

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 другий (магістерський)

ДОСЛІДЖЕННЯ МЕТОДУ СТРУКТУРНОЇ КЛАСИФІКАЦІЇ
ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ ЗАСОБІВ НЕЧІТКОЇ
КЛАСТЕРИЗАЦІЇ ДАНИХ

(тема)

Виконав:

студент 2 курсу, групи ІНФМ-20-1

Єрмоєнко В.М.

(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

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

Допускається до захисту

Зав. кафедри

(підпис)

Кобилін О.А.

(прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

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

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Єрьоменку В'ячеславу Миколайовичу
(прізвище, ім'я, по батькові)1. Тема роботи «Дослідження методу структурної класифікації зображень з використанням засобів нечіткої кластеризації даних»затверджена наказом по університету від « 22 » жовтня 2021 року №
1574Ст.2. Термін подання студентом роботи до екзаменаційної комісії 29 листопада 2021 р.3. Вихідні дані до роботи Моделі отримання структурних описів зображень; математичні моделі методів FAST, ORB, BRIEF; теорія класифікації зображень; засоби нечіткої кластеризації даних; об'єктно орієнтована мова програмування Python; дистрибутив мови програмування Python Anaconda; бібліотека комп'ютерного зору OpenCV, інтегроване середовище для програмування PyCharm.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Огляд методів детектування та обробки дескрипторів ключових точок зображення.

2. Аналіз методів отримання значень дескрипторів зображення.

3. Розробка комп'ютерної моделі для нечіткої кластеризації зображень на основі їх структурних описів.

4. Проведення досліджень над розробленим програмним застосунком.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми розпізнавання зображень у системах комп'ютерного зору; постановка задачі класифікації візуальних об'єктів із застосуванням кластерних центрів; мета роботи; дослідження апарату нечіткої кластеризації даних; тестові зображення; демонстрація роботи програми; результати досліджень, висновки.

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

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
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	22.10.2021	
2	Аналіз завдання, підбір літератури	22.11.21-26.11.21	
3	Аналіз літератури з досліджуваної проблеми	26.11.21-01.12.21	
4	Аналіз технічних засобів	01.11.21-07.11.21	
5	Розробка методу	07.11.21-11.11.21	
6	Програмна реалізація	11.11.21-16.12.21	
7	Оформлення пояснювальної записки	16.11.21-23.12.21	
8	Перевірка на плагіат	25.11.2021	
9	Рецензування	25.11.2021	
10	Підготовка презентації та доповіді	26.11.2021	
11	Занесення роботи в електронний архів	27.11.2021	
12	Попередній захист кваліфікаційної роботи	08.12.2021	

Дата видачі завдання 22 жовтня 2021 р.

Студент _____
(підпис)

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

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 67 с., 23 табл., 11 рис., 1 дод., 44 джерела.

КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, КЛЮЧОВА ТОЧКА, ДЕТЕКТОР, ДЕСКРИПТОР, СТРУКТУРНИЙ ОПИС, БІНАРНИЙ АНАЛІЗ, СТАТИСТИЧНИЙ ЦЕНТР, НЕЧІТКА КЛАСТЕРИЗАЦІЯ.

Об'єктами дослідження є методи кластеризації щодо зображень із набору – «CelebaFaces Attributes Dataset (CelebA)».

Метою дослідження є розробка алгоритмів самоорганізації для навчання нечіткої мережі із застосуванням алгоритму нечіткої самоорганізації «C-means» для забезпечення результативної класифікації зображень за їх структурним описом.

Розглянуто методи класифікації візуальних об'єктів із використанням засобів нечіткої самоорганізації із застосуванням статистичних характеристик опису, побудованого у вигляді множини дескрипторів ключових точок.

У результаті роботи проведено дослідження, здійснена програмна реалізація методу класифікації зображень із використанням засобів нечіткої кластеризації, виявлено особливості їх впровадження для багатовимірних даних описів зображень.

IMAGE CLASSIFICATION, KEY POINT, DETECTOR, DESCRIPTOR, STRUCTURAL DESCRIPTION, BINARY ANALYSIS, STATISTICAL CENTER, FUZZY CLUSTERING METHODS.

The objects of research are clustering methods for images from the set – «CelebaFaces Attributes Dataset (CelebA)».

The aim of the study is to develop self-organization algorithms for training a fuzzy network using the fuzzy self-organization algorithm «C-means» to ensure effective classification of images according to their structural description.

Methods of classification of visual objects using fuzzy self-organization tools using statistical characteristics of a description constructed in the form of a set of key point descriptors are considered.

As a result of the work, a study was conducted, a software implementation of the image classification method was carried out using fuzzy clustering tools, and the features of their implementation for multidimensional image description data were identified.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ	7
1 Аналіз методів розпізнавання візуальних об'єктів	9
1.1 Аналіз сучасних методів класифікації даних.....	9
1.2 Методи виявлення ключових точок і їх дескрипторів	12
1.3 Методи нечіткої сомоорганізації.....	16
1.4 Постановка задачі дослідження	19
2 Класифікація зображень на підставі структурного опису	21
2.1 Аналіз характеристик методів Data Mining.....	21
2.2 Аналіз методів кластеризації	26
2.3 Аналіз алгоритмів самоорганізації	30
3 Результати дослідження з використанням комп'ютерної моделі	37
3.1 Обґрунтування вибору програмного середовища	37
3.2 Особливості програмної реалізації.....	40
3.3 Аналіз результатів програмного моделювання	48
Висновки.....	60
Перелік джерел посилання	61
Додаток А Тестові зображення.....	66

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

ОТ – особлива точка

КТ – ключова точка

ORB – Oriented FAST and Rotated BRIEF

FAST – Features from Accelerated Segment Test

BRIEF – Binary Robust Independent Elementary Features

CelebA – CelebaFaces Attributes Dataset

ВСТУП

У сучасних системах комп'ютерного зору набули поширення технології класифікації чималої кількості даних, де інформація про досліджувані об'єкти подається у вигляді множини ключових точок зображення з потужністю в 100-500 елементів [1-5].

Кожна ключова точка зображення відображає властивості околу координат, та описується вектором – дескриптором розмірністю компонентів в 32 байти, або ж 256 бітів [6-13]. Виникає необхідність застосування апарату інтелектуального аналізу у зв'язку зі значними об'ємними потоками даних, для виявлення істотних закономірностей опису окремих візуальних об'єктів, з метою забезпечення результативної класифікації зі скороченням обчислювальних витрат.

Відомі статистичні методи покривають лише частину потреб по обробці даних [14], і для їх використання необхідно мати чітке уявлення про закономірності аналізованих візуальних об'єктів. У такій ситуації методи інтелектуального аналізу даних набувають особливої актуальності. Їх основна особливість полягає у встановленні наявності та характеру прихованих закономірностей в даних.

Серед методів інтелектуального аналізу даних особливе місце займають класифікація та кластеризація.

Класифікація при відомому заздалегідь угрупованню даних на підмножини (класи) встановлює закономірність, за якою дані групуються саме таким чином.

Кластеризація, ґрунтуючись на встановленій інформації щодо схожості елементів, визначає кластери, в які групуються вхідні дані [15-21]. У широкому колі прикладних завдань знайшли своє застосування методи нечіткої кластеризації, в яких елементи вхідної вибірки вважають належними

одночасно до всіх кластерів, а рішення про ступінь належності до того чи іншого кластеру виносять на підставі значення функції приналежності.

Кластерне представлення є результатом попереднього навчання системи розпізнавання на конкретній купі зразків, які складають базис для розпізнавання [22].

Векторну трансформацію структурного опису-множини можна подати як апроксимацію простору ознак системою кластерів. У результаті такого уявлення стає можливим більш продуктивно обчислювати релевантність на структурах даних типу вектор-множина і вектор-вектор. У той же час виникає необхідність детального вивчення особливостей і параметричного управління такою трансформацією з точки зору результативності побудованих на її основі процедур розпізнавання.

Нечітка кластеризація – одна з найбільш розвинутих сучасних методик інтелектуального аналізу даних, однак, традиційні методи нечіткої кластеризації не дають прийнятних рішень на даних зі складною внутрішньою структурою. Це пов'язано з низкою припущень, які закладаються в ці методи: кластери мають задану форму і особливу внутрішню точку – центр кластера; розбиття визначається, виходячи із взаємозв'язків між центрами кластерів і даними.

Актуальність дослідження полягає у створенні ефективних методів оброблення візуальної інформації задля підвищення результативності і розширення функціональності сучасних систем комп'ютерного зору. Особлива увага приділяється вивченню структури та змісту вхідних даних, які безпосередньо впливають на показники функціонування системи розпізнавання.

1 АНАЛІЗ МЕТОДІВ РОЗПІЗНАВАННЯ ВІЗУАЛЬНИХ ОБ'ЄКТІВ

1.1 Аналіз сучасних методів класифікації даних

Найбільш часто розв'язуваними завданнями Data Mining є класифікація [23].

Класифікація даних в інтелектуальному аналізі даних – це поширений метод, який допомагає організувати складні і великі набори даних, які впорядковані по деякому принципу, та мають одну або декілька подібних класифікаційних властивостей, для визначення відмінності або подібності між цими об'єктами [24-26].

Метод передбачає використання алгоритмів, які можуть бути легко адаптовані для поліпшення якості даних. В зв'язку з цим контрольоване навчання тісно пов'язане з процесом класифікації в інтелектуальному аналізі даних. Кінцевою метою класифікації являється зв'язування змінної з спостережуваними змінними. Фактична змінна повинна бути «якісного» типу.

Класифікація потребує дотримання наступних умов:

- на кожному кроці поділу, необхідно застосовувати тільки одну основу;
- розподіл має бути пропорційним;
- розподіл має бути послідовним;
- члени поділу повинні взаємно викреслювати один одного, їх обсяги не повинні перехрещуватися.

Модель класифікації в інтелектуальному аналізі даних використовується, коли змінна фокусу є більш «якісною». Існує кілька різних типів класифікації, кожен з цих алгоритмів використовується для вилучення корисної інформації з набору даних [27]. Найбільш затребуваними сьогодні є сім типів алгоритмів класифікації.

Алгоритм 1. Класифікація структурованих даних.

Метод часто виконується для неструктурованих і структурованих даних. Зазвичай використовується, коли дані необхідно ретельно розділити на кілька класів. Кінцева мета цього методу – визначити класи та класифікувати, де можуть бути розміщені нові набори даних. Загальними іменами, які можна почути в рамках структурованих даних, будуть функція, класифікатор, модель класифікації та двійкова класифікація;

Алгоритм 2. Логістична регресія.

Широко використовуваний алгоритм машинного навчання, який використовується для класифікації складних наборів даних. При застосуванні цього алгоритму враховуються ймовірності. Ймовірність можливого результату конкретного випробування оцінюється і моделюється з використанням логістичної функції. Алгоритм надзвичайно ефективний і призначений для певної мети, це допомагає зрозуміти змінні, які можуть вплинути на результат конкретного інциденту, Однак передбачення виконуються тільки в двійкових файлах. Це означає, що провісники повинні бути незалежні один від одного.

Алгоритм 3. Наївний Байєс.

Алгоритм фокусується на припущенні про пару незалежних функцій. Наївний метод Байєса відмінно працює в реальних сценаріях, таких як, фільтрація спаму і класифікація документів – це два місця, де вичерпно використовується наївний Байєс. Алгоритм робить особливим те, що є необхідність в мінімальній інформації або навчальних даних, це позбавляє від багатьох додаткових параметрів. Алгоритм являється одним із найшвидших.

Алгоритм 4. Стохастичний градієнтний спуск.

Це ефективний і простий підхід, що відповідає потребам лінійних моделей даних. Алгоритм надзвичайно корисний, коли є великі вибірки, він налаштований для обробки функцій втрат. Також в ньому передбачені штрафи за класифікацію при інтелектуальному аналізі даних. Алгоритм

являється складним, тому що він має кілька гіперпараметрів, і логіка надзвичайно чутлива до таких понять, як масштабування об'єктів.

Алгоритм 5. K -найближчих сусідів.

Відомий як лінива форма навчання. Він не фокусується на внутрішніх моделях, тим не менш, він зберігає кілька примірників даних для навчальних цілей. Алгоритм класифікації розраховується з використанням простої системи голосування, яка фокусується на K -найближчих сусідах по кожній координаті в системі. Алгоритм надзвичайно надійний і простий, коли справа доходить до навчання шуму, що являється більш ефективним коли задіяні величезні обсяги даних.

Алгоритм 6. Дерева рішень.

Алгоритми, які фокусуються на атрибутах даних, які об'єднуються з класами. Дерево рішень генерує послідовність правил, ці правила використовуються для класифікації. У порівнянні з багатьма іншими методами дерева рішень легко візуалізувати і розуміти, що не вимагає великої підготовки даних. За допомогою дерев рішень можна обробляти як категоріальні, так і числові дані.

Алгоритм 7. Випадкові ліси.

Дерева часто розширюються, утворюючи випадкові ліси. Це класифікатори з метаоцінками, які можуть об'єднувати кілька дерев рішень в одне. Роль випадкового лісу полягає в підвищенні точності елементів управління і моделей, що використовуються для складання прогнозів.

Проблеми класифікації присутні в кожній галузі. Мета процесу класифікації полягає в тому, щоб побудувати модель, яка використовує атрибути, що в якості входних параметрів вже прогнозовані та отримує значення залежного атрибута. Процес класифікації полягає в розбитті множини об'єктів на класи за певним критерієм.

1.2 Методи виявлення ключових точок і їх дескрипторів

Для виявлення опису ознак зображення необхідно прив'язуватися до його локальних особливостей – особливих точок [1-5].

Ключовою, або особливою точкою називають таку точку зображення, яка з великою часткою ймовірності буде перебувати на іншому зображенні того ж об'єкта [1-5].

Процес пошуку особливих точок здійснюється за допомогою детектора, метода вилучення особливих точок з зображення, який повинен щодо перетворень зображень забезпечити інваріантність знаходження одних і тих же ключових точок [28, 29]. Детектор дозволяє визначити координати КТ. Для встановлення, яка з ключових точок одного зображення відповідає ключовій точці іншого, необхідно використовувати дескриптор [30].

Особлива точка – це точка зображення, яка задовольняє ряду таких властивостей:

- інваріантність – (invariance) – особливі точки повинні мати стійкість до повороту, зміни масштабу зображення і зміни ракурсу зйомки;
- кількість (quantity) – кількість виявлених особливих точок повинна забезпечувати необхідну їх кількість для виявлення об'єктів;
- стійкість (repeatability) – зміна яскравості, контрастності та колірної гами не повинні мати вплив на місце особливої точки на об'єкті або сцені;
- визначеність (distinctness) – особливість повинна виділятися на тлі серед сусідніх точок;
- стабільність (stability) – зашумленість зображення, на роботу детектора, що не перевищує певний поріг, не повинна впливати;
- інтерпретованість (interpretability) – для подальшої роботи, особливі точки повинні бути представлені в формат придатному для оброблення.

Ключові точки інваріантні щодо зміни обертання зображення і масштабу, а також часто інваріантні до зміни освітленості і 3D-точки зору камери.

Етапи алгоритму, які використовуються для отримання характеристик зображення (рис. 1.1).

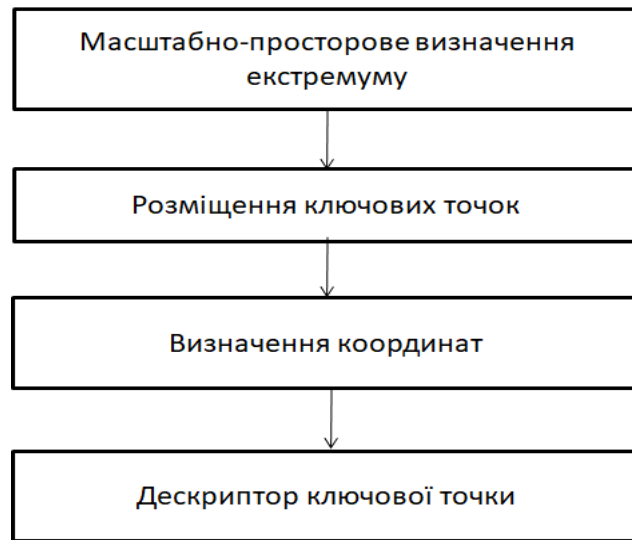


Рисунок 1.1 – Основні етапи алгоритму, який використовується для отримання ключових точок зображення

На першому етапі виконуються розрахунковий аналіз позиції зображення та його масштабу. Ефективність методу досягається використанням функції різниці гаусіанів для визначення потенційно цікавих точок, симетричних відносно орієнтації і масштабу.

На другому етапі для знаходження координат і масштабу на кожному потенційному місці описують детальну модель розміщення ключових точок. Ключові точки вибираються на основі їх ступеня стабільності.

На третьому етапі на основі локального напрямку градієнта зображення привласнюють одну або більше координат кожній ключовій точці. Всі подальші етапи виконуються над даними зображення, які були трансформовані відповідно до заданих координат, масштабом і місцем розташування кожної характеристики, тим самим забезпечуючи інваріантність щодо цих перетворень.

На четвертому етапі вимірюються локальні градієнти зображення при вибраному масштабі в околу кожній ключовій точки. Градієнти

перетворюють в уявлення, яке дозволяє на важливих рівнях змінювати освітленість і спотворювати локальну форму.

Дескриптор – опис особливої точки, що визначає особливості її околиці, являє собою бінарний або числовий вектор певних параметрів [6]. Довжина вектора і вид параметрів визначаються застосуванням алгоритму.

Ідея бінарних дескрипторів полягає в тому, щоб навколо особливої точки виконати опис області двійковий рядком, отриманої шляхом попарного порівняння яскравості пікселів в заданій області.

До сучасних дескрипторів особливих точок відносять такі: LATCH (Learned Arrangements of Three Patch Codes), BRIEF (Binary Robust Independent Elementary Features), FREAK (Fast Retina Keypoint), A-KAZE (Accelerated- KAZE), ORB (Oriented FAST and Rotated BRIEF), BRISK (Binary Robust Invariant Scalable Keypoints) [17, 19, 31].

Алгоритм ORB (Oriented FAST and Rotated BRIEF) став розвитком ідей алгоритмів FAST і BRIEF, в якому зроблена спроба поліпшити продуктивність BRIEF при повороті зображення. У методі ORB запропоновано спочатку виконати розрахунок орієнтації особливої точки і в відповідності з отриманою орієнтацією робити бінарні порівняння.

Алгоритм працює наступним чином:

Крок 1. З використанням деревовидного швидкого алгоритму FAST на оригінальному документі і на кількох зображеннях з піраміди зменшених зображень виявляються особливі точки [32].

Крок 2. Для встановлення точок виконується обчислення міри Харріса, кандидати з низьким значенням міри відкидаються [31].

Крок 3. Розраховується кут орієнтації особливої точки, для цього, напочатку обчислюються моменти яскравості для околу особливої точки:

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y), \quad (1.1)$$

де x, y – піксельні координати;

I – яскравість.

Після виконується обчислення куту орієнтації особливої точки:

$$\theta = a \tan 2(m_{01}, m_{10}). \quad (1.2)$$

В результаті виходить напрямок для околу ОТ.

Крок 4. Маючи кут орієнтації, для бінарних порівнянь, порядок точок в дескрипторі BRIEF, змінює напрям у відповідність з цим кутом (рис. 1.2).

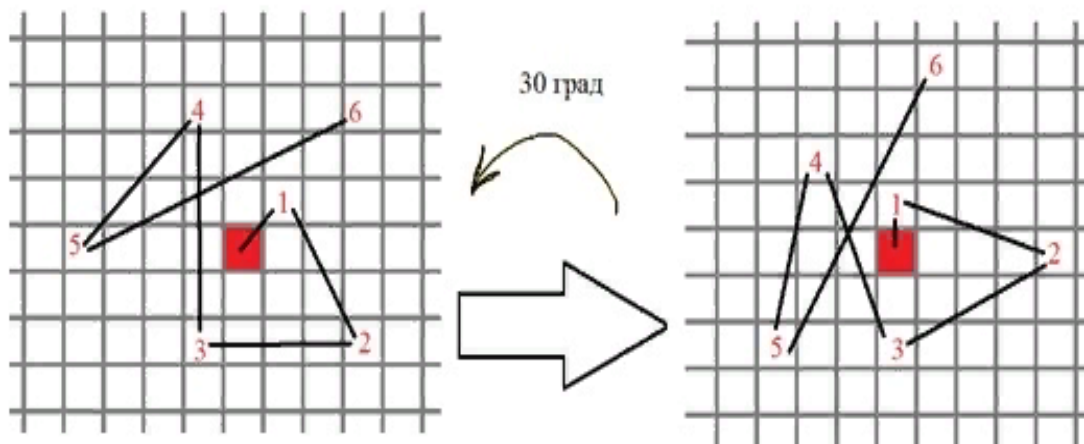


Рисунок 1.2 – Поворот дескриптора BRIEF

Нові положення для точок бінарних тестів обчислюються так:

$$\begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = R(\theta) \begin{pmatrix} x_i \\ y_i \end{pmatrix}. \quad (1.3)$$

Крок 5. За отриманими точкам виконується розрахунок бінарного дескриптора BRIEF.

1.3 Методи нечіткої сомоорганізації

Розглянемо алгоритм самоорганізації для навчання нечіткої мережі [27].

Алгоритм самоорганізації приписує вектор x до відповідної групи даних, що подаються центром c_i , з використанням навчання конкурентного типу подібно до того, як це мало місце в мережах з самоорганізацією Кохонена [33-35]. При навчанні цього типу процес самоорганізації стає можливим при пред'явленні вектора x . Базова форма алгоритму самоорганізації дозволяє точно визначити положення центрів сі відповідних груп даних (кластерів), на які поділяється багатовимірний простір. Ці центри в подальшому можуть використовуватися в гібридному алгоритмі навчання нечіткої мережі в якості початкових значень, що істотно прискорює процес навчання і гарантує збіжність рішення до глобального мінімуму.

Алгоритм нечіткої самоорганізації *C*-means.

В мережі існує K нечітких нейронів з центрами в точках c_i ($i=1,2,\dots,K$). Для навчання, первинні значення цих центрів можуть бути вибрані випадковим чином з областей допустимих значень відповідних компонентів векторів x_j ($j=1,2,\dots,p$). Нехай функція фазифікації задана у формі узагальненої функції Гауса, вираженої формулою (1.4):

$$\mu_A(x_i) = \frac{1}{1 + \left(\frac{x_i - c_i}{\sigma_i} \right)^2}, \quad (1.4)$$

де $\mu_A(x_i)$ представляє оператор A_i .

Поданий на вхід мережі вектор x_j буде належати до різних груп, що подаються центрами c_i , в ступені u_{ij} , причому $0 \leq u_{ij} \leq 1$, а сума ступенів відносин до всіх груп дорівнює 1, тож:

$$\sum_{i=1}^K u_{ij} = 1, \quad (1.5)$$

для $j = 1, 2, \dots, p$.

Відповідно до даного подання, функція похибки визначається як сума окремих похибок належності до центрів c_i з урахуванням ступеня належності m_{ij} . Отож:

$$E = \sum_{i=1}^K \sum_{j=1}^p u_{ij}^m \|c_i - x_j\|^2, \quad (1.6)$$

де m – ваговий коефіцієнт, який приймає значення з інтервалу $[1, \infty)$.

Мета навчання з самоорганізацією виступає в підборі таких центрів c_i , щоб забезпечити досягнення мінімуму функцій (1.7) з одночасним дотриманням умов обмеження (1.6) для заданої множини навчальних векторів x_j .

Таким чином появляється завдання мінімізації нелінійної функції (1.7) з p обмеженнями типу (1.6). Вирішення цього завдання зводиться до мінімізації функції Лагранжа, визначеної у вигляді:

$$LE = \sum_{i=1}^K \sum_{j=1}^p u_{ij}^m \|c_i - x_j\|^2 + \sum_{j=1}^p \lambda_j \left(\sum_{i=1}^K u_{ij} - 1 \right), \quad (1.7)$$

де $\lambda_j (j = 1, 2, \dots, p)$ – це множники Лагранжа.

Доведено, що рішення заданого вигляду (1.8) можна представити у вигляді:

$$c_i = \frac{\sum_{j=1}^p u_{ij}^m x_j}{\sum_{j=1}^p u_{ij}^m}, \quad (1.8)$$

і

$$u_{ij} = \frac{1}{\sum_{k=1}^K \left(\frac{d_{ij}^2}{d_{kj}^2} \right)^{\frac{1}{m-1}}}, \quad (1.9)$$

де d_{ij} – це евклідова відстань між центром x_j і вектором, c_i $d_{ij} = \|c_i - x_j\|$.

Так як точні значення центрів c_i на початку процесу не відомі, алгоритм представляється в ітераційному вигляді.

Алгоритм сформульований наступним виглядом:

Крок 1. Виконати випадкову ініціалізацію коефіцієнтів u_{ij} , вибираючи їх значення з інтервалу $[0, 1]$ таким чином, щоб дотримувалася умова (1.6).

Крок 2. Визначити до центрів c_i відповідно до (1.9).

Крок 3. Розрахувати значення функції похибки згідно з виразом (1.7). Якщо її значення виявиться нижче встановленого порогу або якщо зменшення цієї похибки щодо передідуєї ітерації є незначним, то завершити обчислення. Останні значення центрів складають шукане рішення. В іншому випадку перейти до Кроку 4.

Крок 4. Розрахувати нові значення u_{ij} за формулою (1.10) і перейти до Кроку 2.

Така процедура нечіткої самоорганізації називається алгоритмом *C-means*.

Багаторазове повторення ітераційної процедури веде до досягнення мінімуму функції E , який необов'язково буде глобальним мінімум. Якість знайдених центрів, що оцінюється значенням функції похибки E , істотно залежить від попереднього підбору як значень u_{ij} , так і центрів c_i .

Найкращим може бути визнано таке розміщення центрів, при якому вони розташовуються в областях, що містять найбільшу кількість пред'явлених векторів x_j . При такому підборі центрів вони будуть представляти вектори даних c_j з найменшою сумарною похибкою.

Алгоритм підходить, якщо заздалегідь невідоме число кластерів, або необхідно однозначно віднести кожен об'єкт до одного кластера.

Тому початок ітераційної процедури розрахунку оптимальних значень центрів має представлятися процедурою їх ініціалізації. До найбільш відомим алгоритмом ініціалізації відносяться алгоритми пікового групування і різницевого групування даних.

1.4 Постановка задачі дослідження

Класифікація зображень засобами нечіткої кластеризації даних, використовуючи набір структурних описів зображень у бінарному вигляді є актуальним завданням для проведення досліджень з обробки та розпізнавання зображень. Ставиться завдання розробки та моделювання алгоритму класифікації зображень з використанням методу нечіткої кластеризації даних «C-means».

Об'єктами дослідження є методи кластеризації щодо зображень із набору – «CelebFaces Attributes Dataset (CelebA)».

Метою дослідження є розробка алгоритмів самоорганізації для навчання нечіткої мережі із застосуванням алгоритму нечіткої самоорганізації «C-means» для забезпечення результативної класифікації зображень за їх структурним описом.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз існуючих методів детектування ключових точок зображень;
- провести аналіз існуючих методів кластеризації зображень;

- сформуванати множини дескрипторів для кожного з еталонів;
- реалізувати комп'ютерну модель нечіткої кластеризації зображень на основі їх структурних описів;
 - на основі розробленого застосунку провести дослідження класифікації на обраних еталонах;
 - зробити висновки відносно результативності класифікації на основі проведених експериментальних досліджень.

2 КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ НА ПІДСТАВІ СТРУКТУРНОГО ОПИСУ

2.1 Аналіз характеристик методів Data Mining

Розвиток інформаційних технологій, окремо, прогрес у методах збору, зберігання та обробки даних дозволив багатьом організаціям збирати величезні масиви даних, які необхідно досліджувати. Об'єм даних, які потребують обробки, настільки великі, що можливостей експертів не вистачає для проведення розрахунків.

На сьогоднішній день інтенсивно розвивається напрямок, пов'язаний з інтелектуалізацією методів обробки та аналізу даних. Інтелектуальні системи аналізу [23] даних призначені мінімізувати зусилля особи, яка приймає рішення, в процесі аналізу даних, а також в налаштуванні алгоритмів аналізу. Багато систем аналізу даних здатні виявляти приховані закономірності в системі, що піддається аналізу.

Data Mining – це поєднання широкого математичного інструментарію і останніх досягнень у сфері інформаційних технологій. В цій технології гармонійно об'єдналися строго формалізовані методи і методи неформального аналізу, тобто кількісний і якісний аналіз даних.

Основу методів Data Mining утворюють різноманітні методи класифікації, прогнозування та моделювання. До цих методів відносяться статистичні методи такі як: описовий аналіз, факторний аналіз, кореляційний і регресійний аналіз, аналіз часових рядів, дисперсійний аналіз, компонентний аналіз, дискримінантний аналіз. Такі методи, однак, припускають деякі апіорні уявлення про аналізовані дані, що дещо розходиться з цілями Data Mining (виявлення раніше невідомих нетривіальних і практично корисних знань).

Одним з основних призначень методів Data Mining полягає в наочному поданні результатів обчислень, що дозволяє використовувати інструментарій

людьми, які не мають спеціальної математичної підготовки. У той же час, застосування статистичних методів аналізу даних вимагає високого рівня володіння математичною статистикою і теорією ймовірностей.

Знання, що здобуваються методами Data mining, прийнято представляти у вигляді моделей.

До моделі представлення знань відносяться:

- дерева рішень;
- кластер;
- асоціативні правила;
- математична модель.

Методи побудови таких моделей відносяться до області штучного інтелекту. До алгоритмів і методів Data Mining належать:

- метод опорних векторів;
- дерева рішень, символічні правила;
- лінійна регресія;
- штучні нейронні мережі;
- ієрархічні методи кластерного аналізу;
- методи найближчого сусіда і k -найближчого сусіда;
- кореляційно-регресійний аналіз;
- байєсівські мережі;
- неієрархічні методи кластерного аналізу, в тому числі алгоритми k -середніх і k -медіани;
- еволюційне програмування та генетичні алгоритми;
- різноманітні методи візуалізації даних і безліч інших методів;
- метод обмеженого перебору.

Більша кількість аналітичних методів, які використовуються в технології – це відомі математичні методи і алгоритми. Новизна в їх використанні полягає в можливості їх застосування при вирішенні тих чи інших конкретних проблем, обумовлених можливостями програмних і

технічних засобів. Слід зазначити, що більшість методів були розроблені в рамках теорії штучного інтелекту.

Метод представляє собою норму або правило, спосіб, певний шлях, прийом рішень задачі теоретичного, пізнавального, практичного, управлінського характеру.

Різноманітні методи Data Mining характеризуються певними властивостями, які можуть суттєво відзначатись при виборі засобу аналізу даних.

Основні властивості і характеристики методів Data Mining: точність, перевіреність, масштабованість, інтерпретованість, трудомісткість, гнучкість, швидкість і популярність.

Масштабованість – властивість обчислювальної системи, при додаванні до неї обчислювальних ресурсів, забезпечує передбачуване зростання системних характеристик, таких як, швидкості реакції, загальної продуктивності та інші.

У таблиці 2.1 наведена порівняльна характеристика деяких поширених методів. Оцінка кожної з характеристик проведена наступними категоріями, в порядку зростання: надзвичайно низька, дуже низька, низька/нейтральна, нейтральна/низька, нейтральна, нейтральна/висока, висока, дуже висока.

Таблиця 2.1 – Порівняльна характеристика методів Data Mining

Алгоритм	Точність	Масштабованість	Інтерпретованість	Придатність до використання
1	2	3	4	5
Лінійна регресія	Нейтральна	Висока	Висока/нейтральна	Висока
Нейронні мережі	Висока	Низька	Низька	Низька
Метод візуалізації	Висока	Дуже низька	Висока	Висока

Продовження таблиці 2.1

1	2	3	4	5
Дерева рішень	Низька	Висока	висока	Висока/ нейтральна
Нейронні мережі	Висока	Нейтральна	Низька	Висока/ нейтральна
<i>K</i> - найближчого сусіда	Низька	Дуже низька	Висока/ низька	нейтральна
Алгоритм	Трудомісткість	Різномісність	Прудкість	Популярність
Лінійна регресія	Нейтральна	Нейтральна	Висока	Низька
Нейронні мережі	Нейтральна	Низька	Дуже низька	Низька
Метод візуалізації	Дуже висока	Низька	Надзвичайно низька	Висока/ нейтральна
Дерева рішень	Висока	Висока	Висока/ нейтральна	Висока/ нейтральна
Нейронні мережі	Низька/ нейтральна	Нейтральна	Низька/ Нейтральна	Нейтральна
<i>K</i> - найближчого сусіда	Нейтральна/ низька	Низька	Висока	Низька і

Всі методи Data Mining за принципом роботи з вихідними навчальними даними можна поділити на дві основні групи. Верхній рівень у цій класифікації визначається на основі того, чи зберігаються дані після Data Mining або вони фільтруються для подальших досліджень.

При безпосередньому використанні або збереженні вихідні дані зберігаються в явному деталізованому вигляді і використовуються на стадіях прогностичного моделювання або аналізу винятків. При використанні методів цієї групи можуть виникнути складності аналізу великих баз даних.

До групи даних методів відносяться:

- метод найближчого сусіда;
- метод *k* -найближчого сусіда;
- кластерний аналіз.

Для віднесення статистичних методів до Data Mining розділяють два підходи. Перший підхід протиставляє статистичні методи і Data Mining, класичні статистичні методи окремим напрямком аналізу даних. В другому підході, статистичні методи аналізу являються частиною математичного інструментарію Data Mining.

Класифікація даного вигляду поділяється на дві групи методів:

- статистичні методи, засновані на використанні усереднених накопичених даних, які відображені в ретроспективних даних;
- кібернетичні методи, що включають стек різнорідних математичних підходів.

Недоліком використання такої класифікації, є те що і статистичні, і кібернетичні алгоритми тим чи іншим чином орієнтуються на зіставлення статистичного досвіду з результатами дослідження поточної ситуації. Перевагою є її зручність для інтерпретації – класифікація використовується при описі математичних засобів сучасного підходу до вилучення знань з масивів вихідних спостережень (ретроспективних і оперативних [36-40]).

Статистичні методи являють собою чотири взаємопов'язаних розділи:

- попередній аналіз природи статистичних даних (перевірка гіпотези стаціонарності, нормальності, незалежності, однорідності, оцінка виду функції розподілу, її параметрів і т.п.);
- виявлення зв'язків і закономірностей (лінійний і нелінійний регресійний аналіз, кореляційний аналіз та ін.);
- багатовимірний статистичний аналіз (лінійний і нелінійний дискримінантний аналіз, кластерний аналіз, компонентний аналіз, факторний аналіз та ін.);
- динамічні моделі та прогноз на основі часових рядів.

Методи Data Mining класифікуються за завданнями. Відповідно до даної класифікації виділяються дві групи. Перша група – це підрозділ методів Data Mining на вирішальні завдання сегментації, на завдання кластеризації і класифікації, та завдання прогнозування. В другій групі

класифікації за завданнями методи можуть бути спрямовані на вилучення прогнозуючих і описових результатів.

Прогнозуючі методи використовують значення одних змінних для передбачення невідомих (пропущених) або майбутніх значень інших (цільових) змінних, до них відносяться:

- лінійна регресія;
- дерева рішень;
- метод опорних векторів;
- метод найближчого сусіда;
- нейроні мережі.

Описові методи використовуються для знаходження зразків або шаблонів, що описують дані, які піддаються інтерпретації з точки зору аналітика, до них відносяться [41]:

- алгоритм k -середніх;
- k -медіани;
- ієрархічні методи кластерного аналізу;
- методи крос-табличної візуалізації;
- самоорганізовані карти Кохонена.

2.2 Аналіз методів кластеризації

Завдання кластеризації схоже до завдання класифікації і являється її логічним продовженням. Відмінність полягає в тому, що класи досліджуваного набору даних завчасно не визначені.

Кластерний аналіз – це задача розбиття множини об'єктів на групи кластерів [15-21, 23]. Всередині кожної групи об'єкти повинні бути «схожі» між собою, а об'єкти різних групи мають бути як можна більш відмінні один від одного. Головною відмінністю класифікації від кластеризації являється

те, що перелік груп чітко не визначений і встановлюється в процесі роботи алгоритму (табл. 2.2).

Таблиця 2.2 – Порівняння класифікації та кластеризації

Характеристика	Кластеризація	Класифікація
Контрольованість навчання	Неконтрольоване навчання	Контрольоване навчання
Стратегія	Навчання без учителя	Навчання з вчителем
Наявність позначки класу	Мітки класу навчальної множини невідомі	Навчальна множина супроводжується міткою, яка вказує клас, до якого належить спостереження
Підстава для класифікації	Наявність множини даних з метою встановлення існування класів або кластерів даних	Нові дані класифікуються на підставі навчальної множини

В загальному вигляді застосування кластерного аналізу зводиться до наступних етапів:

Етап 1. Для кластеризації відбираються об'єкти вибірки.

Етап 2. Для оцінювання об'єктів у вибірці визначаються множини змінних. За необхідності значення змінних нормалізуються.

Етап 3. Між об'єктами розраховується значення міри подібності.

Етап 4. Для створення кластерів застосовується метод кластерного аналізу.

Етап 5. Представляються результати аналізу.

Після аналізу та отримання результатів можливе коригування обраної метрики і методу кластеризації до отримання оптимального результату.

Розрізняються два основних підходи до кластеризації. Перший метод – чітка кластеризація (або жорстка кластеризація), другий – нечітка кластеризація. Особливістю методу чіткої кластеризації є те, що межа між кластерами є визначеною. Однак у багатьох реальних випадках межі між кластерами не можуть бути чітко визначені. Деякі шаблони можуть належати більш ніж одному кластеру. У таких випадках метод кластеризації забезпечує кращий і більш корисний метод класифікації цих шаблонів.

Для визначення подібності об'єктів використовуються міри відстаней [24]. Для цього, для кожного об'єкта складається вектор характеристик – як правило, це набір числових значень. Однак існують також алгоритми, що працюють з якісними характеристиками.

Після формулювання вектору характеристик, виконується нормалізація, щоб при розрахунку «відстані» всі компоненти давали однаковий внесок. При процесі нормалізації всі значення приводяться до деякого діапазону, наприклад, $[-1, 1]$ або $[0, 1]$. Для кожної пари об'єктів розраховується ступінь схожості між ними. Існує безліч метрик, основні з них:

Евклідова відстань – являється найбільш поширеною функцією відстані, що представляє собою геометричну відстань в багатовимірному просторі:

$$\rho(x, x') = \sqrt{\sum_i^n (x_i - x'_i)^2}. \quad (2.1)$$

Квадрат евклідової відстані застосовується для додання більш віддаленим один від одного об'єктам більшої ваги. Відстань обчислюється наступним чином:

$$\rho(x, x') = \sum_i^n (x_i - x'_i)^2. \quad (2.2)$$

Манхетенська відстань (відстань міських кварталів) – відстань є середнім різниць по координатах. У більшості випадків, міра відстані призводить до таких же результатів, як і для відстані Евкліда. Проте для цього заходу вплив окремих великих викидів зменшується оскільки вони не зводяться в квадрат. Формула для розрахунку манхетенської відстані:

$$\rho(x, x') = \sum_i^n |x_i - x'_i|. \quad (2.3)$$

Відстань Чебишева – відстань являється вигідною, при необхідності визначити два об'єкта як «різні», якщо вони розрізняються по якій-небудь одній координаті. Відстань Чебишева розраховується за формулою:

$$\rho(x, x') = \max(|x_i - x'_i|). \quad (2.4)$$

Ступенева відстань – застосовується при необхідності зменшити або збільшити вагу, що відноситься до розмірності, для якої відповідні об'єкти сильно відрізняються. Ступенева відстань розраховується за наступною формулою:

$$\rho(x, x') = \sqrt[r]{\sum_i^n (x_i - x'_i)^2}, \quad (2.5)$$

де p, r – параметри, визначені користувачем.

Параметр p відповідає за поступове зважування різниць за окремим координатами, параметр r відповідає за прогресивне зважування великих відстаней між об'єктами. Отримана відстань збігається з відстанню Евкліда якщо обидва параметри – p і r – дорівнюють двом.

Оскільки результати кластеризації можуть істотно різнитись при використанні різних метрик, вибір використаного заходу повністю лежить на досліднику.

Алгоритми кластеризації розділяються на плоскі та ієрархічні, чіткі та нечіткі. Ієрархічні алгоритми будують систему вкладеного розбиття, таким чином на виході отримується дерево кластерів, коренем якого є вся вибірка, а розвітвленням найбільш дрібні кластери. Одне розбиття об'єктів на кластери будують плоскі алгоритми.

Чіткі алгоритми кожному об'єкту вибірки ставиться у відповідність номер кластера, кожен об'єкт належить тільки одному кластеру. Нечіткі алгоритми ставлять у відповідність кожному об'єкту набір речовинних значень, які показують ступінь відносин до кластерів об'єкта. Кожен об'єкт відноситься до кожного кластеру з деякою ймовірністю.

2.3 Аналіз алгоритмів самоорганізації

Алгоритм пікового групування [40] був запропонований Р. Єгером і Д.Філевим. В якості міри щільності розміщення вектором x_j в ньому генерується так звані пікові функції. При використанні P вхідних векторів створюється сітка, рівномірно накриває простір векторів x_j . Вузли цієї сітки розглядаються як потенційні центри m_i , і для кожного з них розраховується пікова функція $m(v)$:

$$m(v) = \sum_{j=1}^P \left(\frac{\|v - x_j\|^{26}}{2\sigma^2} \right). \quad (2.6)$$

Коефіцієнт σ – це константа, індивідуально підбирається для кожної конкретної задачі, a b – показники ступеня узагальненої функції Гауса, яка застосовується в нечіткій мережі.

Величина функції $m(v)$ розглядається як оцінка висоти пікової функції. Вона пропорційна кількості векторів x_j , що знаходяться в околиці потенційного центру V . мале значення $m(v)$ свідчить про те, що центр V розташовується в області, в якій зосереджена невелика кількість векторів x_j . Слід звернути увагу, що коефіцієнт має незначний вплив на підсумкові пропорції між $m(v)$ для різних m , тому підбір його величини не є критичним.

Після розрахунку значень $m(v)$ для всіх потенційних центрів серед них відбираються перші (c_1), що мають найбільше значення $m(v)$. Для вибору наступних центрів необхідно перш за все виключити c_1 і вузли, розташовані в безпосередній близькості від c_1 . Це можна зробити шляхом перевизначення пікової функції за рахунок відсікання від неї функції Гауса з центром в точці c_1 . Якщо це знову визначається функцію позначити $m_{new}(v)$, то отримаємо:

$$m_{new}(v) = m(v) - m(c_1) \exp\left(-\frac{\|v - x_j\|^{26}}{2\sigma^2}\right). \quad (2.7)$$

Необхідно звернути увагу, що нова функція $m_{new}(v)$ має нульове значення в точці. Рисунок 2.1 ілюструє типовий процес пікового групування в двовимірному просторі. Помітно, що послідовне відсікання центрів (з максимальними значеннями пікової функції) дозволяє виявляти і усувати чергові центри.

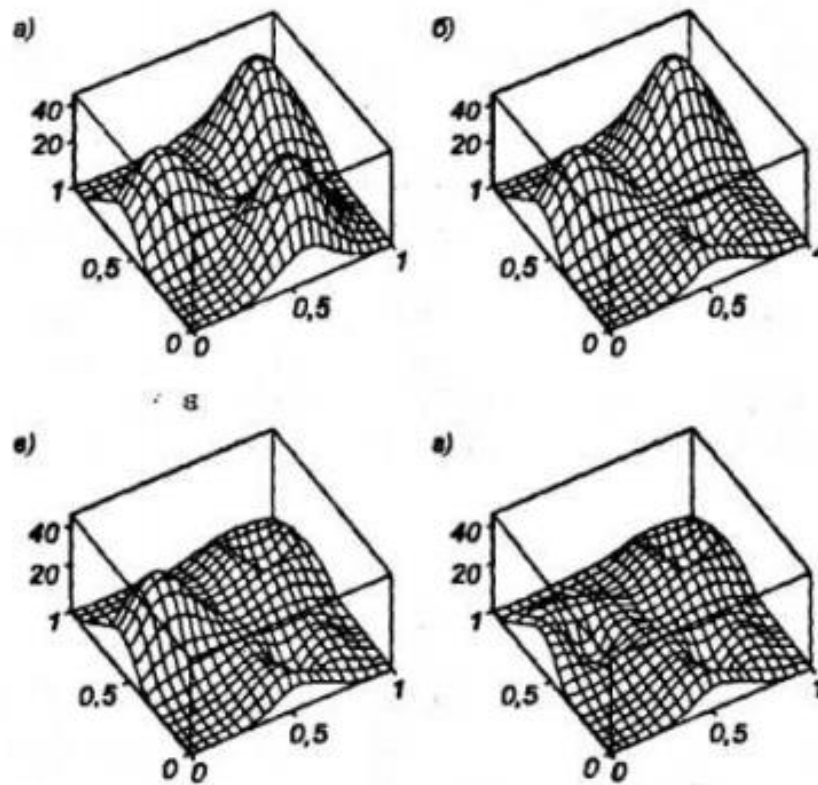


Рисунок 2.1 – Ілюстрація функціонування алгоритму пікового групування

Процес знаходження наступних центрів c_2, c_3, \dots здійснюється послідовно на модифікованих значеннях функції $m_{new}(v)$, одержуваних при виключенні найближчого оточення центру, виявленого на попередньому етапі. Він завершується в момент локалізації всіх центрів, що використовуються в моделі нечіткої мережі. Метод пікового групування ефективний, якщо розмірність вектора x не надто велика. В іншому випадку (при великій кількості компонентів x) число потенційних центрів наростає лавиноподібно, і процес розрахунку чергових пікових функцій стає занадто тривалим, а процедура малоефективною.

Алгоритм різницевого групування даних [34] – це модифікація алгоритму пікового групування, в якому навчальні вектори x_i розглядаються в якості потенційних центрів V . Пікова функція $D(x_i)$ в цьому алгоритмі задається у вигляді:

$$D(x_i) = \sum_{j=1}^p \exp \left(- \frac{\|v - x_j\|^{2b}}{\left(\frac{r_a}{2}\right)^2} \right). \quad (2.8)$$

Значення коефіцієнта r_a визначає сферу сусідства. На значення $D(x_i)$ істотно впливають тільки ті вектори x_j , які розташовані в межах цієї сфери. При великій щільності точок навколо x_i (потенційного центру) значення функції $D(x_i)$ велика. Навпаки, мале її значення свідчить про те, що в околі x_i знаходиться незначна кількість даних. Така точка вважається «невдалим» кандидатом в центри. Після розрахунку значень пікової функції для кожної точки x_j відбирається вектор x , для якого міра щільності $D(x_i)$ виявилася найбільшою. Саме ця точка стає першим відібраним центром c_1 . Вибір наступного центру можливий після виключення попереднього і всіх точок, що лежать в його околі. Так само, як і в методі пікового групування, пікова функція перевизначається у вигляді:

$$D_{new}(x_i) = D(x_i) - (c_1) \exp \left(- \frac{\|v - x_j\|^{2b}}{\left(\frac{r_b}{2}\right)^2} \right). \quad (2.9)$$

При новому визначенні функції D коефіцієнт r_b позначає нове значення константи, що задає сферу сусідства чергового центру. Зазвичай дотримується умова $r_b \geq r_a$. Пікова функція $D_{new}(x_i)$ приймає нульове значення при $x_i = c_1$ і близька до нуля в найближчій околиці цієї точки.

Після модифікації значень пікової функції відшукується наступна точка x , для якої величина $D_{new}(x)$ виявляється максимальною. Ця точка стає

наступним центром c_2 . Процес пошуку чергового центру відновлюється після виключення компонентів, відповідних вже відібраним точкам. Ініціалізація завершується в момент фіксації всіх центрів, передбачених початковою умовою.

Відповідно до описаного алгоритму відбувається самоорганізація безлічі векторів x , що складається в знаходженні оптимальних значень центрів, що представляють безліч даних мінімальною похибкою. Якщо ми маємо справу з безліччю навчальних даних у вигляді пар векторів (x_i, d_i) так, як це відбувається при навчанні з учителем, то для знаходження центрів, соответствующих безлічі векторів d_i , досить сформулювати розширену версію векторів x в формі:

$$x_i \leftarrow [x_i, d_i]. \quad (2.10)$$

Процес групування, що проводиться з пред'явленням розширених векторів x_i , дозволяє визначити також розширені версії центрів c_i . Якщо взяти до уваги, що розмірність кожного нового центру дорівнює сумі розмірностей векторів x і d , то в описі цього центру можна легко виділити частину p , відповідний вектору d . Таким чином можна отримати центри як вхідних змінних, так очікуваних вихідних значень:

$$c_i = [p_i, q_i], \quad (2.11)$$

для $i = 1, 2, \dots, K$.

У разі застосування нечітких правил з одним виходом вектори d і q зводяться до скалярних величин d і q відповідно. Таким чином, при використанні правила виведення Ванга-Менделя процес самоорганізації дозволяє відновити функцію $f(x)$, що апроксимує множину даних (x_i, d_i) для $i = 1, 2, \dots, p$. Зокрема, при введених вище позначеннях формула має вигляд:

$$f(x) = \frac{\sum_{l=1}^K q_l \exp\left(-\frac{\|v - x_j\|^{2b}}{\sigma_1^{2b_l}}\right)}{\sum_{l=1}^K \exp\left(-\frac{\|v - x_j\|^{2b}}{\sigma_1^{2b_l}}\right)}, \quad (2.12)$$

згідно з яким всі центри підбираються оптимальним чином. При цьому інші параметри (b_l, σ_l) , менш критичні для збіжності алгоритму, можуть ефективно підбиратися гібридним методом при невеликій кількості ітерацій. Звичайно, ітераційний процес при реалізації гібридного методу охоплює також і розрахунки координат центрів, однак з урахуванням їх вдалого початкового розміщення зміни, що вносяться в процесі навчання, зазвичай оказиваються дуже незначними.

У результаті проведеного аналізу, у рамках інтелектуального аналізу даних існує безліч методів для кластеризації даних, які відрізняються один від одного як математичним апаратом, так і результатами обробки інформації.

Більшість цих методів вимагає заздалегідь визначити кількість кластерів та щоб ці кластери мали опуклу або округлу форму, та є лінійно роздільними. Але у реальних задачах дотримуватись всіх цих вимог важко, оскільки кластери можуть буди довільної форми або бути лінійно нероздільними.

Для вирішення даних задач існують методи, які засновані на ядерному підході для кластеризації даних. В рамках цього підходу передбачається, що кластери мають довільну форму, але вони не перетинаються у просторі ознак та вся вибірка даних задається заздалегідь, тобто працюють ці підходи у пакетному режимі.

На сьогоднішній день існує безліч методів для кластеризації даних, але всі ці методи чіткі, тобто вони передбачають, що кластери є лінійно

роздільними, та працюють у пакетному режимі. У реальних задачах кластери можуть перетинатись, а кожне спостереження може належати декільком кластерам з відповідною належністю. Для вирішення такої задачі існують методи нечіткої кластеризації, проте сформовані кластери мають округлу форму.

У ситуаціях, коли кластери перетинаються, виникає необхідність розробки методів нечіткої кластеризації для потоків даних. Існує ще одна проблема, що кількість кластерів не відома, особливо, коли дані обробляються в онлайн режимі.

3 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ З ВИКОРИСТАННЯМ КОМП'ЮТЕРНОЇ МОДЕЛІ

3.1 Обґрунтування вибору програмного середовища

У рамках кваліфікаційної роботи був розроблений метод структурної класифікації даних з використанням засобів нечіткої кластеризації даних. Для реалізації було обране – інтегроване середовище розробки для мови програмування Python PyCharm. Це обумовлено тим, що PyCharm володіє рядом таких переваг як:

- поставляється з інтелектуальним редактором коду, розумною навігацією коду, швидким та безпечним рефакторінгом;
- являється інтегрованим з такими функціями, як налагодження, тестування, розгортання, профілювання, віддалена розробка та інструменти бази даних;
- за допомогою Python PyCharm забезпечує підтримку фреймворкі веб-розробки python, JavaScript, CSS, HTML, Angular JS та функцій редагування Live;
- має потужну інтеграцію з ноутбуком IPython, консоллю python та науковим стеком.

IDE максимально адаптується під потреби користувача. За роки розроблення IntelliJ була створена велика кількість плагінів для PyCharm, які забезпечують підтримку додаткових систем контролю версій, інтеграцію з фреймворками і інструментами, а також розширюють можливості редактора, наприклад за рахунок емуляції Vim.

PyCharm забезпечує розумну платформу для розробників, які допомагають їм, коли справа доходить до автоматичного заповнення коду, виявлення помилок, швидкого виправлення тощо. Надає багатофункціональну підтримку, збільшуючи багато факторів економії

коштів. Підтримує міжплатформну розробку, таким чином розробники можуть писати сценарії на різних платформах. PyCharm має гарну функцію настроюваного інтерфейсу, завдяки цьому збільшується продуктивність. Дозволяє створювати окреме середовище Conda і інсталивати потрібні бібліотеки для проекту.

Також важливим елементом є можливість запустити консоль розробника REPL для Python, яка має багато переваг над стандартною консоллю. Серед яких перевірка синтаксису на льоту за допомогою інспекцій, зіставлення лапок і дужок і, звичайно, авто доповнення.

Python є інтерпретованою об'єктно-орієнтованою мовою програмування високого рівня із строгою динамічною типізацією. Його об'єктно-орієнтований підхід та мовні конструкції спрямовані на те, щоб допомогти користувачам писати логічний, чіткий код для великих і малих проектів.

Python – це мова програмування з декількома парадигмами. Повністю підтримуються об'єктно-орієнтоване і структуроване програмування, і багато його функцій підтримують функціональне програмування і аспектно-орієнтоване програмування. Інші парадигми підтримуються за допомогою розширень, включаючи в себе логічне програмування і розробку.

В наявності мається велика кількість модулів, що підключаються до програми, які забезпечують різноманітні додаткові можливості.

Інтерпретатор Python реалізовано практично на всіх платформах та операційних системах. Інтерпретатор написаний на C і вихідний код доступний для будь-яких маніпуляцій. У випадку необхідності є можливість вставити його в свою програму та використовувати як вбудовану оболонку. Або, написавши на C свої доповнення, отримати розширений інтерпретатор з новими можливостями.

Anaconda Distribution є дистрибутивом Python, який працює як менеджер пакетів, менеджер середовища та має колекцію пакетів з відкритим кодом. Основна особливість дистрибутива – оригінальний менеджер дозволу

залежностей conda з графічним інтерфейсом Anaconda Navigator, що дозволяє відмовитися від стандартних менеджерів пакетів, таких, як pip для Python. Дистрибутив завантажується один раз, і вся подальша конфігурація, в тому числі установка додаткових модулів, може проводитися в офлайн режимі. Крім того, забезпечується можливість ведення декількох ізольованих середовищ з роздільним дозволом версійних залежностей в кожній.

Цей набір для обробки Python даних має велику кількість функцій, серед яких можна виділити наступне:

- мультиплатформенність;
- дозволяє встановлювати та керувати пакетами, залежностями та середовищами для обробки даних;
- має такі інструменти для аналізу даних, як numpy, pandas, dask, scikit-learn;
- дозволяє компілювати Python у машинний код для швидкого виконання.

Програмна реалізація детектора ORB відображена у багатьох бібліотеках комп'ютерного зору, таких як OpenCV.

OpenCV (бібліотека комп'ютерного зору) – це бібліотека програмного забезпечення для машинного навчання та комп'ютерного зору з відкритим вихідним кодом. Був створений для забезпечення загальної інфраструктури для додатків комп'ютерного зору і прискорення використання машинного сприйняття в комерційних продуктах. Будучи продуктом з ліцензією BSD, OpenCV спрощує для підприємств використання і зміна коду.

Бібліотека налічує понад 2500 оптимізованих алгоритмів, які включають в себе повний набір як класичних, так і сучасних алгоритмів машинного навчання і комп'ютерного зору.

Він має інтерфейси C++, Python, Java і MATLAB і підтримує Windows, Linux, Android і Mac OS. OpenCV в основному орієнтується на додатки для бачення в реальному часі, використовує інструкції MMX і SSE, коли вони доступні. Існує понад 500 алгоритмів і приблизно в 10 разів більше функцій,

які складають або підтримують ці алгоритми. OpenCV спочатку написаний на C++ і має шаблонний інтерфейс, який легко працює з контейнерами STL.

Єдине, що повинен зробити користувач, працюючи із графічними компонентами, – це визначити характеристики використовуваних знарядь малювання. Вам не буде потрібно стежити за системними ресурсами при створенні, виборі й звільненні знарядь. Канва сама подбає про це.

3.2 Особливості програмної реалізації

У Python в операційній системі Windows реалізовано дослідження та оцінювання ефективності методу структурної класифікації зображень використовуючи засоби нечіткої кластеризації. Визначення особових точок зображення виконується за допомогою алгоритму ORB [7-9]. Отримання та обчислення дескрипторів КТ реалізовано у вигляді програмного забезпечення на основі модуля feature2d бібліотеки OpenCV.

Етап кластеризації зображень при застосуванні таких методів полягає в обчисленні коефіцієнтів приналежності, засобами нечіткої кластеризації C-Means [27], між описами об'єктів, що задані у вигляді множини дескрипторів. Такій підхід дає можливість програмі працювати із візуальними образами, розділяючи їх на кластери за схожими характеристиками, подібно людині. Виникає необхідність провести порівняльний аналіз характеристик дескрипторів, можливостей для виявлення ОТ та їх обчислення. Проведення нечіткої кластеризації, при різних видах вхідних даних, для отримання найкращих результатів.

Метою дослідження є вивчення алгоритмів самоорганізації для навчання нечіткої мережі та застосування алгоритму нечіткої самоорганізації «C-means» у системах комп'ютерного зору з метою ефективного формування простору ознак, що забезпечує результативну класифікацію

зображень за їх структурним описом у вигляді множини дескрипторів ключових точок.

Для роботи програмного застосунку були використані зображення (рис. 3.1) із набору «CelebFaces Attributes Dataset (CelebA)» – це великомасштабний набір даних атрибутів осіб з більш ніж 200 тисячами зображень знаменитостей, кожне з яких містить 40 анотацій [42]. Зображення в цьому наборі даних охоплюють великі варіації поз і фоновий безлад. CelebA має велику різноманітність, велика кількість і багаті анотації, в тому числі. Обрано візуально різні зображення для дослідження можливості їх розподілу за власними класами.



Рисунок 3.1 – Досліджувані зображення із набору CelebA

Варто зазначити, що ОТ, отримані методом ORB, який має кращу швидкість в обчисленні ОТ і розрахунку їх дескрипторів, природно підходять для більш складних зображень, такі як обрані на (рис. 3.1), що дозволяє

використовувати його в задачах, де необхідна обробка великої кількості зображень. Для виявлення потенційних ключових точок, ORB використовує модифікацію FAST-9, далі, якщо кількість ОТ перевищує значення 500, то до них застосовується детектор кутів Harris, для виключення найменш значущих. Після обчислюється кут орієнтації, обчислюючи моменти яскравості пар точок всередині квадратного вікна, що центроване відносно координат ОТ та узгоджено з її орієнтацією. Отримав кут, послідовність точок для бінарних порівнянь в дескрипторі BRIEF повертається у відповідність з цим кутом, за отриманими точками обчислюється бінарний дескриптор BRIEF.

Для проведення дослідження на обраних зображеннях були обчислені структурні описи з використанням детектора ORB, на рисунку 3.2 околиці КТ зображені у вигляді кілець.



Рисунок 3.2 – Зображення з околицями КТ

Отримані дескриптори мають вигляд бінарного вектора розміром 256. Вектори зберігаються в матриці, де кількість рядків дорівнює числу виявлених дескрипторів, в даному випадку, по 200 векторів для кожного з еталонів. Число стовпців дорівнює 256 для бітового подання дескрипторів, або 32 для байтового.

Для дослідження нечіткої кластеризації, програмно були реалізовані функції як для бітового, так і для байтового подання, такі як: об'єднання усієї множини дескрипторів ORB в єдину базу(матрицю), реалізовано розрахунок початкових вагових характеристик матричної функції приналежності, для кожного елементу бази еталонів, на базі заданих вагових характеристик виконується розрахунок центрів кластерів, виконується обчислення похибки, обчислення нових значень приналежності векторів до кожного з класів, функція визначення класу об'єктів.

Для запуску програмного застосунку, спочатку необхідно інсталиювати мову програмування Python, завантажити та інсталиювати дистрибутив Anaconda для подальшого налаштування середовища, бідклучити бібліотеку OpenCV, бібліотеку Matplotlib, Math для наукових обчислень Numpy, завантажити та інсталиювати інтегроване середовище розробки PyCharm для мови програмування Python, виконати налаштування для підключення версії Python та підключення інтерпретатора.

Для інсталиювання дистрибутиву Python, потрібно виконати наступну послідовність дій:

Крок 1. Для своєї системи завантажити дистрибутив Python з офіційної сторінки, версії не нижче 3.10.0 (рис. 3.3).

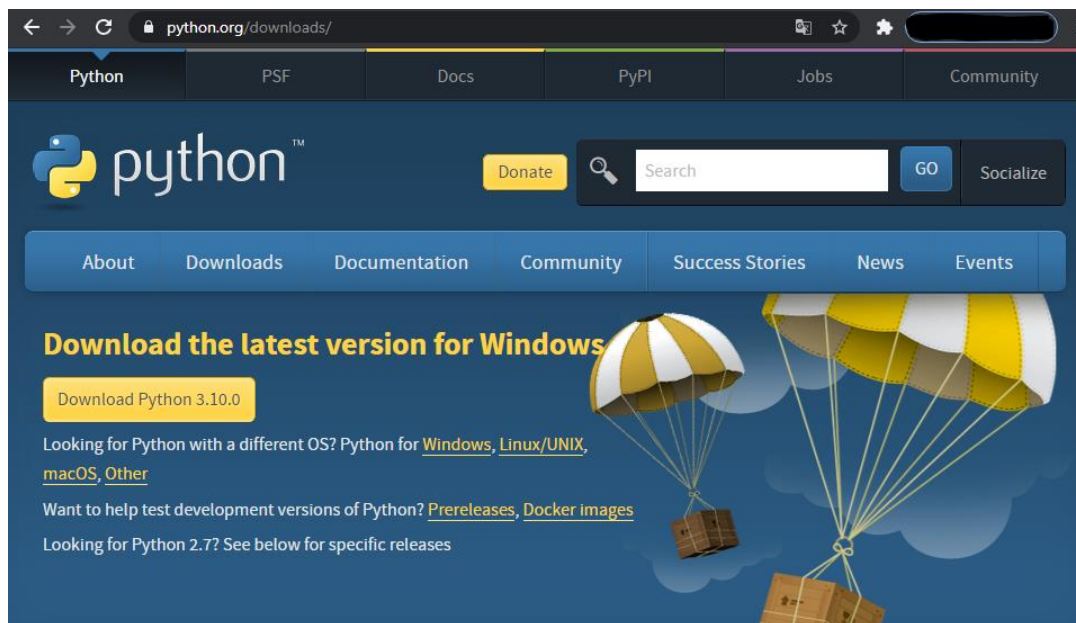


Рисунок 3.3 – Офіційна сторінка завантаження Python дистрибутиву

Крок 2. Після завантаження, запустити інсталятор, переконатись в тому, що встановлено прапорець в розділі «Install launcher for all users», та в розділі «Add Python 3.x to PATH» та обрати Install Now (рис. 3.4), дочекатись завершення.

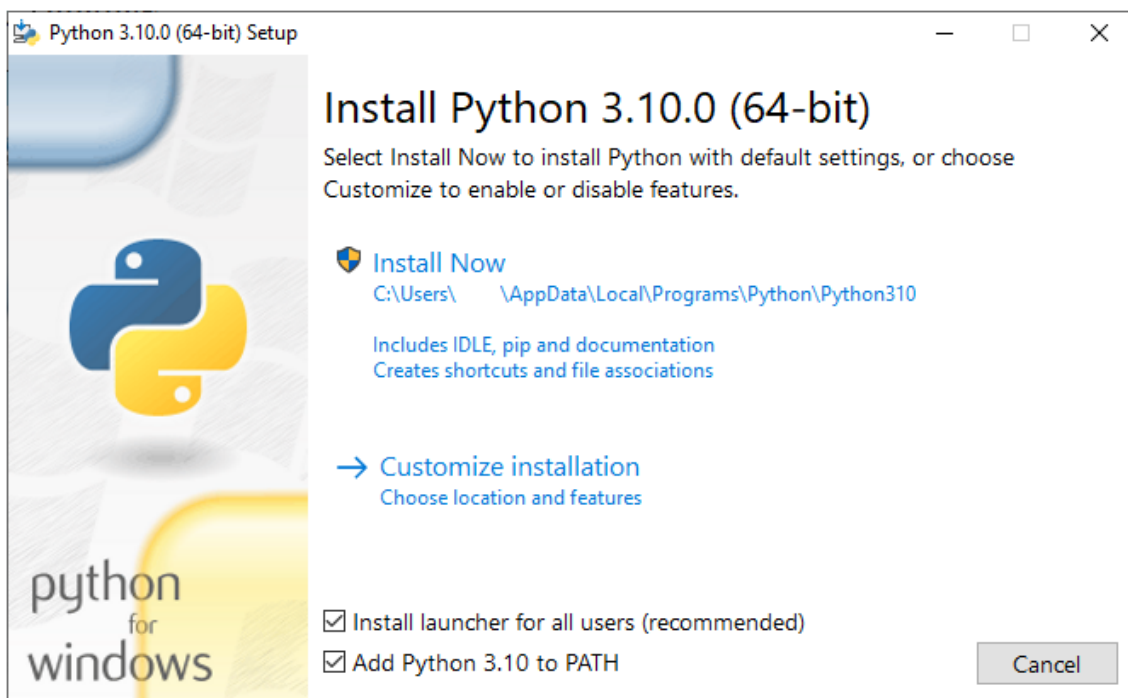


Рисунок 3.4 – Інсталятор Python

Крок 3. Переконайтесь, що Python був інстальований в систему Windows. Для цього в cmd виконати набір команди «python». Якщо установка пройшла успішно, після виконання команди, консоль Windows видасть версію мови.

Крок 4. Завантажити Anaconda під необхідну систему з офіційного сайту розробника (рис. 3.5).



Рисунок 3.5 – Завантаження Anaconda

Крок 5. При запуску, інстальатор виконає завантаження додаткових файлів. Інсталяція проходить в стандартному режимі Step-by-step. Під час процесу, є два місця, які необхідно підтвердити. Перша перевірка полягає в тому, що потрібно додавати Anaconda в змінні середовища, це включає можливість використання команд conda, jupyter, ipython та інших в cmd. Друге – встановити Python 3.x для Anaconda в якості версії Python для

системи за замовчуванням. Погоджуючись з експресними налаштуваннями, завершити інсталяцію (рис. 3.6).

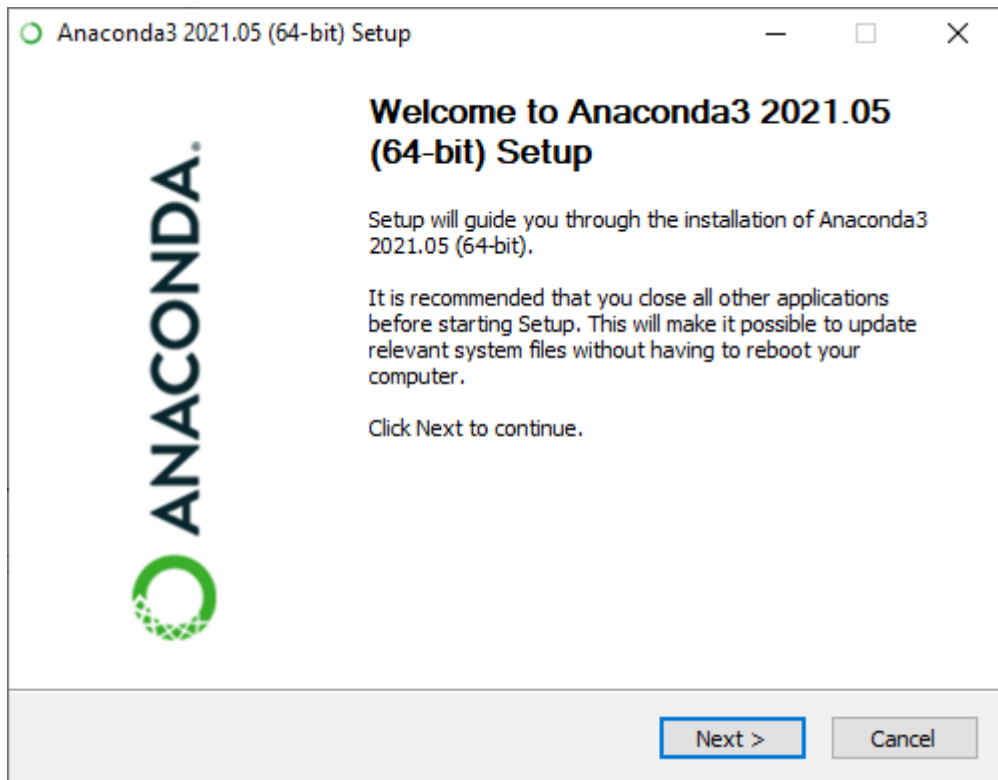


Рисунок 3.6 – Anaconda інсталятор

Крок 6. Після того, як Anaconda буде інстальована, необхідно створити та активувати нову середу для організації залежностей одним із двох запропонованих способів:

1. Через Anaconda Navigator.
2. Через командну строку шляхом вводу команд:
 - `conda create --name test python = 3.8;`
 - `source activate test.`

Створивши та активувавши середу, необхідно встановити потрібні залежності через Anaconda Navigator обравши із списку бібліотек необхідні та натиснути кнопку Apply дочекатись поки не завершиться завантаження, або через командну строку за допомогою команд:

- `conda install numpy;`
- `conda install math;`

- *conda install matplotlib;*
- *conda install cv;*
- *conda install ski-learn.*

Крок 7. Завантажити PyCharm Community (рис. 3.7).

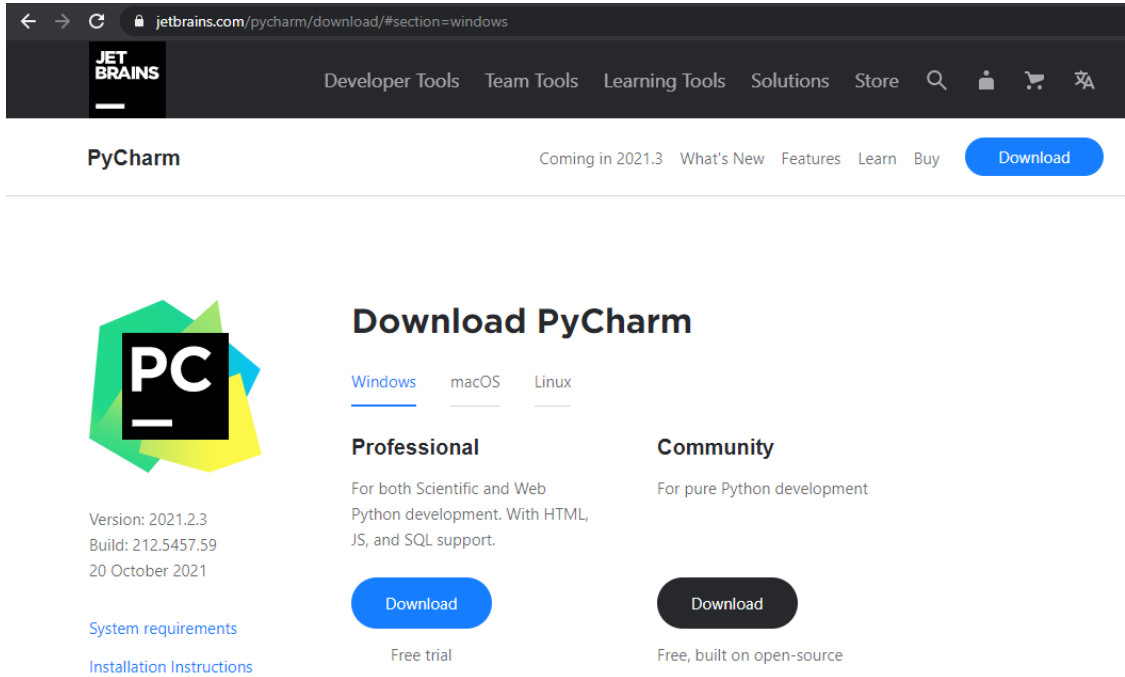


Рисунок 3.7 – Сторінка завантаження PyCharm

Крок 8. Встановити дистрибутив PyCharm, інсталяція відбувається у режимі крок-за-кроком.

Крок 9. Відкрити файл у проекті PyCharm.

Крок 10. Для запуску нового середовища з PyCharm, потрібно перейти за наступним шляхом (рис. 3.7): File > Settings > Project > Interpreter, вибрати шестерню «Додати локально в поле інтерпретатора проекту», та обрати створений інтерпретатор Python за шляхом:

C:\Users\nameusers\Anaconda3\envs\nameEnvironment\python.exe.

Крок 11. Натиснути кнопку Apply для виходу з меню зі збереженням усіх змінених налаштувань.

Крок 12. Закрити меню Settings та виконати запуск коду шляхом натискання кнопки Run, або натиснувши на клавіатурі Shift + F10, або через Python Shell написавши: *python file_name.py*.

Якщо інструкції що до налаштування програми були виконані правильно, то програма повинна запуснитись, та вивести результати розрахунків.

3.3 Аналіз результатів програмного моделювання

В розробленому застосунку у середовищі мови програмування Python PyCharm із використанням засобів бібліотеки комп'ютерного зору OpenCV були отримані результати, наведені нижче. Над обраними зображеннями не проводилося ніякої додаткової обробки. Інваріантність, яку дає метод ORB дозволяє відкинути ресурсоємну операцію конверсії розмірів для кожного зображення.

У результаті дослідження були проведені експерименти на різних вхідних даних для виявлення результативності досліджуваного методу.

За допомогою використаного детектора ORB було отримано бітове представлення з 200 дескрипторів для кожного зображення, кожен дескриптор має розмір 256 біт, тобто в результаті використання методу маємо матрицю даних 256×200 для кожного з чотирьох вхідних зображень (рис. 3.8).

```
array([[ 0 0 1 1 0 0 0 0 1 1 0 1 1 0 1 ... 1 1 1 1 0 0 1 0 0 0 1 0 1 0 1 0],
       [ 0 1 0 1 1 1 1 0 0 0 1 0 0 1 1 ... 0 0 1 1 1 0 0 1 0 1 0 1 0 1 0 1],
       [ 0 1 0 1 1 0 1 1 0 1 0 1 1 0 1 0 ... 0 0 1 0 1 1 0 1 0 1 0 1 1 1 1 1],
       ...,
       [ 1 0 0 0 1 1 0 1 1 0 1 1 0 1 0 1 ... 1 1 0 1 0 0 0 0 0 1 1 1 0 1 0 0],
       [ 1 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 ... 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0],
       [ 1 1 1 0 0 0 1 1 1 1 1 0 1 1 0 0 ... 1 1 0 1 0 0 0 0 0 1 0 0 1 1 0 0]],
```

Рисунок 3.8 – Вигляд бінарного подання

Суть першого дослідження полягає перевірі наступних умов:

- перевірка коректної роботи програмного застосунку в ідеальних умовах;
- перевірка впливу способів завдання початкових вагових коефіцієнтів на кінцевий результат.

Тест 1. Вхідні дані для перевірки: 20 векторів, 4 класи (10, 20, 30, 40) по 5 векторів (табл. 3.1), вагові коефіцієнти 0,7 для основного класу та 0,1 для інших. Результат обчислень похибки представлений у таблиці 3.1, вагових коефіцієнтів у таблиці 3.2 і векторів у таблиці 3.3.

Результат обчислень представлений у таблицях 3.2, 3.3.

Таблиця 3.1 – Похибка на кожній ітерації

Номер ітерації	E
1	119100,5917
2	13286,46305
3	11300,23269
4	11313,98159
5	11314,72533
6	11314,76458
7	11314,76665
8	11314,76676
9	11314,76677
10	11314,76677

Таблиця 3.2 – Вагові коефіцієнти u_{ij} на десятій ітерації

1	2	3	4
1	2,62E-07	6,47E-08	2,96E-08
1	2,62E-07	6,47E-08	2,96E-08
1	2,62E-07	6,47E-08	2,96E-08
1	2,62E-07	6,47E-08	2,96E-08
1	2,62E-07	6,47E-08	2,96E-08
3,01E-06	0,999993	2,92E-06	7,6E-07
3,01E-06	0,999993	2,92E-06	7,6E-07
3,01E-06	0,999993	2,92E-06	7,6E-07
3,01E-06	0,999993	2,92E-06	7,6E-07
3,01E-06	0,999993	2,92E-06	7,6E-07
5,22E-05	0,00021	0,999539	0,0002

Продовження таблиці 3.2

1	2	3	4
5,22E-05	0,00021	0,999539	0,0002
5,22E-05	0,00021	0,999539	0,0002
5,22E-05	0,00021	0,999539	0,0002
5,22E-05	0,00021	0,999539	0,0002
0,013069	0,029447	0,121057	0,836427
0,013069	0,029447	0,121057	0,836427
0,013069	0,029447	0,121057	0,836427
0,013069	0,029447	0,121057	0,836427
0,013069	0,029447	0,121057	0,836427

Таблиця 3.3 – Центри на десятій ітерації

	1	2	3	4	5	6	7	8	9	10	11	12
c1	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006
c2	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018
c3	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145
c4	40	40	40	40	40	40	40	40	40	40	40	40
	13	14	15	16	17	18	19	20	21	22	23	24
c1	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006
c2	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018
c3	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145
c4	40	40	40	40	40	40	40	40	40	40	40	40
	25	26	27	28	29	30	31	32	33	34	35	36
c1	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006
c2	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018
c3	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145
c4	40	40	40	40	40	40	40	40	40	40	40	40
	37	38	39	40	41	42	43	44	45	46	47	48
c1	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006
c2	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018
c3	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145
c4	40	40	40	40	40	40	40	40	40	40	40	40
	49	50	51	52	53	54	55	56	57	58	59	60
c1	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006	10,006
c2	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018	20,018
c3	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145	30,145
c4	40	40	40	40	40	40	40	40	40	40	40	40

Після десятої ітерації, з вхідними даними: 20 ідеальних векторів, для 4-х класів по 5 векторів, вагові коефіцієнти 0,7 для основного класу та 0,1 для

інших, всі 20 векторів отримали правильні значення класів (табл. 3.4).
Максимум u_{ij} відповідає всім класам.

Таблиця 3.4 – Відповідність між класами

Клас	Число правильних рішень
Клас 1	5
Клас 2	5
Клас 3	5
Клас 4	5

Тест 2. Вхідні дані для перевірки: 20 векторів, 4 класи (10, 20, 30, 40) по 5 векторів, вагові коефіцієнти задані випадковим чином (табл. 3.5), зберігаючи умову рівності одиниці (3.3). Результат обчислень похибки представлений у таблиці 3.6, вагових коефіцієнтів у таблиці 3.7 і векторів у таблиці 3.8.

Таблиця 3.5 – Початкові вагові коефіцієнти

1	2	3	4
0,3	0,1	0,5	0,1
0,3	0,2	0,2	0,3
0,1	0,1	0,4	0,4
0,2	0,1	0,2	0,5
0,5	0,2	0,2	0,1
0,1	0,2	0,6	0,1
0,1	0,1	0,1	0,7
0,4	0,3	0,2	0,1
0,1	0,1	0,5	0,3
0,5	0,2	0,1	0,2
0,4	0,3	0,1	0,2
0,2	0,1	0,2	0,5
0,3	0,2	0,3	0,2
0,1	0,5	0,1	0,3
0,6	0,1	0,1	0,1
0,2	0,2	0,3	0,3
0,1	0,5	0,2	0,2
0,2	0,4	0,3	0,1
0,5	0,1	0,1	0,3
0,1	0,1	0,1	0,6

Продовження таблиці 3.8

	13	14	15	16	17	18	19	20	21	22	23	24
c1	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998
c2	40	40	40	40	40	40	40	40	40	40	40	40
c3	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003
c4	20	20	20	20	20	20	20	20	20	20	20	20
	25	26	27	28	29	30	31	32	33	34	35	36
c1	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998
c2	40	40	40	40	40	40	40	40	40	40	40	40
c3	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003
c4	20	20	20	20	20	20	20	20	20	20	20	20
	37	38	39	40	41	42	43	44	45	46	47	48
c1	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998
c2	40	40	40	40	40	40	40	40	40	40	40	40
c3	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003
c4	20	20	20	20	20	20	20	20	20	20	20	20
	49	50	51	52	53	54	55	56	57	58	59	60
c1	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998	29,998
c2	40	40	40	40	40	40	40	40	40	40	40	40
c3	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003	10,003
c4	20	20	20	20	20	20	20	20	20	20	20	20

В результаті тестування моделі в ідеальних умовах впливають такі висновки:

- спосіб подачі вагових коефіцієнтів мінімально впливає на отримані результати, хоч число ітерацій відрізняється, головне – дотримуватись умови, щоб сума коефіцієнтів для кожного з дескрипторів дорівнювала одиниці, та щоб всі початкові коефіцієнти не були рівними між собою;
- модель запрограмована вірно, це обумовлено тим, що в результаті досліджень були отримані передбачувані дані.

Тест 3. Вхідні дані для перевірки: 20 бітових векторів, 4 класи по 5 векторів (16 біт), вагові коефіцієнти 1,0 для основного та 0 для інших. Результат обчислень похибки представлений у таблиці 3.9, вагових коефіцієнтів у таблиці 3.10 і векторів у таблиці 3.11.

Таблиця 3.9 – Похибка на кожній ітерації

Номер ітерації	Е
1	69,61
2	73,88
3	73,73
4	73,69
5	73,68

Таблиця 3.10 – Вагові коефіцієнти на п'ятій ітерації

1	2	3	4
0,220236	0,237909	0,268789	0,258285
0,262471	0,249442	0,250546	0,237542
0,250665	0,235499	0,271556	0,24228
0,264943	0,267388	0,232742	0,234928
0,26712	0,275034	0,222502	0,235344
0,242446	0,255668	0,228581	0,273305
0,271809	0,279285	0,217591	0,231314
0,241012	0,237555	0,28596	0,235473
0,252537	0,253902	0,261245	0,232316
0,264057	0,27581	0,20766	0,252472
0,234158	0,230782	0,276056	0,259004
0,234238	0,229792	0,302653	0,233317
0,254674	0,256044	0,25929	0,229992
0,235002	0,224089	0,299605	0,241303
0,263088	0,25183	0,243068	0,242014
0,250219	0,253497	0,225105	0,271179
0,251022	0,26558	0,238736	0,244662
0,257625	0,259914	0,223768	0,258692
0,232826	0,230179	0,255646	0,281349
0,230342	0,231648	0,247008	0,291002

Таблиця 3.11 – Центри на п'ятій ітерації

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
c1	0,326	0,434	0,605	0,332	0,514	0,334	0,786	0,503	0,67	0,545	0,653	0,638	0,606	0,396	0,535	0,669
c2	0,339	0,435	0,626	0,307	0,506	0,343	0,789	0,472	0,7	0,55	0,689	0,638	0,603	0,408	0,491	0,679
c3	0,301	0,44	0,681	0,427	0,428	0,236	0,732	0,567	0,6	0,688	0,602	0,547	0,69	0,336	0,614	0,733
c4	0,442	0,494	0,692	0,331	0,56	0,289	0,696	0,461	0,638	0,617	0,659	0,58	0,701	0,462	0,568	0,722

Після п'яти ітерацій у бітовому поданні, з вхідними даними: 20 бітових векторів, для 4-х класів по 5 векторів; вагові коефіцієнти 1,0 для основного класу та 0 для інших.

В результаті 10 отримали правильні класи (табл. 3.12). Максимум ціл відповідає своєму 3-му класу.

Таблиця 3.12 – Відповідність між класами

Клас	Кількість відносин
Клас 1	1
Клас 2	2
Клас 3	4
Клас 4	3

Тест 4. Вхідні дані для перевірки: 20 бітових векторів, 4 класи по 5 векторів (64 біт), вагові коефіцієнти 1,0 для основного та 0 для інших. Результат обчислень похибки представлений у таблиці 3.13 (похибка не зменшується), вагових коефіцієнтів у таблиці 3.14 і векторів у таблиці 3.15.

Таблиця 3.13 – Похибка на кожній ітерації

Номер ітерації	E
1	228,8
2	289,91
3	288,94
4	288,77
5	288,79

Таблиця 3.14 – Вагові коефіцієнти на п'ятій ітерації

1	2	3	4
0,234672	0,247023	0,245113	0,265489
0,253714	0,250141	0,255462	0,240684
0,254766	0,25148	0,248807	0,244947
0,256274	0,253365	0,256128	0,234233
0,260227	0,255519	0,242331	0,241922
0,247169	0,250031	0,238789	0,264012
0,254552	0,254125	0,240428	0,250895
0,246074	0,249023	0,268939	0,235964
0,249005	0,250017	0,277207	0,223771

Продовження таблиці 3.14

1	2	3	4
0,260355	0,255174	0,232997	0,251474
0,251471	0,251553	0,25629	0,240686
0,246701	0,248375	0,278732	0,226193
0,251709	0,252051	0,259406	0,236835
0,245687	0,247754	0,271564	0,234995
0,255326	0,252496	0,251466	0,240711
0,245262	0,247014	0,236377	0,271347
0,250778	0,252812	0,253506	0,242904
0,25281	0,25054	0,233359	0,263292
0,244652	0,246958	0,238875	0,269515
0,242944	0,245016	0,236298	0,275743

Таблиця 3.15 –Центри на п'ятій ітерації

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
c1	0,35	0,44	0,64	0,34	0,51	0,32	0,77	0,51	0,66	0,58	0,65	0,62	0,63	0,4	0,55	0,68
c2	0,35	0,45	0,65	0,34	0,5	0,31	0,77	0,5	0,66	0,6	0,66	0,61	0,64	0,4	0,55	0,7
c3	0,31	0,45	0,67	0,38	0,45	0,28	0,77	0,55	0,65	0,64	0,63	0,59	0,65	0,35	0,58	0,72
c4	0,42	0,48	0,67	0,35	0,56	0,3	0,71	0,47	0,65	0,6	0,68	0,6	0,69	0,46	0,55	0,72
	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
c1	0,4	0,65	0,77	0,37	0,29	0,19	0,7	0,3	0,78	0,44	0,29	0,57	0,51	0,49	0,3	0,41
c2	0,4	0,65	0,76	0,37	0,3	0,2	0,71	0,3	0,76	0,45	0,3	0,57	0,51	0,5	0,3	0,41
c3	0,37	0,69	0,76	0,32	0,31	0,18	0,74	0,31	0,77	0,48	0,28	0,52	0,54	0,52	0,3	0,44
c4	0,45	0,63	0,73	0,36	0,32	0,26	0,67	0,31	0,71	0,46	0,36	0,57	0,47	0,51	0,31	0,36
	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
c1	0,56	0,82	0,56	0,6	0,49	0,76	0,67	0,66	0,77	0,51	0,51	0,73	0,39	0,51	0,54	0,65
c2	0,56	0,82	0,56	0,61	0,5	0,76	0,66	0,66	0,77	0,51	0,5	0,72	0,41	0,51	0,55	0,66
c3	0,5	0,84	0,55	0,64	0,46	0,79	0,69	0,6	0,77	0,44	0,45	0,69	0,4	0,46	0,59	0,66
c4	0,61	0,75	0,56	0,56	0,57	0,71	0,59	0,7	0,72	0,56	0,56	0,68	0,41	0,54	0,54	0,64
	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
c1	0,72	0,48	0,63	0,72	0,82	0,39	0,6	0,34	0,64	0,82	0,57	0,55	0,46	0,54	0,71	0,4
c2	0,71	0,49	0,62	0,71	0,81	0,4	0,61	0,35	0,65	0,81	0,59	0,55	0,46	0,55	0,71	0,4
c3	0,73	0,51	0,63	0,69	0,83	0,37	0,64	0,33	0,65	0,84	0,63	0,58	0,45	0,55	0,65	0,36
c4	0,66	0,54	0,55	0,7	0,75	0,47	0,57	0,4	0,68	0,74	0,62	0,54	0,45	0,58	0,75	0,45

Після п'яти ітерацій у бітовому поданні, з цими вхідними даними 11 векторів отримали правильні класи (табл. 3.16). Максимум u_{ij} відповідає своєму класу.

Таблиця 3.16 – Відповідність між класами

Клас	Кількість відносин
Клас 1	3
Клас 2	0
Клас 3	4
Клас 4	4

Тест 5. Вхідні дані для перевірки: 20 байтових векторів, 4 класи по 5 векторів (32 байт), вагові коефіцієнти 1,0 для основного та 0 для інших. Результат обчислень похибки представлений у таблиці 3.17 (похибка не зменшується), вагових коефіцієнтів у таблиці 3.18, вектора у таблиці 3.19 і відстань між векторами у таблиці 3.20

Таблиця 3.17 – Похибка на кожній ітерації

Номер ітерації	E
1	2369534,0
2	864450,27
3	844405,65
4	839116,63
5	837007,5
6	836000,22
7	835437,48
8	835071,74
9	834801,01
10	834579,5

Таблиця 3.18 – Вагові коефіцієнти на десятій ітерації

1	2	3	4
0.243039	0.242029	0.265032	0.249899
0.269367	0.268048	0.201219	0.261364
0.273406	0.271910	0.188346	0.266336
0.269250	0.271304	0.197460	0.261984
0.261892	0.263497	0.215901	0.258708
0.276626	0.277664	0.171578	0.274130
0.269121	0.271204	0.191933	0.267740
0.241824	0.240806	0.273906	0.243461
0.282625	0.282453	0.165497	0.269423
0.276202	0.277941	0.179049	0.266805
0.219469	0.217981	0.337803	0.224745

Продовження таблиці 3.18

1	2	3	4
0.247955	0.248196	0.246203	0.257644
0.190729	0.190534	0.424113	0.194621
0.178041	0.177681	0.460902	0.183375
0.188483	0.188099	0.429268	0.194148
0.266746	0.265486	0.199689	0.268076
0.285395	0.285278	0.155799	0.273526
0.259859	0.261525	0.209720	0.268895
0.253785	0.253736	0.227449	0.265028
0.262149	0.261473	0.209392	0.266984

Таблиця 3.19 – Центри на десятій ітерації

	1	2	3	4	5	6	7	8	9	10	11
c1	124,08	161,946	125,909	133,248	173,337	157,981	147,696	153,665	128,449	143,269	158,308
c2	123,804	162,359	125,401	133,426	173,767	158,717	147,947	153,879	128,565	142,91	158,02
c3	97,914	127,499	85,606	107,65	134,725	101,765	115,536	106,204	118,529	132,42	136,101
c4	125,948	161,367	124,902	131,18	172,616	157,174	143,987	152,606	130,426	142,017	153,463
	12	13	14	15	16	17	18	19	20	21	22
c1	126,722	133,552	153,727	119,661	115,929	166,962	151,058	182,608	124,297	163,373	140,852
c2	125,587	133,602	153,999	119,75	116,1	167,169	150,026	182,784	123,787	163,061	140,607
c3	99,43	166,43	89,793	105,584	92,285	153,25	130,913	132,638	114,441	161,789	170,07
c4	122,672	136,93	151,787	116,221	114,315	165,515	150,65	181,729	121,382	163,954	141,935
	23	24	25	26	27	28	29	30	31	32	
c1	149,517	178,544	130,21	134,507	120,833	149,093	116,872	137,349	138,495	143,644	
c2	149,493	179,174	130,109	134,128	120,246	149,593	117,271	136,552	138,894	144,094	
c3	134,835	105,665	151,764	159,657	83,248	102,699	100,168	119,154	107,603	138,566	
c4	149,518	175,627	135,508	138,232	119,334	145,063	117,202	136,889	140,493	141,526	

Таблиця 3.20 – Міра подібності між векторами

вектора	відстань
$c_i[0], c_i[1]$	2,3637
$c_i[0], c_i[2]$	187,8764
$c_i[0], c_i[3]$	12,9687
$c_i[1], c_i[2]$	188,5989
$c_i[1], c_i[3]$	13,6096
$c_i[2], c_i[3]$	180,1848

Після десяти ітерацій у байтовому поданні з цими вхідними даними 13 векторів отримали правильні класи (табл. 3.21).

Таблиця 3.21 – Відповідність між класами

Клас	Кількість відносин
Клас 1	2
Клас 2	3
Клас 3	4
Клас 4	4

ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований метод структурної класифікації зображень з використанням засобів нечіткої кластеризації даних.

Задля отримання структурного опису еталонів обрано метод ORB завдяки тому, що метод являється менш вимогливим до обчислювальних ресурсів, дозволяє використовувати дата-сети великого розміру, забезпечує високу швидкість детектування за рахунок спрощення процесу оброблення та побудови і використання дескрипторів бінарного типу.

ORB дозволяє вибрати довільний спосіб побудови детектора з будь-яким способом визначення дескриптора та навпаки, оптимізуючи бажану продуктивність вирішуваних задач.

Проаналізовано основні особливості кластерного аналізу даних, розглянуто основні алгоритми нечіткої кластеризації, описані переваги використаних засобів для розробки застосунку.

Дослідження показало, що ефективність класифікації візуальних об'єктів з використанням засобів нечіткої кластеризації даних суттєво залежить безпосередньо від вхідних даних і від значень початкової системи вагових коефіцієнтів. В результаті експериментів досліджуваний метод не зміг продемонструвати ефективну кластеризацію-класифікацію на вхідних векторах-дескрипторах зображень, які розрізняються людським зором. У той же час при поданні на вхід ідеальних даних метод успішно виконав кластеризацію за 2-3 ітераційних проходи.

Результати роботи було апробовано у вигляді 2 тез доповідей під час 25го Міжнародного молодіжного форуму «Радіоелектроніка і молодь у XXI столітті» [43], IV Міжнародної наукової конференції «Science, theory and practice» [44].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Путятін, Є. П., Гороховатський, В. О., & Матат, О. О. (2006). Методи та алгоритми комп'ютерного зору: навч. посіб. Харків: ТОВ «Компанія СМІТ.
2. Гороховатский, В. А. (2014). Структурный анализ и интеллектуальная обработка данных в компьютерном зрении.
3. Lyashenko, V., Kobylin, O., & Ahmad, M. A. (2014). General methodology for implementation of image normalization procedure using its wavelet transform. *International Journal of Science and Research (IJSR)*, 3(11), 2870-2877.
4. Gorokhovatskyi, V., Gadetska, S., & Ponomarenko, R. (2019, May). Recognition of visual objects based on statistical distributions for blocks of structural description of image. In *International Scientific Conference "Intellectual Systems of Decision Making and Problem of Computational Intelligence"* (pp. 501-512). Springer, Cham.
5. Gorokhovatskyi, V., Gadetska, S., & Stiahlyk, N. (2020). Image structural classification technologies based on statistical analysis of descriptions in the form of bit descriptor set.
6. Gorokhovatskiy, V. A. (2011). Compression of descriptions in the structural image recognition. *Telecommunications and Radio Engineering*, 70(15).
7. Gorokhovatskiy, V. A., & Putyatin, Y. P. (2009). Image Likelihood Measures on the Basis of the Set of Conformities. *Telecommunications and Radio Engineering*, 68(9).
8. Gorokhovatsky, V. A., Gadetska, S. V., & Stiahlyk, N. I. (2019). Вивчення статистичних властивостей моделі блочного подання для множини дескрипторів ключових точок зображень. *Радіоелектроніка, інформатика, управління*, (2), 100-107.

9. Кучеренко, Е. И. (2005). Интеллектуальные технологии в задачах принятия решений технологических комплексов на основе нечеткой интервальной логики. Восточно-Европейский журнал передовых технологий, (2), 92-96. [6-13]
10. Oliinyk, A., Subbotin, S., Lovkin, V., Blagodariov, O., & Zaiko, T. (2017). The system of criteria for feature informativeness estimation in pattern recognition. *Радіоелектроніка, інформатика, управління*, (4 (43)).
11. Gorokhovatskyi, O., Gorokhovatskyi, V., & Peredrii, O. (2018). Analysis of Application of Cluster Descriptions in Space of Characteristic Image Features. *Data*, 3(4), 52.
12. Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.
13. Гороховатський, В. О., & Пономаренко, Р. П. (2020). Класифікація зображень на підставі формування незалежної системи кластерів у складі структурних описів бази еталонів.
14. Гороховатський, В. О., Гадецька, С. В., & Пономаренко, Р. П. (2019). Статистичні розподіли та ланцюжкове подання даних при визначенні релевантності структурних описів візуальних об'єктів.
15. Gorokhovatskyi, V.O., Tvoroshenko, I.S., and Vlasenko N.V. (2020) Using Fuzzy Clustering in Structural Methods of Image Classification. *Telecommunications and Radio Engineering*, 79 (9), pp. 781-791.
16. Гороховатский В.А. Распознавание изображений в условиях неполной информации/ В.А. Гороховатский.– Х.: ХНУРЭ, 2003. –112с.
17. M. Ayaz Ahmad, Volodymyr Gorokhovatskyi, Iryna Tvoroshenko, Nataliia Vlasenko, Syed Khalid 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

18. Гороховатський, В. О., & Гадецька, С. В. (2018). Застосування статистичних мір релевантності для векторних структурних описів об'єктів у задачі класифікації зображень.

19. 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.

20. Gorbachuk, V., Ermolieva, T., & Obersteiner, M. Robustness, iterative stochastic quasigradient procedures, and adaptive (artificial intelligence) learning for cat risk management.

21. Botvin, M., & Gertsy, A. (2020). ПОРІВНЯННЯ АЛГОРИТМІВ КЛАСТЕРНОГО АНАЛІЗУ ПРИ РОЗПІЗНАВАННІ ОБ'ЄКТІВ. *Транспортні системи і технології*, (36), 112-120.

22. Gorokhovatsky, V. (2014). *Structural analysis and intellectual data processing in computer vision*. SMIT: Kharkiv, Ukraine.

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

24. Гороховатський, В. О., Власенко, Н. В., & Рибалка, М. О. (2021). Застосування засобів хешування даних для прискорення класифікаційних рішень у структурних методах розпізнавання зображень.

25. Gorokhovatsky, V., Stiahlyk, N., & Tsarevska, V. (2021). КОМБІНАЦІЙНИЙ МЕТОД ПРИСКОРЕНОГО МЕТРИЧНОГО ПОШУКУ ДАНИХ У ЗАДАЧАХ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ. *Advanced Information Systems*, 5(3), 5-12.

26. Кобилін, О. А., & Творошенко, І. С. (2021). *Методи цифрової обробки зображень: навч. посібник*. Харків: ХНУРЕ.

27. Гороховатський, В. О., & Творошенко, І. С. (2021). *Методи інтелектуального аналізу та оброблення даних: навч. посібник*

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

29. Rahman, M., Khan, M. J., Asghar, M. A., Amin, Y., Badnava, S., & Mirjavadi, S. S. (2019). Image local features description through polynomial approximation. *IEEE Access*, 7, 183692-183705.

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

31. 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.

32. Гороховатський, В. А., & Путятин, Е. П. (2008). Структурное распознавание изображений на основе моделей голосования признаков характерных точек. Реєстрація, зберігання і обробка даних.

33. Гороховатський, В. О., Пупченко, Д. В., & Стяглик, Н. І. (2020). Дослідження трансформацій простору даних при навчанні мережі кохонена у методах структурної класифікації зображень.

34. Гороховатський, В. О., & Солодченко, К. Г. (2018). Застосування апарату аналізу та оброблення бітових даних у методах класифікації зображень за множиною ключових точок. Системи управління, навігації та зв'язку, (2), 63-67.

35. Осовский, С. (2004). Нейронные сети для обработки информации. Финансы и статистика.

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

37. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., & Al-Dhaifallah, M. (2021). Methods of Classification of Images on the Basis of the Values of Statistical Distributions for the Composition of Structural Description Components. *IEEE Access*, 9, 92964-92973.

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

39. Гороховатський, В. О., & Гадецька, С. В. (2020). Теоретичні засади статистичного аналізу даних у структурних методах класифікації зображень.

40. Івохін, Є. В., & Апанасенко, Д. В. (2013). Про застосування методів кластеризації нечітких даних спеціального вигляду. *Вісник Київського національного університету імені Тараса Шевченка. Серія: Фізико-математичні науки*, (1), 167-171.

41. Kobylin, O. A., Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). The application of non-parametric statistics methods in image classifiers based on structural description components. *Telecommunications and Radio Engineering*, 79(10).

42. Liu Z. Large-scale CelebFaces Attributes (CelebA) Dataset. URL: <https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html> (дата звернення: 22.10.2021).

43. Єрмоєнко, В. (2021). Розпізнавання зображень на основі статистичних характеристик для компонентів структурного опису. *Міжнародний молодіжний форум «Радіоелектроніка і молодь у XXI столітті»*. Зб. матеріалів форуму. Т. 7, 10. Харків: ХНУРЕ. 2021. (20-22).

44. Гороховатський, В., Єрмоєнко, В. (2021). Класифікація зображень з використанням засобів нечіткої кластеризації даних. *The IV International Science Conference «Science, theory and practice»*, October 12 – 15, 2021, Tokyo, Japan. 477. (427).