 | Рецензування | 28.05.22 | |
| 10 | Підготовка презентації та доповіді | 29.05.22-30.05.22 | |
| 11 | Занесення роботи в електронний архів | 31.05.22 | |
| 12 | Попередній захист кваліфікаційної роботи | 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: 62 с., 5 табл., 17 рис., 2 дод., 36 джерел.

КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, ДЕСКРИПТОРИ КЛЮЧОВИХ ТОЧОК, ДЕТЕКТОР ORB, ШТУЧНА НЕЙРОННА МЕРЕЖА, МЕРЕЖА КОХОНЕНА, МЕТРИКА ХЕММІНГА, ЕВКЛІДОВА ВІДСТАНЬ, ШВИДКОДІЯ КЛАСИФІКАЦІЇ.

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

Метою роботи є розробка методу класифікації зображень у системах комп'ютерного зору із застосуванням технології самонавчання. Опис зображення подається у вигляді множини дескрипторів ключових точок. Як метод формування множини дескрипторів застосовано детектор ORB.

Класифікація зображення здійснюється на підставі процедури голосування дескрипторів вхідного зображення та на множині опису еталонів. Досліджено вплив застосування нейронної мережі Кохонена на результат класифікації, проведено порівняння розробленого методу за швидкодією та результативністю із методом без впровадження навчання.

Одержані результати можуть бути використані у сферах застосування технологій комп'ютерного зору, аналізу даних та зображень.

IMAGE CLASSIFICATION, KEY POINT DESCRIPTORS, ORB DETECTOR, ARTIFICIAL NEURAL NETWORK, KOHONEN NETWORK, HAMMING'S METRIC, EUCLIDEAN DISTANCE, CLASSIFICATION SPEED.

The object of research is classifiers of images in computer vision systems.

The aim of the research is to develop a method of image classification in computer vision systems using self-learning technology. The description of the image is presented in the form of a set of descriptors of key points. The ORB detector is used as a method of forming the set of descriptors.

The classification of the image is carried out on the basis of the procedure of voting of the descriptors of the input image and on the set of description of standards. The influence of the application of the Kohonen neural network on the result of classification is investigated, the comparison of the developed method on speed and efficiency with the method without introduction of training is carried out.

The results can be used in the areas of computer vision, data and image analysis.

ЗМІСТ

| | |
|--|--|
| Перелік умовних позначень, символів, одиниць, скорочень і термінів | 6 |
| Вступ..... | 7 |
| 1 Огляд основних методів класифікації зображень..... | 9 |
| 1.1 Основні поняття та методи теорії розпізнавання образів..... | 10 |
| 1.2 Поняття ключових точок та їх дескрипторів | 15 |
| 1.3 Аналіз детекторів ключових точок | 18 |
| 1.4 Огляд сучасних алгоритмів розпізнавання | 23 |
| 1.4.1 AdaBoost..... | 23 |
| 1.4.2 Класифікація методом нечітких правил | 26 |
| 1.4.3 Штучні нейронні мережі | 28 |
| 1.5 Постановка задачі | 30 |
| 2 Моделі нейронних мереж для аналізу зображень..... | 31 |
| 2.1 Моделі нейронів та методи їх навчання | 31 |
| 2.2 Базові моделі мереж із самоорганізацією..... | 37 |
| 2.3 Мережа Кохонена | 39 |
| 2.4 Оцінювання ефективності класифікації зображень | 42 |
| 3 Комп'ютерне моделювання методу навчання та класифікації зображень . | 44 |
| 3.1 Обґрунтування вибору середовища програмної реалізації | 44 |
| 3.2 Програмна реалізація..... | 46 |
| 3.3 Тестування розробленої моделі..... | 48 |
| Висновки | 53 |
| Перелік джерел посилання | 55 |
| Додаток А Тестові зображення..... | Ошибка! Закладка не определена. |
| Додаток Б Результати комп'ютерного моделювання..... | Ошибка! Закладка не определена. |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

КТ – ключова точка

КЗ – комп'ютерний зір

ХО – характерні ознаки

ORB – Oriented FAST and rotated BRIEF

АС – Accuracy

RC – Recall

FAST – Features from Accelerated Segment Test

BRIEF – Binary Robust Independent Elementary Features

SIFT – Scale Invariant Feature Transform

SURF – Speeded Up Robust Features

SOM – Self-Organizing Map, самоорганізуюча карта

WTA – Winner-Takes-All, переможець отримує все

WTM – Winner-Takes-Most, переможець отримує більше

ВСТУП

До основних тенденцій розвитку інформаційних систем та кібернетики в цілому на початку XXI століття можна виразити як біологізація та гібридизація [1-5]. Біологізація полягає в побудування та дослідженні моделей поведінки складних об'єктів та способів управління ними на основі імітації механізмів, реалізованих природою в живих істотах. Яскравими прикладами можна вважати штучний інтелект (ШІ), генетичні та еволюційні алгоритми, робототехніці. Натомість гібридизація складається з використання різноманітних методів для обробки інформації про один і той же об'єкт, задля досягнення максимальної подібності моделі об'єкта до оригіналу. Зважаючи на це, технології штучного інтелекту все більш серйозно розглядаються в якості базису для створення швидкісних обчислювальних систем паралельної обробки інформації [3, 5].

Одним з активно розвиваючих напрямків ШІ є штучні нейронні мережі (ШНМ), що працюють за принципом біологічних нейронних мереж і вдають із себе систему взаємодіючих між собою штучних нейронів. Серед основних галузей застосування ШНМ можна виділити: класифікація зображень, обробка звукової та текстової інформації, прогнозування, прийняття рішень, оптимізація, аналіз даних. Існує ряд методів та запропоновано багато алгоритмів для вирішення задачі класифікації зображень [6], проте усі ці ідеї часто поступаються у точності результату або у швидкодії штучним нейронним мережам.

Унікальний ефект стосовно задачі класифікації надають мережі, які мають здатність до самоорганізації – нейронні мережі Кохонена, вони забезпечують топологічне упорядкування вхідного простору образів. На відміну від інших методів, топологічне упорядкування класів зберігає подібність вхідних образів, що є особливо корисним при класифікації багатьох класів. Налаштування вагових коефіцієнтів мережі Кохонена

здійснюється методом конкурентного навчання з досить високою швидкістю оброблення [7-13].

Класифікація зображень застосовується при автоматизації, ідентифікації видів, медицині, навігації, роботизації, системах розпізнавання тексту, виявленні подій, задачах взаємодії між людиною та комп'ютером, моделюванні. Тому на цей час комп'ютерний зір, а також пов'язані з ним технології програмної та апаратної обробки, є одним з перспективніших напрямків в світі комп'ютерних технологій. Постає питання створення алгоритмів, спрямованих на вирішення нагальних завдань класифікації, які працюють максимально точно, з мінімумом помилок та з високою швидкістю [10].

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

Актуальність роботи полягає у розвиненні методу класифікації зображень на основі апарату локальної класифікації дескрипторів ключових точок зображення, впроваджуючи навчання нейронної мережі Кохонена для даних описів еталонних зображень як множини дескрипторів.

1 ОГЛЯД ОСНОВНИХ МЕТОДІВ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

Цифрова обробка зображень наразі в значній мірі присутня майже в будь-якій області техніки. Однією з основних задач, що постає перед сучасними інформаційними системами, є задачі розпізнавання об'єктів у системах технічного зору, а саме слів та тексту взагалі, цифр, жестів, розпізнавання за фото, відбитками пальців, сітківкою ока, визначення класу, до якого відноситься об'єкт на зображенні [11].

Серед існуючих методів вирішення проблем аналізу візуальних об'єктів мало тих, що в достатній мірі дозволяють вирішити багато важливих задач розпізнавання. Це пов'язано з наявністю шуму серед даних, з викривленням структурних частин об'єктів під впливом геометричних перетворень, з просторовими локальними перешкодами. Традиційні математичні моделі обробки графічної інформації не враховують зовнішній вплив перелічених перешкод.

Швидкісний розвиток оптичних та розрахункових технологій, зростання інтересу до проблем розпізнавання образів спонукає до розширення функціональності та підвищення результативності сучасних систем. До цього сприяє розвиток таких дисциплін як статистика, лінійна алгебра та аналітична геометрія, біологія, теорія передачі інформації, дослідження операцій. До теперішнього часу запропоновано чимало підходів до розпізнавання. В основному це підхід з позицій теорії рішень, або дискримінантний підхід [14]. Пізніше почали інтенсивно розвиватися інтелектуальні системи, призначені для опису форми аналізованих зображень або виділення з них метричної або семантичної інформації. Аналіз семантики є вершиною ієрархічної процедури обробки зображень, складність якої пов'язана з тим, що спочатку ця процедура вирішує нечітку задачу.

1.1 Основні поняття та методи теорії розпізнавання образів

Термін «розпізнавання» часто використовують поряд з терміном «класифікація», але вони не є повністю взаємо замінюваними. Класифікація – це процес призначення міток об’єктів відповідно до деякого опису властивостей об’єктів [4, 15]. В такому разі класифікатором є пристрій або алгоритм, який отримує в якості вхідних даних опис об’єкту та видає в якості результату мітку класу:

$$\hat{C}: X \rightarrow C, \quad (1.1)$$

де $C = \{C_1, C_2, \dots, C_k\}$ – скінченна та зазвичай невелика множина міток класів, які не перетинаються;

X – множина об’єктів.

Кожен з наявних об’єктів може бути представлений деяким екземпляром x_i із простору ознак R^n , де $n > 0$. В даній формулі $\hat{c}(x)$ – є оцінкою істинної, але невідомої функції $c(x)$, значення якої ми намагаємося апроксимувати.

Класифікатор призначає номер класу об’єктам спочатку заданого набору $\{x_1, x_2, \dots, x_m\}$ вже класифікованих екземплярів для навчання (набір для тренування), а потім на наступному етапі об’єктам, створеним на етапі застосування [4, 5, 15]:

– загальний класифікатор призначає номери класів $1, 2, \dots, k$ для $k > 1$ класів і 0 для «некласифікованих»;

– бінарний класифікатор призначає номери класів -1 та $+1$ у випадках, коли нас цікавить лише те, чи відбулася конкретна подія (наприклад, «на зображенні є автомобіль»), визначена позначкою $+1$.

На теперішній час базовими є такі класифікатори, як лінійний класифікатор, логістична регресія, машини опорних векторів, дерева рішень та випадкові ліси, класифікатор Байєса [4, 6, 15, 16]. Поширеними підходами

класифікації є: за найближчим середнім значенням, за найбільшою апостеріорною ймовірністю та за допомогою нейронних мереж.

Розпізнавання образів аналізує вхідні дані та намагається визначити закономірності. У той час як дослідницьке розпізнавання шаблонів спрямоване на ідентифікацію шаблонів даних загалом, описове розпізнавання шаблонів починає категоризувати виявлені шаблони [7, 17-19]. Отже, розпізнавання образів має справу з обома цими сценаріями, і застосовуються різні методи розпізнавання образів залежно від варіанту використання та форми даних.

Вхідними даними для розпізнавання образів можуть бути слова або тексти, зображення або аудіофайли. Отже, розпізнавання образів є ширшим порівняно з комп'ютерним зором, яке зосереджується на розпізнаванні зображень.

Враховуючи шаблон (образ), його розпізнавання та класифікація можуть складатися з одного з наступних трьох завдань:

- контрольована класифікація, яка визначає вхідний шаблон як член попередньо визначеного класу (описовий підхід);
- неконтрольована класифікація, яка призначає вхідний шаблон до невизначеного досі класу (дослідницький підхід).

Також можливе використання комбінованих підходів до навчання, в яких можна поєднувати контрольовану класифікацію зі стратегіями неконтрольованого аналізу даних, наприклад, для зменшення розмірності даних шляхом об'єднання окремих об'єктів у кластери або для отримання додаткових статистичних показників [8-11, 15].

Підходи до майже всіх наукових областей та зокрема в розпізнаванні образів можна розрізнити за двома площинами – так звані Платонова та Аристотелева точки зору, а також внутрішні (дедуктивні) та зовнішні (індуктивні) спостереження (рис. 1.1) [12].

Платонова точка зору декларує, що існують деякі основні ідеї, закономірності, знання, які використовуються протягом тривалого часу і

поступово адаптуються до нових спостережень (феноменів). Тобто існує основа теорія, яка може бути стратифікована на ряд гіпотез, які можна перевірити завдяки ряду спостережень.

Натомість підхід Аристотеля уникає спекуляцій на великих теоріях, які можуть виходити за рамки фактів. Саме спостереження є основою, на яких дослідник будує свої знання. На їх основі виявляються закономірності та закономірності, які використовуються для формулювання деяких попередніх гіпотез.

Індуктивний підхід сприймає об'єкти спостереження як вони є в реальному світі, завдяки сенсорам, тоді як дедуктивний підхід описує об'єкт через ряд характеристик або вектор ознак.



Рисунок 1.1 – Підходи до розпізнавання образів

Історично склалося два основних підходи, які використовуються для задач розпізнавання – статистичне та синтаксичне розпізнавання шаблонів.

При статистичному розпізнаванні образи групується відповідно до їх ознак, а кількість ознак визначає як шаблон буде розглядатися у d -вимірному просторі. Ці елементи підібрані таким чином, щоб різні екземпляри займали простір без перетинання.

Метод працює так, що за вибраними атрибутами складаються кластери. Машина навчається та адаптується, а потім використовує визначені шаблони для подальшої обробки та навчання. Метою статистичного розпізнавання є вибір атрибутів, які дозволяють розділяти вектори образів до різних категорій у просторі d -вимірних ознак [10, 13, 17, 20, 21]. До структурного розпізнавання можна віднести класифікацію зображень.

Другий підхід – це синтаксичне (або структурне) розпізнавання образів (рис. 1.2). Він використовується для проблем розпізнавання, які включають складні шаблони, що розглядаються з ієрархічної точки зору. Цей підхід спирається на примітивні підшаблони (наприклад, літери алфавіту). Паттерн описується залежно від того, як примітиви взаємодіють один з одним. Прикладом такої взаємодії є те, як вони складаються в слова та речення. Наведені навчальні зразки показують, як розробляються граматичні правила та як згодом речення будуть розпізнаватися. До синтаксичного розпізнавання можна віднести обробку природної мови (natural language processing) [4, 22].

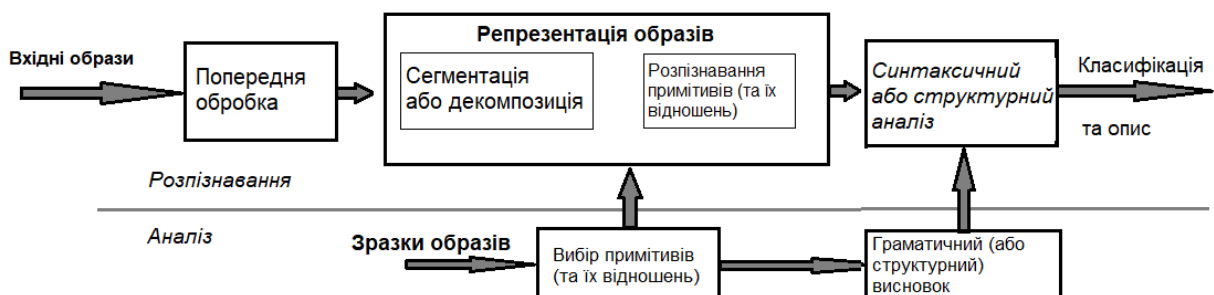


Рисунок 1.2 – Блок-схема системи синтаксичного розпізнавання образів

Додатково до основних можна назвати гібридне розпізнавання та підхід, заснований на технології штучних нейронних мереж (ШНМ), іменованій як розпізнавання нейронних шаблонів. Ці підходи є більш універсальними, тому все частіше використовується в технологіях обробки даних.

Одним з основних понять теорії розпізнавання є образ. Образ – модель явища, процесу або об'єкту абстрактного чи реального світу, яка відображає властивості об'єкту, що необхідно розпізнати за ознаками.

І для теорії, і для практики важливим є питання про те, яке представлення або спосіб кодування об'єкта потрібно застосовувати в процесі розпізнавання, як визначити важливі ознаки. Вирішенням можуть стати технології, засновані на аналізі особливостей об'єкта (зокрема зображення в задачах комп'ютерного зору) в окремих точках. На основі аналізу властивостей кінцевого числа характерних ознак (ХО) можна вирішити практичні завдання будь-якої складності. В такому разі зображення розглядається як точка в багатовимірному просторі або вектор ознак [15, 20, 23, 24].

Розпізнавання зображень на основі системи ХО має багато переваг, серед них: пришвидшення роботи алгоритму обробки зображень, скорочення розмірності простору ознак, універсальна можливість структурного аналізу об'єктів, стійкість до впливу фонових впливів та помилковості об'єктів.

Ознака розпізнавання – характеристика деякої властивості аналізованого об'єкту. Множина ознак розпізнавання, що формують вектор-реалізацію образу, характеризує образ.

Зображення робляться при різному освітленні, під різними кутами огляду, за різних погодних умов зовнішньої сцени тощо. Тобто зображення схильні до спотворення під впливом багатьох чинників, через що в аналізі зовнішніх сцен часто запрошують певні типи інваріантності до операцій [5].

Припустимо що є вхідні зображення I , на сцені $S \in \mathfrak{S}$ та камеру (так званий процес зображення) C . Для зображень $I = C(S)$, визначено процедуру

аналізу \mathcal{X} , що відображає зображення I у деякий векторний вихід $R(I) = r$, результат. Прикладом такої процедури може бути детектування ХО. Загалом маємо тотожність (1.2) [5].

$$R(I) = R(C(S)) = r. \quad (1.2)$$

Тепер припустимо, що ми маємо зміну в записаній сцені S через переміщення об'єкта, зміни освітлення, поворот або нахил камери тощо. Це визначає нову сцену $S_{new} = N(S)$ та нове зображення $I_{new} = C(S_{new})$. Процедура \mathcal{X} є інваріантною до змін N (в ідеальному випадку), якщо ми отримуємо в результаті (1.3) такий же результат r для I_{new} як і у випадку I раніше.

$$R(I_{new}) = R(C(N(S))) = r. \quad (1.3)$$

Визначенням характерних ознак, які б були стійкими до графічних перетворень, таких як масштабування, поворот, зміщення, займаються алгоритми детектування ключових точок та їх дескрипторів.

1.2 Поняття ключових точок та їх дескрипторів

Ключова точка (або точка інтересу) – це певна точка (піксел) на зображенні, яка визначається певною інтенсивністю зображення навколо неї через досягнення значення екстремуму локальної особливості. Ключова точка може бути використана для отримання дескриптора. Дескриптор — це скінченний вектор, який підсумовує властивості ключової точки. Не кожен детектор ключових точок має свій особливий спосіб визначення дескриптора. Ключова точка і дескриптор разом визначають ознаку [5]. Відповідно до цього можна виділити дві основні простори даних – простір сигналів, в якому

формується розпізнавані образи, та простір ознак, в якому реалізується прийняття рішень про клас образів.

Визначимо математичну базу застосування дескрипторів КТ в задачі розпізнавання образів. Маємо I – простір зображення об'єктів; $i \in I$ – об'єкт, що необхідно розпізнати; $M[J] = \{1, 2, \dots, J\}$ – номери класів об'єктів I_1, I_2, \dots, I_j , що відповідають умові $I = \bigcup_{k=1}^J I_k$, $I_m \cap I_n = \emptyset$, де \emptyset – пуста множина та $m \neq n$. На цих множинах визначена функція $\hat{f}: I \rightarrow M$ – невідома функція, що пов'язує образ $i \in I$ з номером $j \in M$ відповідного класу [15, 16].

Простір характеристики об'єктів, що представлені дескрипторами (числовий вектор) ключових точок представимо у вигляді $Z = \{z | z \in R^n\}$. Об'єкт $i \in I$ за допомогою відображення $\theta: I \rightarrow Z$ пов'язується з його характеристикою $z(i) = \{z_{i=1}^n\}$, де $z(i)$ – скінченна множина, потужність якої – $s = \text{card } z(i)$. Простір Z має підмножини Z_1, Z_2, \dots, Z_n , що віднайдені шляхом впливу відображення $\theta: I_k \rightarrow Z_k$, та додатково задовольняють умовам $Z_i \cap Z_k = \emptyset$ та $Z = \bigcup_{i=1}^J Z_i$.

Опишемо вирішальне правило, що пов'язує скінченну множину $z(i)$ характеристик об'єкту з номером відповідного класу: $\hat{f}: Z \rightarrow M$. Для проведення розпізнавання потрібно провести ряд послідовних трансформувальних множин об'єктів у класи, що представлено формулою 1.4:

$$I \rightarrow \theta(I) \rightarrow Z \rightarrow \hat{f}(Z) \rightarrow M. \quad (1.4)$$

Основним завданням (1.4) є визначення правила \hat{f} , спираючись на подання у просторі Z . Продуктивність роботи правила \hat{f} залежить від обрання детекторів характерних ознак в просторі Z та $z(i)$. Ступінь якості розпізнавання визначає правильність віднесення вхідного об'єкту до певного класу.

КТ мають свої риси та властивості [5, 7, 9, 24]:

– інваріантність (визначення не залежить від афінних перетворень);

- інтерпретація (представлена так, щоб можна було виокремити інформацію про неї);
- відмінність (відрізняється від усіх інших точок, унікальні у своїй області зображення);
- стабільність (стійкість до геометричних перетворень зображення та шумів);
- локальність (розглядається у певній області, околі);
- чисельність (достатня кількість КТ для прийняття результативного рішення);
- ефективність (робота у режимі он-лайн або наближений до нього).

Для побудови КТ необхідно виокремити фрагмент з решти фону. Процес формування множини дескрипторів КТ складається з двох стадій. На першій стадії детектором виокремлюються КТ, кожна у певному околі. На другій стадії будуються дескриптори, які можна вважати варіантом представлення КТ, до якого може бути застосована міра відстані або подібності [6]. Щоб описати точки з координатами $p=(x,y)$, застосовується формула просторового околу (1.6). У даному випадку окіл $O(p)$ отримано у результаті використання оператора з побудови множини точок на основі близькості.

$$O(p) = [\{p_i = (x_i, y_i)\}, \forall p_i: \rho(p_i, p) \leq v], \quad (1.5)$$

де v – розмір околу;

ρ – деяка міра відстані між точками на площині.

Застосування для розпізнавання об'єктів множини КТ замість самого зображення дає можливість добитися помітного зменшення обсягів зайнятої пам'яті, але залишити майже незмінними значення достовірності та перешкодозахищеності за рахунок вилучення та обробки нової інформації, яка знаходиться у структурних елементах та зв'язках [25]. КТ становить числовий вектор з фіксованим розміром, що містить дійсні числа $\lambda^k =$

$(\lambda_1, \lambda_2, \dots, \lambda_n)^k$,

$R^1, i = \overline{1, n}$. Вектор атрибутів – точка багатовимірного векторного простору R^n , $\lambda_i \in R^n$ [15]. Значення n є розмірністю вектору атрибутів та залежить від алгоритму детектора.

1.3 Аналіз детекторів ключових точок

Розгляд методів для формування ключових точок та їх дескрипторів розпочнемо з їх принципових відмінностей. Перш за все детектори КТ можна поділити на дві основні групи за способом представлення дескрипторів [5]:

– дескриптори з плаваючою комою (характерні представники: SIFT, SURF, CSLBP);

– бінарні дескриптори, дані в яких представлені вектором двійкових (характерні представники: BRIEF, BRISK, ORB, KAZE).

Розмірність бінарних дескрипторів відповідає ступеням двійки. Таке подання дозволяє пришвидшити процес порівняння значень шляхом застосування двійкових операцій, що полегшує апаратну реалізацію.

Також існує класифікація детекторів у залежності від напрямку пошуку:

– дескриптори кутів (наприклад, ORB, BRISK);

– дескриптори областей (наприклад, SIFT, SURF, KAZE, AKAZE).

Розглянемо спочатку детектори дескрипторів із плаваючою комою. SIFT (Scale Invariant Feature Transform) – детектор ключових точок, що базується на побудові простору змінного масштабу, та розрахунку в ньому функції Лапласіана Гаусіана з різним значенням параметру згладжування. Якщо точка виявиться локальним екстремумом різниці Гаусіан, то вона буде визначена як КТ. Надалі множина віднайдених КТ звужується шляхом вилучення точок, що перебувають на границі об'єктів або мають малий контраст, а потім обчислюються орієнтації відібраних точок. Для кожної КТ розраховується орієнтація через гістограму градієнтів в околі. Вибір

напрямку орієнтації здійснюється шляхом порівняння вибору максимального з компонентів гистограми. Напрямки відображаються на 36 лічильниках, кожен з яких представляє інтервал у 10 градусів. Лічильники мають початкове значення 0. Якщо напрямок знаходиться в межах 10 градусів, представлених лічильником, то відповідна величина додається до лічильника. Загалом це визначає градієнтну гистограму.

Локальні максимуми значень лічильників, що становлять щонайменше 80% глобального максимуму, визначають домінантні напрямки. Якщо існує більше одного домінантного напрямку, то ключова точка використовується у зв'язку з кожним із цих домінуючих напрямків. Такий метод дозволяє КТ бути інваріантною до поворотів. Окрім поворотів метод SIFT стійкий до зсувів, шуму, зміни масштабу, низького контрасту та освітлення, позиції наглядача.

Детектор SURF (Speeded Up Robust Features) застосовує ідею схожу із SIFT, натомість він був створений для кращої продуктивності під час виконання. Цей метод використовує інтегральні зображення I_{int} (зображення, в яких значення яскравості кожного пікселя є сумою всіх значень яскравості певного прямокутного регіону) та маски в декількох масштабах, які є апроксимованими другими похідними функції Гауса (рис. 1.3) [5]. Такий підхід значно пришвидшує час розрахунку.

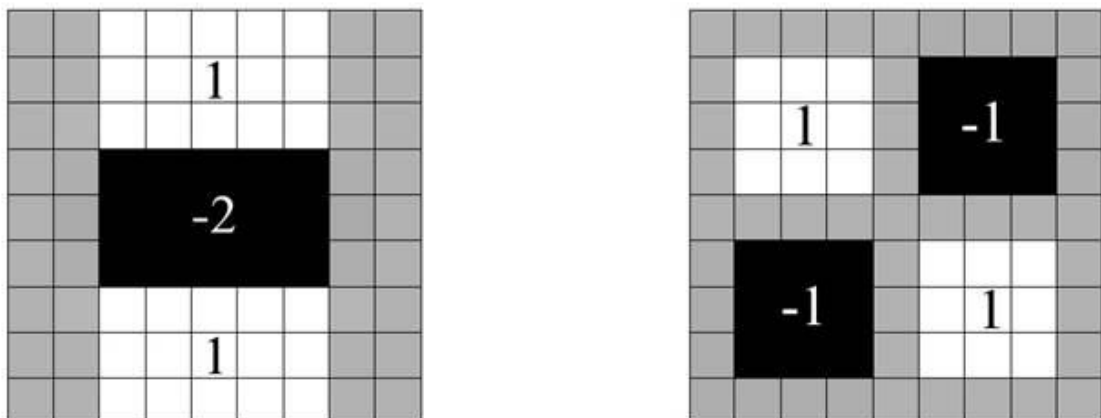


Рисунок 1.3 – Маски, що застосовуються для детектору SURF

Дескриптор SURF формується після виокремлення КТ та представляє набір з 64 чисел, що комбінують інформацію про локальні градієнти підрегіонів прямокутної сітки навколо КТ, що визначають домінуючий напрямок градієнту всієї області (рис. 1.4).

Алгоритм методу SURF є інваріантним до повороту та масштабування за рахунок застосування масок різного масштабу вздовж кожної осі та під кутом 45 градусів.

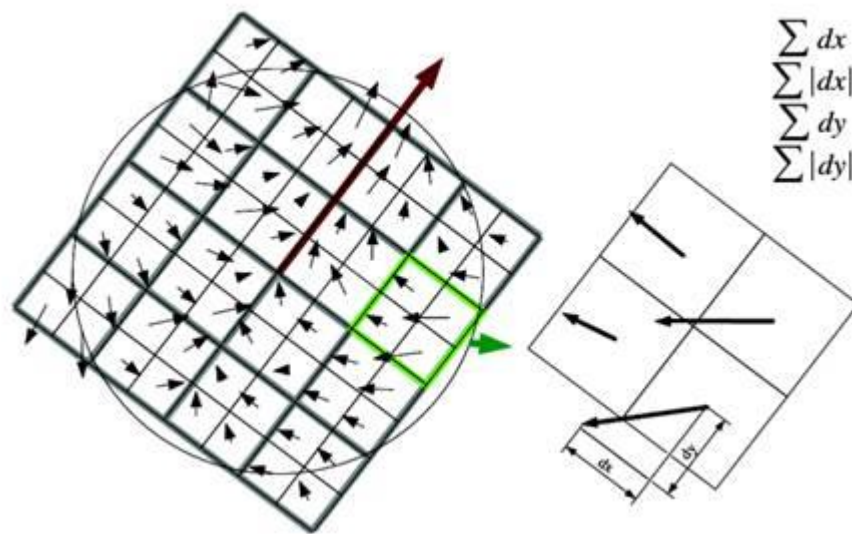


Рисунок 1.4 – Графічне представлення дескриптора SURF

Недоліком даного методу є неможливість ефективного застосування на великих приладах через складність процедури виявлення КТ та великої кількості необхідного часу.

Методи SURF та SIFT мають спільні недоліки: нечітке виокремлення об'єкта порівняно з фоном; об'єкти, що знаходяться на зображенні з низько вираженою текстурою, скоріш за все будуть не розпізнані. Ці методи заклали основу для розробки більш продуктивних за швидкістю методів ORB та BRISK [5].

Детектор ORB (oriented FAST and rotated BRIEF) є бінарним детектором дескрипторів КТ. Він поєднує ключові точки, визначені за детектором кутів FAST, із розширеним дескриптором BRIEF:

- ORB виконує багатомасштабне виявлення КТ за алгоритмом FAST (для стійкості до масштабування) та визначає домінанту орієнтацію;
- ORB застосовує розрахований головний напрямок для відображення дескриптора BRIEF у керований дескриптор BRIEF (для стійкості до повороту).

Алгоритм FAST (Features from Accelerated Segment Test) визначає навколо точки-кандидата p дискретний окіл певного радіусу (за замовчуванням 16 пікселів), в якому повинно існувати N сусідніх точок, в яких яскравість I_p усіх пікселів або більше, або менше порогового значення t . Далі яскравість обраної точки p зіставляється зі значеннями вертикальних та горизонтальних пікселів околу. Якщо для трьох пікселів з чотирьох виконується умова $I_i < I_p - t$ або $I_i > I_p + t$ при $i = 1, \dots, 4$, то тільки в цьому випадку проводиться повний тест для усіх 16 точок.

Алгоритм BRIEF (Binary Robust Independent Elementary Features) скорочує чисельність дескрипторів КТ з вектору розміром 128 елементів (як у SIFT) до тільки 128 бітів, що значно спрощує репрезентацію даних. Для початку дане зображення I перетворюється на бінарні рядки для певних областей. Для кожної області виконується Гаусове згладжування, визначене для $\sigma > 0$. Далі складається вибірка з n пар пікселів (p, q) , обраних навколо певної точки у визначеному порядку для $(2k+1) \times (2k+1)$ сусідів (наприклад, $k = 4$ до $k = 7$). Таким чином, порядок цих пар і параметри k і σ визначають конкретну версію дескриптора BRIEF. У процесі побудови дескриптора зіставляються значення яскравості у визначеній області (1.6).

$$s(p, q) = \begin{cases} 1: I(p) - I(q) > 0, \\ 0: I(p) - I(q) \leq 0, \end{cases} \quad (1.6)$$

де p, q – певні точки зображення;

$I(x)$ – значення яскравості зображення I в точці x .

Таким чином, буде одержано рядок розмірності (128, 256, 512). Для

порівняння знайдених дескрипторів використовується відстань Хеммінга. Даний метод є незалежним до переміни освітлення, спотворень, має високий рівень швидкодії та порівняння. Недоліком є залежність від повороту зображення.

Автори алгоритму ORB [5, 15, 24] розширили можливості перелічених алгоритмів. Для зображення I в алгоритмі FAST застосовують окіл S з радіусом $\rho = 9$ (але це залежить від розмірності самого зображення) з різними масштабами, вибираючи з усього переліку T найбільш схожих на кут точок, де $T > 0$ це попередньо визначений поріг кількості фільтром Харріса [5]. Далі визначаються моменти m_{10} та m_{01} (1.7) для даного околу, які розраховують напрям θ для ключової точки (1.8).

$$m_{a,b}(S) = \sum_{(x,y) \in S} x^a y^b \cdot I(x, y), \quad (1.7)$$

$$\theta = \tan^{-1} 2(m_{10}, m_{01}). \quad (1.8)$$

Для бінаризації вектору XO пари пікселів (p_i, q_i) при $0 \leq i \leq 255$ в методі BRIEF вибираються всередині диску з радіусом ρ із нормальним розподілом значень. Ці пари формують матрицю S , яку потім повертають множенням на матрицю повороту R_θ :

$$S_\theta = R_\theta S = R_\theta \begin{bmatrix} p_0 & \dots & p_{255} \\ q_0 & \dots & q_{255} \end{bmatrix} = \begin{bmatrix} p_{0,\theta} & \dots & p_{255,\theta} \\ q_{0,\theta} & \dots & q_{255,\theta} \end{bmatrix}. \quad (1.9)$$

Таким чином контрольований дескриптор BRIEF для алгоритму ORB розраховується як сума $s(p_{0,\theta} q_{0,\theta}) \cdot 2^0 \cdot + \dots + s(p_{255,\theta} q_{255,\theta}) \cdot 2^{255}$, s визначена у (1.6). Переходячи від оригінального дескриптора BRIEF до контрольованого, значення в дескрипторі більше корелюються для поліпшення дискримінаційної продуктивності розрахованих ознак [5].

Перелічені модифікації в детекторі ORB роблять його стійким до масштабування, повороту, зміни освітлення, дозволяють скоротити кількість необхідної пам'яті для збереження даних без втрати корисної інформації. Саме тому в даній роботі було обрано даний детектор ключових точок.

1.4 Огляд сучасних алгоритмів розпізнавання

1.4.1 AdaBoost

AdaBoost – це адаптивний мета-алгоритм машинного навчання. Мета-алгоритмом він називається через те, що звертається до підпроцедур, які називаються слабкими класифікаторами [14].

Слабкі класифікатори – це класифікатори, які не відповідають очікуванням за критерієм результативності та можуть бути трохи кращими за випадкове вгадування [5]. Але об'єднавши декілька слабких класифікаторів за певним правилом, можна отримати сильний класифікатор, спрямований на задовільне вирішення проблеми.

На початку маємо $w > 1$ слабких класифікаторів, визначених функціями h_j (1.9), які відображають простір дескрипторів R^n у множину $\{-1; +1\}$: при $h_i > 0$ клас визначається як «+1», при $h_i < 0$ клас «-1».

$$h(x) = w^T x + b, \quad (1.10)$$

де w – вектор ваг вхідних сигналів;

x – вектор вхідних даних;

b – вектор зміщення.

AdaBoost є адаптивним через ітеративне скорочення помилкових класифікацій: ваги помилково класифікованих дескрипторів коригуються з метою створення комбінованої (зваженої) програми з $w > 0$ слабких класифікаторів.

Даний мета-алгоритм застосовується в задачах навчання з учителем при наявності розмічених даних. Наприклад, його успішно застосовують в задачі розпізнавання обличчя на зображенні як апарат бінарної класифікації («присутнє обличчя» та «не присутнє») або в більш складних варіантах мультикласового розпізнавання. В якості дескрипторів можуть використовуватись вектори $V(\psi_p)$ з набору хвиль Хаара ψ_p (Haar wavelets) [5], які будуть точками в n -мірному просторі R^n .

Приклад алгоритму AdaBoost для бінарної класифікації [5, 25].

Крок 1. На вхід подається послідовність з N розмічених зразків $[(x_1, y_1), \dots, (x_N, y_N)]$, вибираються w слабких класифікаторів, визначається кількість ітерацій T .

Крок 2. Ініціалізація вектору ваг: $w_i^1 = 1/N$ для $i = 1, \dots, N$.

Крок 3. В циклі для $t = 1, \dots, T$ виконуємо Кроки 4-7.

Крок 4. Розраховуємо зважені помилки W_j (1.10) для кожного слабого класифікатора ($j = 1, \dots, w$).

$$W_j = \sum_{i=1}^N w_t(i) \cdot \llbracket y_i \neq h_j(x_i) \rrbracket, \quad (1.11)$$

де $w_t(i)$ – вага на ітерації номер t зразка номер i ;

y_i – мітка зразка;

$h_i(x_i)$ – результат класифікації i -ого зразка j -ого слабого класифікатора;

$\llbracket R \rrbracket$ – функція, що повертає 1 в разі вірності умови R та 0 в протилежному випадку.

Крок 5. Визначимо номер класифікатора-переможця $a(t)$ з мінімальною зваженою помилкою W_j . Якщо $W_{a(t)} \geq 0,5$, виходимо з циклу.

Крок 6. Визначаємо параметри α_t та c_t для оновлення ваг зразків:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - W_{a(t)}}{W_{a(t)}} \right), \quad (1.12)$$

$$c_t(i) = \frac{1}{s_t} \cdot \exp(-\alpha_t y_i h_{a(t)}(x_i)), \quad (1.13)$$

$$s_t = \sum_{i=1}^N \exp(-\alpha_t y_i h_{a(t)}(x_i)) \cdot w_t(i), \quad (1.14)$$

де α_t – коефіцієнт довіри класифікатору-переможцю;

$W_{a(t)}$ – зважена помилка класифікатора-переможця;

$c_t(i)$ – коефіцієнт для зміни ваг $w_t(i)$;

$h_{a(t)}$ – результат класифікації переможця;

s_t – нормалізуючий параметр для виконання умови $\sum w_{t+1} = 1$.

Крок 7. Оновлення ваг за правилом $w_{t+1}(i) = c_t(i) \cdot w_t$.

Результуюче правило для сильного класифікатора, згенерованого алгоритмом AdaBoost буде наступним:

$$H(x) = \text{sign}(\sum_{t=1}^T \alpha_t \cdot h_{a(t)}(x)). \quad (1.15)$$

Таким чином класифікатор фокусує увагу на невірні класифікованих об'єктах, залишаючи немінними усі інші, що зменшує розрахункові втрати.

Перевагами даного алгоритму є:

- швидкість;
- простота програмування;
- мала кількість параметрів для налаштування (вибирається тільки параметр кількості ітерацій T);
- універсальність до характеру даних (чисельні, текстові, дискретні);
- гнучкість – може комбінуватись із будь-яким алгоритмом навчання;
- мало схильний до перенавчання.

Натомість цей алгоритм чутливий до шуму в даних та викидам. Також якщо обрані слабкі класифікатори є надто складними (схильними до перенавчання) або слабкими, то це позначиться на роботі всього AdaBoost.

1.4.2 Класифікація методом нечітких правил

Так як при класифікації зображень дескриптори КТ подаються довгими векторами, що генеруються сотнями для кожного зображення, виникає проблема зі швидкістю ідентифікацією об'єктів [26-34]. Щоб знайти схожі до заданого зображення, потрібно порівняти всі дескриптори ознак з усіх зображень, зазвичай за деякою мірою відстані. Таке порівняння є занадто трудомістким, тому у всьому світі тривають дослідження для прискорення цього процесу.

Головна проблема застосування таких методів як гістограми кластерів КТ або «мішок слів» («bag-of-features», BoF) є те, що всі одиничні дескриптори або апроксимації наборів дескрипторів мають бути порівняні між собою, що потребує значних обчислювальних витрат. Більше того, підхід «мішка слів» потребує переробки класифікатора, щоб додати нові візуальні класи в систему. Для вирішення цих проблем було винайдено метод класифікації за нечіткими правилами [26, 32].

В цьому підході застосовується модифікована версія алгоритму AdaBoost, щоб вибрати найважливіші ознаки з великої кількості елементарних прямокутних атрибутів, подібних базовим функціям Хаара. Вибрані ознаки використовуються як слабкі класифікатори для завдання пошуку зображень на основі вмісту, зокрема для зображень, що містять обличчя. Натомість, можна використовувати інші детектори дескрипторів КТ (SIFT, SURF, ORB тощо) до більш широкого класу зображень. Тобто включення нових візуальних класів у систему вимагає лише додавання нових нечітких правил до бази правил без її реструктуризації.

Сутність цього алгоритму полягає в застосуванні нечіткої логіки та нечітких правил як керованого представлення кластерів візуальних ознак. Комбінуючи нечітку логіку та мета-алгоритм бустингу, вдається вибрати найбільш репрезентативний набір ознак зображення для кожного розглянутого класу об'єкту.

На кожному кроці випадковим шляхом обирається одна ознака з набору позитивних зображень (зображень з обличчям, наприклад), враховуючи ваги ознак, розрахованих алгоритмом AdaBoost. Обрана характерна ознака становить базу для побудови слабкого класифікатора. Далі шукають схожі вектори ознак серед усіх позитивних зображень. Схожість може бути визначена як Евклідова відстань між значеннями ознак. Слабкий класифікатор, отриманий у формі нечіткого правила, та вибрана ознака складають базу для визначення початкових параметрів нечіткого правила [32]. В такому разі правила R для класу c при застосуванні детектору ORB для вилучення характерних ознак мають наступну форму:

$$R_t^c: IF x_1 \text{ is } G_{1,t}^c \text{ AND } x_2 \text{ is } G_{2,t}^c \text{ AND } \dots \quad (1.16)$$

$$\dots \text{ AND } x_{256} \text{ is } G_{256,t}^c \text{ THEN image } i \in \omega_c(\beta_t^c),$$

де $t = 1, \dots, T^c$ це номер правила в конкретному запуску AdaBoost;

T^c – це число правил для певного класу зображення ω_c ;

β_t^c – це параметр важливості класифікатора, згенерований на t -ому кроці алгоритму AdaBoost для класифікації об'єктів з класу ω_c ;

$G_{n,t}^c$ – слабкий класифікатор, створений за наборами нечітких правил з Гаусівською функцією належності.

У процесі навчання слабкі класифікатори налаштовуються відповідно до зображень, правильно віднесених до свого класу [2, 3, 6, 15]. Такий підхід може бути корисним для пошуку, заснованому на вмісті зображення у множині складних графічних об'єктів, збережених у базах даних.

Головні переваги даної системи полягають у наступному:

- новий метод для автоматичної побудови бази нечітких правил для класифікації зображень на основі локальних ознак;
- ефективна техніка для швидкого навчання та класифікації зображень;

- метод для автоматичного пошуку найбільш значущих локальних ознак на даному класі зображень;
- гнучка система індексування та зберігання ознак зображень;
- розширення системних знань є ефективним, оскільки додавання нових візуальних класів до системи вимагає створення тільки нових нечітких правил, тоді як у випадку «мішку слів» це вимагає нового словника та повторного вивчення класифікаторів.

Тим не менш найпопулярнішим апаратом обробки даних складної структури на сьогодні є штучні нейронні мережі.

1.4.3 Штучні нейронні мережі

Штучна нейронна мережа – це математична модель, побудована за принципом функціонування біологічних нейронних мереж, яка є мережею вузлів, званих нейронами, які отримують вхідний сигнал, змінюють свій внутрішній стан відповідно до цього входу (збуджуються) і виробляють вихідний сигнал, залежний від входу та збудження. Мережа утворюється з'єднанням виходів певних нейронів зі входами інших з утворенням орієнтованого зваженого графу.

Таким чином кожен нейрон можна вважати агрегатором, який сумує вхідний сигнал x_j ($j = 1, 2, \dots, N$) попередніх нейронів (1.17) та пропускає його крізь функцію активації (1.18), яка породжує вихідний сигнал y_i :

$$u_i = \sum_{j=0}^N w_{ij}x_j, \quad (1.17)$$

$$y_i = f\left(\sum_{j=1}^N w_{ij}x_j(t) + b_{i0}\right), \quad (1.18)$$

де y_i – вихідний сигнал i -ого нейрона;

$x_j(t)$ – вхідний сигнал j -ого нейрона у момент часу t ;

w_{ij} – ваги синаптичних зв'язків між нейронами, що регулюють момент збудження нейрона;

b_{i0} – поріг, після перевищення якого настає момент збудження нейрона;

$f(u_i)$ – нелінійна активаційна функція, яка формує вихідний сигнал.

Функції, які можуть виконувати ШНМ, можна розподілити на декілька основних груп: апроксимація й інтерполяція; розпізнавання та класифікація образів; стискання даних; прогнозування; ідентифікація; управління; асоціації [2].

Для класифікації та розпізнавання образів ШНМ навчається важливішим ознакам, таким як геометричне відображенні точкової структури зображення, відносно розміщення важливіших елементів образу, компоненти перетворення Фур'є тощо. В процесі навчання виділяються ознаки, які відрізняють образи один від одного, на яких складається база прийняття рішень про віднесення образу до відповідного класу.

Перевагами штучних нейронних мереж є розмаїття прикладних задач, висока точність порівняно з іншими моделями, гнучкість та простір для вдосконалення.

До недоліків відносяться:

- природа «чорного ящика» – невідомо як і чому модель видає певні вихідні дані;
- необхідність величезного об'єму даних (мільйони зразків);
- необхідність значних обчислювальних ресурсів (складні моделі навчаються тижнями).

Архітектура мережі, структура нейронів, вибір активаційної функції та метод навчання моделі напряму впливають на результативність проведених операцій. В цьому сенсі нейронні мережі із самоорганізацією на основі конкуренції мають унікальні властивості з точки зору пристосування до конкретних даних прикладної задачі аналізу зображень. Саме тому дана архітектура буде детально розглянута в даній кваліфікаційній роботі.

1.5 Постановка задачі

Класифікація візуальних об'єктів за множиною дескрипторів КТ є актуальною задачею у сфері розпізнавання зображень для комп'ютерного зору. Поставлено задачу розроблення методів класифікації розпізнаваних зображень та отримання прийнятної результативності за рахунок застосування конкурентних методів самонавчання.

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

Метою роботи є розроблення методів класифікації, що базуються на використанні технологій самонавчання. Опис зображення подається у вигляді множини дескрипторів ключових точок. Для формування множини дескрипторів ключових точок використовувався детектор ORB.

Для досягнення мети необхідно вирішити такі завдання:

- ознайомитись з алгоритмами класифікації на основі самонавчання;
- розробити й програмно реалізувати метод класифікації із впровадженням мережі із самоорганізацією на основі конкуренції на множині еталонних описів класів;
- експериментально перевірити результативність розробленого методу класифікації для прикладної бази зображень.

2 МОДЕЛІ НЕЙРОННИХ МЕРЕЖ ДЛЯ АНАЛІЗУ ЗОБРАЖЕНЬ

2.1 Моделі нейронів та методи їх навчання

Моделі нейронів впродовж розвитку історії на різних етапах зазнали значних перетворень, за якими їх можна класифікувати на моделі без пам'яті, адаптивні моделі та моделі з керованою пластичністю [4].

На першому етапі моделювання нейронних мереж властивості мережі, пов'язані характером передачі сигналу, залишались фіксованими. Загальною схемою, яка складає основу більшості моделей штучного нейрона, вважається модель МакКалока-Пітса, зображена на рисунку 2.1, яка містить суматор зважених вхідних сигналів та нелінійний блок вихідного сигналу, функціонально залежний від вихідного сигналу суматора. Поступово модель визомніювалась, враховуючи вади попередніх зразків. Властивості нелінійної функції, особливо її неперервність, особливим чином впливають на вибір способу навчання нейронів, а саме підбору коефіцієнтів для синаптичних зв'язків між ними. Ще одним важливим фактором є вибір стратегії навчання – навчання з учителем (окрім даних відомі ще очікувані вихідні сигнали) або без учителя (із даними як такими) [2, 3, 6].

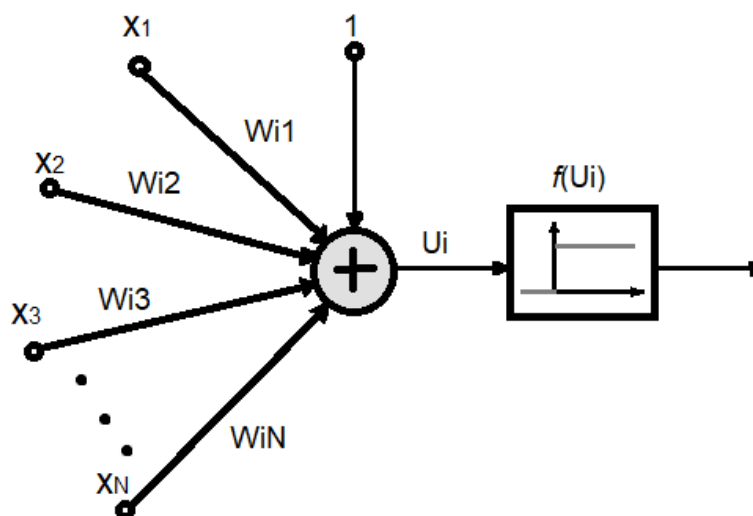


Рисунок 2.1 – Модель нервової клітини МакКалока-Пітса

Вагові коефіцієнти в разі навчання з учителем підбираються на основі міри відстані d_i між поточним виходом системи та очікуваним. Таким чином з кожною ітерацією навчання відстань d_i має скорочуватись та наближувати модель до вірної конфігурації. В разі навчання без учителя проводиться конкуренція нейронів між собою за стратегіями «переможець отримує все» (Winner Takes All – WTA) або «Переможець отримує більше» (Winner Takes Most – WTM) або із врахуванням кореляції навчальних та вихідних сигналів (навчання Хебба). Розглянемо деякі з найбільш репрезентативних моделей, реалізуючих кожний підхід [2, 3, 15].

Першою моделлю нейрона став простий персептрон МакКалока-Пітса (1.18). Його нелінійна функція (функція активації) була дискретною функцією ступінчастого вигляду, тому вихідний сигнал мав тільки два значення – 0 або 1:

$$y_i(u_i) = \begin{cases} 1 & \text{для } u \geq 0, \\ 0 & \text{для } u < 0, \end{cases} \quad (2.1)$$

де u_i – вихідний сигнал суматору за формулою (1.17).

В формулі (2.1) мається на увазі, що вхідний вектор x має розмірність N елементів та доповнений нулевим членом $x_0 = 1$, який формує сигнал поляризації. Персептрон потребує навчання з учителем, тому з кожним вхідним вектором x дається очікуваний результат r . Налаштування ваг між нейронами проводиться відповідно до правила персептрона за наступним алгоритмом.

Крок 1. Початково ініціюються значення ваг w_{ij} , зазвичай випадковим шляхом.

Крок 2. Розраховуються вихідні значення y_i за вхідним вектором x з навчальної вибірки.

Крок 3. За отриманим виходом y_i уточнюються значення ваг w_{ij} , відповідно до очікуваного виходу r_i :

– в разі, якщо $y_i = 0$, а $r_i = 1$, тоді значення ваг уточнюються за формулою $w_{ij}(t+1) = w_{ij}(t) + x_j$, де t позначає номер попереднього циклу, а $(t+1)$ – номер поточного циклу;

– в разі, якщо $y_i = 1$, а $r_i = 0$, тоді значення ваг уточнюються за формулою $w_{ij}(t+1) = w_{ij}(t) - x_j$, де t позначає номер попереднього циклу, а $(t+1)$ – номер поточного циклу;

– якщо значення співпали, тоді вагові коефіцієнти не змінюються.

Крок 4. Якщо вектор x не є останнім, тоді обирається наступний вхідний вектор з навчальної вибірки та перехід на Крок 2, в іншому разі перехід на Крок 5.

Крок 5. Якщо поточний цикл t не є останнім і поточна різниця між y_i та r_i не є мінімально задовільною, тоді перехід на Крок 2.

Параметром зупинки навчання, окрім кількості ітерацій циклу, виступає функція похибки, значення якої алгоритм намагається мінімізувати. Найчастіше в якості функції похибки обирають суму різниці квадратів.

Хоча алгоритм є доволі простим, його ефективність при великих обсягах вибірки є відносно невеликою, а кількість циклів навчання зростає дуже стрімко, при чому без гарантування досягання мінімуму. Для усунення даної вади необхідно використовувати неперервну функцію активації, завдяки чому функція похибки теж буде неперервною, що дозволить використовувати в процесі навчання інформацію про величину градієнта.

Адаптація та пам'ять стали властивостями, які впроваджувались із параметричними змінами в мережі (додавання коефіцієнтів, налагодження яких впливає на швидкість навчання моделі). На зміну персептрону прийшов сигмоїдальний нейрон, в якому функція активації мала наступний вигляд (2.2):

$$f(x) = \frac{1}{1+e^{-\beta x}}, \quad (2.2)$$

де β є параметр, який підбирається самостійно користувачем.

Значення параметра β лежить у діапазоні $(0; 1)$ і впливає на вид графіку сигмоїдальної функції. При малих значеннях β графік є достатньо пологим, тоді як по мірі збільшення крутизна графіку збільшується та наближається до ступінчастого типу (рис. 2.2). Зазвичай параметр β беруть рівним 1.

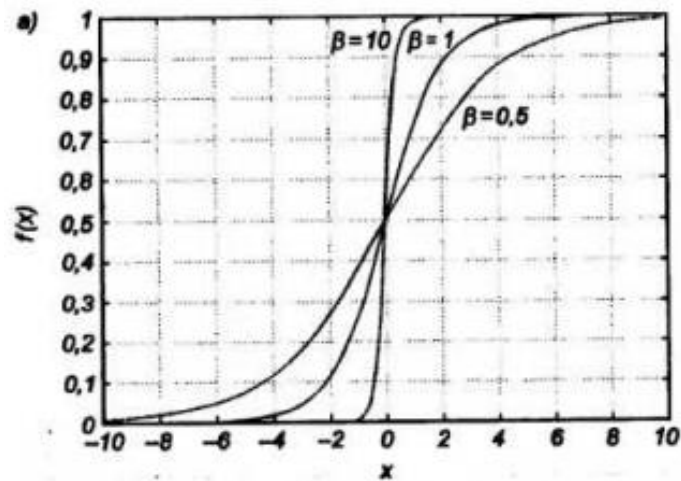


Рисунок 2.2 – Графік сигмоїдальної функції

Важливою властивістю сигмоїдальної функції є її диференційованість, що дозволяє застосовувати градієнтні методи, такі як метод найшвидшого спуску, відповідно до якого уточнення вектора ваг проводиться у напрямку від'ємного градієнта цільової функції. В разі застосування середньої квадратичної помилки як цільової функції E значення j -ї складової градієнту буде мати вигляд:

$$\nabla_j E = \frac{dE}{dw_{ij}} = x_j (y_i - r_i) \frac{df(u_i)}{du_i} = x_j \delta_i, \quad (2.3)$$

де u_i – зважена сума вхідних сигналів;

$f(u_i)$ – сигмоїдальна функція;

$\delta_i = (y_i - r_i) * df(u_i)/du_i$ – коефіцієнт корегування;

$\nabla_j E$ – градієнт цільової функції.

Навчання даного виду нейрон проводиться за формулою (2.4). На ефективність навчання сильно впливає вибір коефіцієнту навчання α та коефіцієнту згаджування сигмоїди β . Існують варіанти, де ці параметри обираються константами або задаються змінною величиною, значення якої змінюється в процесі навчання адаптивним шляхом. Хоча адаптивний шлях і є більш ефективним, він є одночасно найбільш трудомістким, бо значення коефіцієнту навчання підбирається на кожному кроці шляхом мінімізації цільової функції однієї змінної

$$w_{ij}(t + 1) = w_{ij}(t) - \alpha \delta_i x_j. \quad (2.4)$$

Необхідно зазначити, що використання градієнтного методу навчання нейрона гарантує досягнення тільки локального мінімуму. У випадку полімодальної цільової функції є імовірність не вийти із локального мінімуму та бути досить далеко від глобального [26, 35].

Якщо необхідно скласти нейронну мережу з нейронами, схема яких зображеному на рисунку 2.1, із реалізацією простого ефекту запам'ятовування, особливо в тих випадках, коли мова йде про асоціативну або адресовану за змістом пам'ять, то зміну взаємозв'язків між елементами такої мережі традиційно описують за гіпотезою Хебба.

У процесі вивчення нервових клітин Д. Хебб виявив, що якщо деяка клітина A перебуває у стані близькому для збудження клітини B , а також періодично або постійно бере участь її активації, тоді має місце процес росту або метаболічної зміни в одній або обох клітинах, щоб посилити ефективність клітини A , як однієї з впливових на клітину B . Тобто якщо j -а клітина з вихідним сигналом y_j пов'язана з i -ою клітиною, яка має вихідний сигнал y_i , зв'язком з вагою w_{ij} , тоді на силу цього зв'язка впливатимуть значення y_j та y_i (2.5).

$$\frac{dw_{ij}}{dt} = \alpha y_i y_j, \quad (2.5)$$

$$w_{ij}(t + 1) = w_{ij}(t) + \alpha y_i y_j, \quad (2.6)$$

де α – коефіцієнт швидкості навчання в інтервалі $(0; 1)$;

dw_{ij} – приріст значення ваг зв'язку між клітинами i та j ;

y_i, y_j – вихідний сигнал нейронів i та j відповідно.

Правило Хебба характеризується тим, що в результаті його використання ваги можуть необмежено зростати, оскільки в кожному циклі навчання сумується його поточне значення із приростом Δw_{ij} (2.6). Одним зі способів стабілізації процесу навчання за правилом Хебба введення обмеження, які б описували ефект насичення, ефект забування завдяки зменшенню ваг на коефіцієнт забування γ , який обирається емпірично.

Ще однією важливою властивістю пам'яті в моделях нейронних мереж є асоціативні області. З досліджень властивостей кори головного мозку було виявлено, що різні області мозку організовані у відповідності з різними чуттєвими модальностями: існують області, що вирішують спеціальні задачі, такі як управління мовленням або аналіз сенсорних сигналів. Результати проведених експериментів виявили тонку структуру всередині багатьох областей, відповідно до якої сигнали реакцій отримуються в корі головного мозку в тому ж топографічному порядку, в якому вони були прийняті органами чуття. Ці структури називаються мапами [3].

Моделі клітин, спеціалізовані за спільними властивостями, будуються за теорією змагального навчання, яка запроваджує самоорганізацію загальної системи. Згідно цього методу «змаганням» прийнято вважати процес, при якому нейронні моделі M_i однієї системи одночасно збуджуються одним і тим же сигналом X . Нейронна модель M_c , що активізувалася в найбільшому ступені, вважається «переможцем» та може придушити активність інших нейронів, наприклад, латеральним гальмуванням [3, 24]. Також однією з умов самоорганізації є модифікація моделей в околі переможця ρ , тим самим вони будуть більше відповідати заданому сигналу, а зв'язок нейронів близьких до переможців стає міцнішим. Отже, дані три процеси (поширення вхідного

сигналу X , вибір переможця та адаптація моделей в його околі) описують самоорганізацію, на засадах якої виникають торографічно організовані «мапи» (самоорганізовані мапи).

Самоорганізовані мапи застосовуються як математичний апарат, придатний для практичного застосування при розв'язанні широкого спектру задач, тому в рамках даної роботи розглянемо одну з архітектур, реалізуючих конкурентне навчання – мережу Кохонена.

2.2 Базові моделі мереж із самоорганізацією

Метою навчання мережі із самоорганізацією на основі конкуренції є таке налагодження зв'язків між нейронами, яке мінімізує очікувану похибку апроксимації вхідного вектора. При p вхідних векторів x та застосуванні евклідової метрики, похибка квантування може бути виражена формулою (2.7).

$$E_q = \frac{1}{p} \sum_{i=1}^p \|x_i - w_{w(t)}\|^2, \quad (2.7)$$

де E_q – похибка квантування;

$w_{w(t)}$ – вага нейрона-переможця для заданого вектору x_i .

Базовою моделлю нейрона для реалізації конкурентного навчання можна вважати модель «Переможець отримує все» (Winner Takes All - WTA). Нейрони даного типу мають вхідний модуль у вигляді стандартного суматора за формулою (1.16), але ваги w_{ij} позначають не зв'язки нейронів між собою, а між вхідним сигналом та кожним окремим нейроном. Кожен з нейронів отримує однаковий вхідний сигнал x_j , а його вихід залежить від вагів w_{ij} , які на початку обираються випадково. В результаті порівняння фактичного значення u_i кожного нейрона обирається переможець із максимальним значенням, який може оновити значення w_{ij} (2.8) [2, 3].

$$w_{ij}(t + 1) = w_{ij}(t) + \alpha(x_j - w_{ij}(t)). \quad (2.8)$$

У ході навчання вагові коефіцієнти переможців наближуються до значень навчальних векторів x . Якщо на вхід мережі будуть подаватись багато подібних за значенням векторів x , тоді буде перемагати один і той же нейрон. Через це його ваги будуть усереднені значенням всіх навчальних векторів, які зробили його переможцем. Наслідком такої конкуренції стає самоорганізація процесу навчання. Нейрони, які перемагали завдяки деякій групі близьких за значенням даних, будуть краще розрізняти свою категорію даних серед інших. Системи такого типу частіше за все застосовуються для класифікації векторів.

Окрім зазначеної вище моделі нейронів, в якій на кожній ітерації навчається тільки один нейрон, для навчання мереж із самоорганізацією широко застосовується алгоритм типу «Переможець отримує більше» (Winner Takes Most – WTM), в якому, окрім переможця, ваги уточнюються сусідні до переможця елементи: чим ближче – ти більше змінюються. В такому разі формула для оновлення ваг буде виглядати наступним чином:

$$w_i(t + 1) = w_i(t) + \alpha_i \cdot G(i, c) \cdot (x - w_i(t)), \quad (2.9)$$

де w_i – вага i -го нейрона;

α_i – коефіцієнт навчання, окремий для кожного нейрона;

x – вектор вхідного сингалу;

t – ітерація в циклі навчання;

$G(i, c)$ – функція сусідства, яка визначає коефіцієнт зміни ваг, відповідно до відстані від i -го нейрона до переможця c .

Алгоритм WTA є окремим випадком алгоритму WTM, в якому функція сусідства бінарна та приймає значення 1 та 0. Іншим варіантом алгоритму «Переможець отримує більше» є мережа Кохонена.

2.3 Мережа Кохонена

Мережа Кохонена (Self-Organizing Map – SOM) є розрахунковим методом для візуалізації та аналізу даних високої розмірності, особливо експериментально отриманої інформації. Він був розроблений у 1984 році фінським науковцем Теуво Кохоненом. Формально SOM можна визначити як нелінійне, упорядковане, гладке відображення різноманіття вхідних даних високої розмірності на елементи регулярного масиву низької розмірності (найчастіше на двовимірну сітку) [3, 26].

Мережа Кохонена не є першим алгоритмом у зазначеному напрямку вивчення моделей асоціативної пам'яті та адаптивного навчання, до нього були просторово впорядковані детектори ліній фон дер Мальсбурга (1973 р.) і модель нейронного поля Амарі (1980 р.), проте потужність самоорганізації попередників була слабою. Винахід Кохонена складався у взаємодії двох підсистем різної природи: конкуренції нейронів типу «Переможець отримує все» та інша підсистема, яка контролюється нейронною мережею і яка змінює локальну синаптичну пластичність нейронів під час навчання. Тільки завдяки розділенню передачі нейронного сигналу та контролю пластичності стало можливим реалізувати ефективну та надійну самоорганізуючу систему.

Нехай набір вхідних змінних $\{x_i\}$ можна визначити як вектор раціональних чисел $x = [x_1, x_2, \dots, x_n]^T \in R^n$. З кожним елементом вхідної послідовності в мережі пов'язано параметричний вектор раціональних чисел (ваги зв'язків) $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T \in R^n$, названий моделлю. Кожна модель має свої координати в сітці, таким чином нейрони топографічно пов'язані один з одним. Перед початком навчання всі моделі повинні бути ініціалізовані. Від того яким чином відбудеться ініціалізація буде залежати те, як швидко мережа навчиться.

Відстань між x та m_i позначається як $d(x, m_i)$, а відображення входу на SOM визначається як пошук такої моделі m_c , яка найкращим чином відповідає вхідному вектору (пошук найбільш підходящого вузла) [2, 30]:

$$c = \arg \min_i \{d(x, m_i)\}. \quad (2.10)$$

У цьому виразі відстань $d(x, m_i)$ може бути евклідовою метрикою або метрикою Хемінга в разі бінарних векторів. Побудоване відображення повинно бути упорядкованим та наочним представленням розподілу для навчальної множини x . Слід зазначити, що ваги моделі не обов'язково повинна бути векторними величинами: достатньо, щоби міра відстані $d(x, m_i)$ була визначена для всіх елементів x та великої кількості моделей m_i .

В процесі навчання вузли, які топографічно близькі за деякою геометричною відстанню, будуть активувати один одного, за рахунок чого будуть навчатися на одних вхідних даних. У такий спосіб впроваджується ефект локального згладжування, який при достатній кількості приводить до глобального впорядкування нейронів. Топологія сітки може бути прямокутною (рис. 2.3), шестикутною та нелінійної структури.

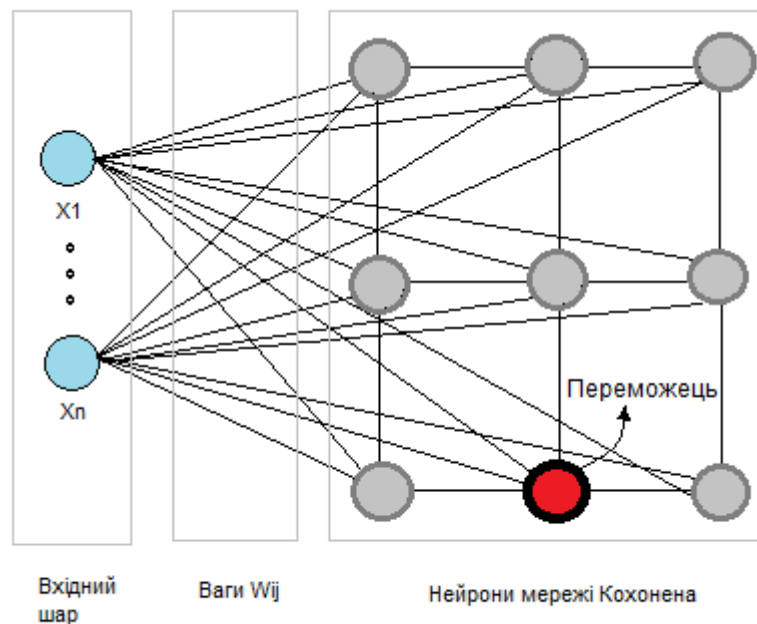


Рисунок 2.3 – Структура мережі Кохонена

Ваги моделей змінюються за формулою (2.11) і в ній центральну роль грає функція сусідства. Вона значним чином впливає на результат

згладжування, бо визначає на яку відстань від переможця та в якій мірі розповсюджується оновлення моделей. Для збереження збіжності алгоритму необхідно, щоб $G(i, c) \rightarrow 0$ при $t \rightarrow \infty$. Загально відомо дві реалізації для функції сусідства: просто окіл вузла c , тобто множина точок, сусідніх з ним (прямокутне сусідство); сусідство гаусівського типу (2.12).

$$w_i(t + 1) = w_i(t) + \alpha(t) \cdot G_t(i, c) \cdot (x - w_i(t)), \quad (2.11)$$

$$G(i, c) = \exp\left(\frac{-d^2(i, c)}{2\sigma^2(t)}\right), \quad (2.12)$$

де w_i – вага i -го нейрона;

$\alpha(t)$ – коефіцієнт швидкості навчання;

t – ітерація в циклі навчання;

x – вектор вхідного сингалу;

$G(i, c)$ – функція сусідства між i -м нейроном та переможцем c ;

$d(i, c)$ – міра відстані між i -м нейроном та переможцем c ;

$\sigma(t)$ – параметр, що визначає ширину ядра.

У формулі для збіжності алгоритму необхідно, щоб коефіцієнт швидкості навчання та параметр ширини ядра мають залежати від ітерації навчання, а саме – монотонно спадати (наприклад, за лінійним, експоненціальним або обернено пропорційним законом).

В даній роботі буде застосовано обернено пропорційний закон для визначення коефіцієнта швидкості навчання за ітерацією за формулою:

$$\alpha_t = \left(1 - \frac{t}{T}\right) * \alpha_0, \quad (2.13)$$

де t – поточна ітерація;

T – загальна кількість ітерацій;

α_0 – початковий коефіцієнт навчання;

α_t – коефіцієнт навчання на ітерації t .

На відміну від прямокутного типу сусідства, де кожен нейрон у рівній мірі адаптується в околиці переможця, при сусідстві гаусівського типу рівень адаптації залежить від значення функції Гауса. Завдяки цій відмінності забезпечуються кращі результати навчання та організація мережі.

Так як мережа Кохонена є одним з перших алгоритмів у сфері конкурентного навчання, вже було запропоновано багато його модифікацій. Проте в межах даної роботи ми оберемо саме його як класичний приклад.

2.4 Оцінювання ефективності класифікації зображень

В даній роботі буде реалізовано класифікацію зображень завдяки мережі Кохонена на основі дескрипторів ключових точок бази еталонів.

Класифікація буде проходити за наступними кроками.

Крок 1. Для кожного зображення I_j розрахуємо дескриптори КТ ORB в кількості 500 векторів, які сформулюють опис $Z_j = \{z_{jk}\}_{k=1}^{500}$.

Крок 2. Генерується мережа Кохонена з кількістю вузлів рівною кількості зображень.

Крок 3. Значення ваг нейронів мережі ініціалізуються.

Крок 4. Навчаємо нейронну мережу на вибірці з усіх дескрипторів КТ за формулою (2.8) моделі нейронів «Переможець отримує все».

Крок 5. Тестуємо результативність моделі на тестувальній вибірці методом голосування: рішення про клас об'єкта визначається за більшістю голосів дескрипторів, віднесених до певного класу [35].

Крок 6. Оцінюємо модель.

Нехай TP та FP позначатимуть числу вірно-позитивних та хибно-позитивних (віднесення до певного класу хибного об'єкта) виборів моделі відповідно. Аналогічно визначимо TN та FN (не віднесення об'єкта до його класу, промах) як вірно-негативні та хибно-негативні зразки відповідно.

Дані категорії застосовуються у певних мірах оцінки ефективності. В даній роботі застосуємо такі міри як точність (accuracy – AC) та повноту (recall – RC).

Точність є відношенням вірних (TP + TN) виборів моделі до всіх виборів, повнота є відношенням вірно-позитивних зразків до всіх потенційно можливих розпізнавань (наприклад, число всіх видимих об'єктів на зображенні).

$$AC = \frac{TP+TN}{TP+TN+FP+FN}, RC = \frac{TP}{TP+FN}. \quad (2.14)$$

Максимальне значення точності відповідає $AC = 1$, що означає, що усі класи визначені вірно там де вони є та без помилкових спрацьовувань. Повнота застосовується у випадках, коли кількість вірно-негативних випадків заздалегідь невідома, через що цей показник не використовується. При максимальному значенні $RC = 1$ усі видимі об'єкти на зображенні визначені та немає хибно-негативних FN випадків (промахів).

3 КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ МЕТОДУ НАВЧАННЯ ТА КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

3.1 Обґрунтування вибору середовища програмної реалізації

У рамках кваліфікаційної роботи було розроблено та досліджено метод класифікації зображень за допомогою мережі Кохонена на множині бінарних дескрипторів бази зображень-еталонів [2, 36].

Для програмної реалізації нами обрано середовище розробки PyCharm Community Edition 2020.2.3, мову програмування Python та бібліотеку OpenCV. Апаратне забезпечення представлено ноутбуком з процесором Intel Celeron CPU N3060 1.60 ГГц та ОЗП 4 ГБ.

Було обрано мову програмування Python через простий синтаксис, відкритий код, велику кількість зручних бібліотек для обробки зображень та аналізу великих. Важливим моментом є графічний інтерфейс – Python має багато варіацій інтерфейсів, що просто імпортуються до інтерпретатора. Програмування на Python можна здійснювати як за функціональним підходом, так і за об'єктно-орієнтованим. При розробці Python був створений як мова програмування високого рівня, що звільняє розробника від необхідності управління пам'яттю, має вбудовані базові структури даних (хеш-таблиці, списки, кортежі тощо) архітектурою та пам'яттю. Ще однією важливою перевагою є виконання коду інтерпретатором за рядками, що дозволяє не компілювати код кожного разу та полегшує процес налаштування. Однак суттєвим недоліком є те, що програми на Python працюють повільніше ніж аналоги на компільованих мовах програмування, таких як C++ або Java.

Інтегроване середовище розробки PyCharm надає великий спектр інструментів для ефективної роботи з мовою програмування Python. Редакція Community є безкоштовною та має відкритий код, підходить для розробки інтелектуальних систем, а також містить підтримку коду та рефакторинг,

візуальну відладку, підтримує системи контролю версій.

PyCharm має наступні переваги:

- підтримка при редагуванні коду (автоматичне доповнення, знаходження та виправлення помилок);
- зручна навігація;
- швидкий рефакторинг;
- інтерактивна консоль та вбудований термінал Python;
- підтримка бібліотек для інтелектуального аналізу даних (Pandas, Numpy, Matplotlib та інші);
- кросплатформеність (Windows, Mac OS чи Linux).

Бібліотека OpenCV (Open Source Computer Vision Library) – це бібліотека комп'ютерного зору з відкритим кодом, яка містить у своєму складі більш ніж 2500 оптимізованих алгоритмів, призначених для задач комп'ютерного зору, аналізу зображень та машинного навчання. Вона реалізована на мовах C/C++, хоча має також інтерфейси розробки для Python, Java та MATLAB.

Бібліотека містить велику кількість інструментів, зрозумілий інтерфейс, що полегшує роботу при вивченні комп'ютерного зору. OpenCV надає доступ до алгоритмів, що використовуються у комп'ютерному зорі такі як виявлення та розпізнавання об'єктів, ідентифікація осіб, класифікація зображень, відстеження рухомих об'єктів, створення 3D-моделей об'єктів, з'єднання окремих зображень в одне для отримання всієї сцени, пошук за зображенням у базі даних, встановлення маркерів для роботи з доповненою реальністю.

OpenCV має низку готових рішень для розв'язання задач високої складності, але за умови відсутності необхідного функціоналу бібліотека дозволяє розробити власний метод, спираючись на існуючі компоненти.

3.2 Програмна реалізація

Попередньо для всіх вхідних зображень застосуємо детектор ключових точок та дескрипторів ORB, кожен дескриптор буде розмірністю 256 значень (нулів або одиниць). В якості прикладу використаємо зображення автомобілів різних марок та моделей: Lamborghini (рис. А.1), Chevrolet (рис. А.2) та Audi (рис. А.3). Для зручності кожне зображення було переведено у чорно-білий спектр та масштабовано до розміру розміром 600×525 пікселів. Для кожного зображення буде визначено по 500 дескрипторів. Зразки визначених ключових точок для еталонних зображень показані на рисунках 3.1 та 3.2.

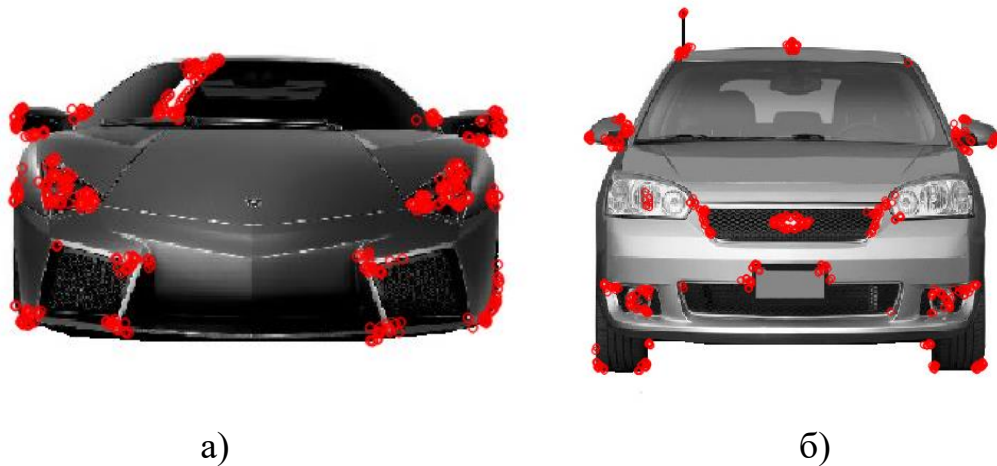


Рисунок 3.1 – Зображення вхідних автомобілів з координатами КТ:

а) автомобіль марки Lamborghini;

б) автомобіль марки Chevrolet



Рисунок 3.2 – Зображення автомобіля марки Audi з координатами КТ

Реалізацію мережі Кохонена напишемо мовою програмування Python із застосуванням парадигми ООП. Клас KohonenNet, який буде виконувати алгоритм. До його базових параметрів можна віднести розмірність сітки нейронів n та m , ваги кожного нейрона $weights$, маркер тренуваності мережі $trained$, коефіцієнт швидкості навчання $learning_rate$ та «щільність» σ для функції сусідства Гауса.

Основними функціями моделі є функція навчання моделі на вибірці дескрипторів fit (лістинг 3.1) та функція класифікації $classify$ (лістинг 3.2).

Лістинг 3.1 Реалізація тренування мережі Кохонена:

```
def fit(self, train_samples, limit, dist_function, n_iterations=100,
with_neighbours=False):
    learning_rate = self.learning_rate
    for iteration in np.arange(1, n_iterations+1):
        for sample in train_samples.astype(float):
            sample = np.array(sample)
            index = self.get_winner_index(sample, dist_function, limit)
            if index == -1: pass
            self.weights[index] = self.weights[index] + learning_rate * (sample -
self.weights[index])
            learning_rate = (1-iteration/n_iterations) * self.learning_rate
        self.trained = True
```

Аргументи: $train_samples$ – список всіх тренувальних зразків даних (дескрипторів), $dist_function$ – функція для розрахунку дистанції між зразками, $n_iteration$ – кількість епох (ітерацій) навчання, $with_neighbours$ – логічна змінна на використання функції сусідства (зокрема за формулою Гауса), $limit$ – значення порогу, за яким будуть видалятися хибні дескриптори. Таким чином лістинг 3.1 демонструє реалізацію навчання нейрона за моделлю «Переможець отримує все».

Лістинг 3.2 Реалізація функції класифікації завдяки мережі Кохонена:

```
def classify(self, test_samples, dist_function, limit):
    if self.trained:
        class_list = np.empty(shape=(len(test_samples)), dtype=np.int8)
        for i, sample in enumerate(test_samples):
            index = self.get_winner_index(sample, dist_function, limit)
            if index == -1: pass
            class_list[i] = index
        return class_list
    else:
        raise Exception("Kohonen network isn't trained!")
```

Аргументи: *test_samples* – список всіх тренувальних зразків (дескрипторів), *dist_function* – функція для розрахунку дистанції між зразками, *limit* – значення порогу, за яким будуть видалятися хибні дескриптори.

При навчанні (2.11) та безпосередньому виконанні алгоритму моделлю розраховується відстань між дескрипторами та вагами нейронів, а потім помножується на коефіцієнт навчання (та функцію сусідства), то необхідно, щоб можна було помножити приріст Δw_{ij} на число з плаваючою комою (аби уточнити нові ваги нейронів). Задля цього застосуємо в якості метрики евклідову відстань.

3.3 Тестування розробленої моделі

Коефіцієнт швидкості навчання є одним з важливих параметрів ефективності роботи мережі Кохонена, питання про оптимальність якого розглядав також сам автор алгоритму. Для підбору оптимального

початкового значення α_0 (2.13) в рамках даної роботи експериментальним шляхом було виявлено, що найкращими значеннями, при фіксованій кількості ітерацій навчання у 100 повторень, є 0,3 (рис. 3.3), коли показники Accuracy та Recall досягають глобального максимуму. Для подальших досліджень застосуємо значення коефіцієнту навчання саме 0.3.

На рисунку 3.3 зображено графіки показників Accuracy і Recall, відповідно до кількості вірно класифікованих зображень, після процесу голосування між дескрипторами, кожен з яких було помічено мережею Кохонена.

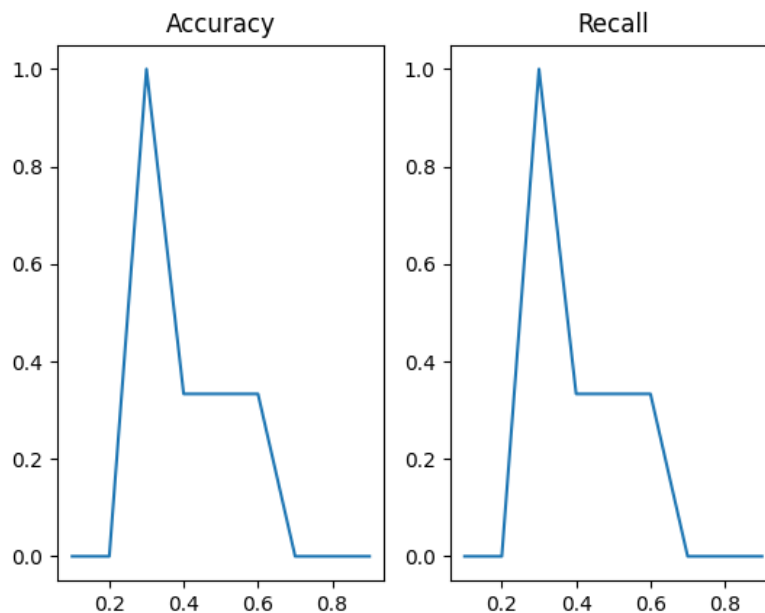


Рисунок 3.3 – Графік ефективності моделі при різних показниках коефіцієнта навчання

Так як попередня ініціалізація ваг нейронів має не менш важливий внесок на результати навчання, ніж коефіцієнт навчання, було проведено дослід, в якому ваги кожного нейрона було проініціалізовано дескриптором кожного зразка зображень однакового індексу, а також середніми значеннями серед усіх дескрипторів зразка (3.1). Кожен клас зображення визначений одним нейроном.

Таблиця 3.1 – Результат класифікації моделі, ініціалізованої дескрипторами з однаковим номером

| | Lamborghini | Chevrolet | Audi | Accuracy | Recall |
|-------------|-------------|------------|------------|----------|--------|
| Lamborghini | 210 | 116 | 174 | 0,42 | 0,42 |
| Chevrolet | 153 | 190 | 157 | 0,38 | 0,38 |
| Audi | 217 | 64 | 219 | 0,43 | 0,43 |

Таблиця 3.2 – Результат класифікації моделі, ініціалізованої середніми значеннями дескрипторів кожного зображення

| | Lamborghini | Chevrolet | Audi | Accuracy | Recall |
|-------------|-------------|------------|------------|----------|--------|
| Lamborghini | 210 | 116 | 174 | 0,42 | 0,42 |
| Chevrolet | 153 | 190 | 157 | 0,38 | 0,38 |
| Audi | 217 | 64 | 219 | 0,43 | 0,43 |

Так як деякі дескриптори перевищили поріг відстані, то вони були відкинуті, через що кількість класифікованих дескрипторів може бути меншою, ніж загальна.

Також було проведено порівняння класифікації моделями без та з навчанням. Тестування проводилося на зображеннях, повернених на 30° проти часової стрілки (рис. А.4, рис. А.5, рис. А.6). В цьому тесті обидві моделі були ініціалізовані середніми значеннями дескрипторів стандартних зображень. Модель без навчання класифікує зображення на основі відстані від кожного нейрона до середніх значень дескрипторів зображень еталонів. Навчена мережа Кохонена була тренувана на вибірці з усіх дескрипторів

стандартних зображень при початковому коефіцієнті навчання $\alpha_0 = 0,3$, порозі $\delta = 15$ та кількості ітерацій 100.

В результаті дослідження моделі без навчання та з навчанням загалом класифікували всі зображення вірно (рис. 3.4, табл. Б.3). Модель із навчанням має значний відсоток хибно класифікованих дескрипторів через близькість центрів кластерів. Середня точність та повнота класифікації дескрипторів становить 0,406, що є слабким показником. Однак дана мережа Кохонена має лише 3 нейрони, кожний з яких відповідає за зображення цілком. Тому при збільшенні кількості нейронів на клас можна покращити результат.

| TEST 1 | TEST 2 | | |
|-----------------|-------------------|-----------------|-----------------|
| ММ без навчання | ММ із навчанням | | |
| Accuracy: 1.0 | Accuracy: 1.0 | | |
| Recall: 1.0 | Recall: 1.0 | | |
| Класи | Class Lamborghini | Class Chevrolet | Class Audi |
| [1 2 3] | [1 2 3] | [1 2 3] | [1 2 3] |
| Результат | [202 118 180] | [167 180 153] | [215 54 231] |
| [1 2 3] | Accuracy: 0.404 | Accuracy: 0.36 | Accuracy: 0.462 |
| | Recall: 0.404 | Recall: 0.36 | Recall: 0.462 |

Рисунок 3.4 – Результат класифікації повернутих зображень двома моделями

ВИСНОВКИ

У рамках кваліфікаційної роботи було розроблено метод класифікації зображень за допомогою мережі Кохонена на множині дескрипторів ключових точок.

У роботі було розглянуто методи розпізнавання образів та класифікації зображень. Проведено аналіз детекторів ключових точок та їх дескрипторів на зображеннях. У даній кваліфікаційній роботі для пошуку КТ та побудови дескрипторів використовувався детектор бінарних дескрипторів ORB, через стійкість до геометричних перетворень та швидкодію обчислення.

Для класифікації зображень було використано модель конкурентного навчання на прикладі мережі Кохонена. Розроблений програмний застосунок було протестовано на зразках зображень автомобілів різних світових марок. Для оцінки результативності моделі було використано показники якості класифікації *Accuracy* та *Recall* дорівнюють одиниці у випадку класифікації еталонних зображень, а також при класифікації зображень із геометричним перетворенням. Також було порівняно ефективність мережі Кохонена із моделлю без навчання на базі центрів кластерів дескрипторів бази еталонів.

Для покращення ефективності запропоновано впровадження структурного аналізу еталонних даних. Це дає можливість подальшого скорочення обсягів обчислень шляхом введення логічного аналізу для компонентів бази еталонів. Також можливе впровадження системи центрів для кожного еталону та статистична обробка множини дескрипторів, що дозволить більш детально розрізняти зразки між собою.

Практична актуальність роботи полягає у побудові моделі класифікації у структурованому просторі даних, підтвердженні працездатності запропонованих модифікацій на прикладах зображень, створенні програмного застосунку для впровадження розробленої моделі класифікації у системах комп'ютерного зору.

Результати роботи апробовано у вигляді тез доповіді під час Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI столітті» [36].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Путятин, Е.П., Аверин С.И. (1990) Обработка изображений в робототехнике. *Москва: Машиностроение*, 320 с.
2. Осовский, С. (2002). Нейронные сети для обработки информации. *Москва: Финансы и статистика*, 244 с.
3. Кохонен, Т. (2008). Самоорганизующиеся карты. *Москва: БИНОМ. Лаборатория знаний*, 655 с.
4. Флах, П. (2015) Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. *Москва: ДМК Пресс*, 400 с.
5. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ. 124 с.
6. Шапиро, Л., Стокман, Д., Богуславский, А. А., & Соколов, С. М. (2013). Компьютерное зрение. *Москва: БИНОМ. Лаборатория знаний*, 762 с.
7. Гороховатский, В. А., Ересько, Ю. Н., Путятин, Е. П., & Стрельченко, В. И. (1990). Локализация объектов на изображениях визуальных сцен. *Автометрия*, 6, 3-7.
8. 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
9. 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.
10. Gorokhovatsky, V.O. and Gadetska, S.V., (2019) Determination of Relevance of Visual Object Images by Application of Statistical Analysis of Regarding Fragment Representation of their Descriptions. *Telecommunications and Radio Engineering*, 78 (3), pp. 211–220.

11. Gorokhovatskyi V., Gadetska S., Ponomarenko R. (2020) Recognition of Visual Objects Based on Statistical Distributions for Blocks of Structural Description of Image. Lecture Notes in Computational Intelligence and Decision Making. Proceedings of the XV International Scientific Conference “Intellectual Systems of Decision Making and Problems of Computational Intelligence” (ISDMCI’2019), Ukraine, May 21–25, 2019, pp. 501-512.

12. Duin, R. P., & Pekalska, E. (2007). The science of pattern recognition. Achievements and perspectives. In Challenges for computational intelligence (pp. 221-259). *Springer*, Berlin, Heidelberg.

13. 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.

14. Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1), pp. 119-139.

15. Гороховатський, В. О., & Гадецька, С. В. (2020). Статистичне оброблення та аналіз даних у структурних методах класифікації зображень. *Харків: ФОП Панов А.Н.*, 125 с

16. 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.

17. Gorokhovatsky, V.A., (2014), Structural analysis and intellectual data processing in computer vision, *SMIT, Kharkiv*, p 316.

18. Путятін Є.П., Гороховатський В.О., Матат О.О. (2006) Методи та алгоритми комп’ютерного зору: навч. посібник.

19. Гороховатский В.А. (2003) Распознавание изображений в условиях неполной информации, *Харьков: ХНУРЭ*, – 112с.

20. Gorokhovatskyi, O., Gorokhovatskyi, V., & Peredrii, O. (2018). Analysis of application of cluster descriptions in space of characteristic image features. *Data*, 3(4), p. 52.

21. 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.

22. Гороховатський, В.О., Творошенко, І.С. (2021) Методи інтелектуального аналізу та оброблення даних: навч. посібник. Харків: ХНУРЕ.

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

24. 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*. 1(47), С. 93–98.

25. Tuytelaars, T. & Mikolajczyk, K. (2008). K.: Local invariant feature detectors: A survey. *FnT Comp. Graphics and Vision*, pp. 177-280.

26. 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.

27. 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.

28. 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.

29. 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.

30. Gorokhovatskyi V., Putyatin Y., Gorokhovatskyi O, Peredrii O. (2018) Quantization of the Space of Structural Image Features as a Way to Increase Recognition Performance. *The Second IEEE International Conference on DataStream Mining & Processing 21-25 August 2018, Lviv, Ukraine.* – pp. 464 – 467.

31. M. A. Ahmad, V. Gorokhovatskyi, I. Tvoroshenko, N. Vlasenko, S. Kh. 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.

32. Scherer, R. (2020). Computer vision methods for fast image classification and retrieval. *Springer International Publishing.*

33. Gorokhovatsky V.A. Putyatin Y. P. (2009) Image Likelihood Measures of the Basis of the Set of Conformities. *Telecommunications and Radio Engineering*, 68 (9), p. 763–778.

34. Gorokhovatskiy, A. V., Gorokhovatskiy, V. A., Vlasenko, A. N., & Vlasenko, N. V. (2014). Quality Criteria for Multidimensional Object Recognition Based Upon Distance Matrices. *Telecommunications and Radio Engineering*, 73(18), pp. 1661 – 1670.

35. Гороховатский В.А., Передрий Е.О. (2009) Корреляционные методы распознавания изображений путем голосования систем фрагментов. *Радіоелектроніка. Інформатика. Управління*, №1(20), с.74–81.

36. Клінов, В.О. (2022, подано до друку). Класифікація зображень з використанням мережі Кохонена. *Радіоелектроніка і молодь у XXI столітті. Харків: ХНУРЕ.*