

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)

«____» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Клінову В'ячеславу Олексійовичу
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження моделі комітету нейронних мереж у задачі класифікації зображень

затверджена наказом по університету від 3 листопада 2023 року № 1280Ст

2. Термін подання студентом роботи до екзаменаційної комісії 25 _____ 12 _____ 2023 р.3. Вихідні дані до роботи математичні моделі класифікації зображень, теоретичні відомості про ансамблеве навчання, теоретичні відомості про застосування нейронних мереж у задачі класифікації зображень, тестові зображення, програмне забезпечення для розробки.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз застосування нейронних мереж у задачі класифікації зображень.2. Математична модель комітету нейронних мереж векторного квантування.3. Застосування структурного опису на базі дескрипторів ключових точок.4. Програмне моделювання розглянутих методів класифікації та їх тестування.5. Аналіз результатів дослідження методів класифікації зображень.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) UML-діаграма програмної моделі, постановка задачі, аналіз результатів роботи, тестові зображення.

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

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	03.11.2023	
2	Аналіз завдання, підбір літератури	03.11.23-05.11.23	
3	Аналіз літератури з досліджуваної проблеми	06.11.23-11.11.23	
4	Аналіз методів класифікації	12.11.23-14.11.23	
5	Розробка та порівняння методів	15.11.23-01.12.23	
6	Програмна реалізація	01.12.23-15.12.23	
7	Оформлення пояснювальної записки	16.12.23-25.12.23	
8	Перевірка на плагіат	26.12.2023	
9	Рецензування	27.12.2023	
10	Підготовка презентації та доповіді	28.12.2023	
11	Занесення роботи в електронний архів	03.01.2024	
12	Попередній захист кваліфікаційної роботи	04.01.2024	

Дата видачі завдання 3 листопада 2023 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

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

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 62 с., 12 табл., 17 рис., 1 дод., 53 джерела.

КОМП'ЮТЕРНИЙ ЗІР, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, ШТУЧНА НЕЙРОННА МЕРЕЖА, МЕРЕЖА КОХОНЕНА, ВЕКТОРНЕ КВАНТУВАННЯ, АНСАМБЛЕВЕ НАВЧАННЯ, КОМІТЕТ НЕЙРОННИХ МЕРЕЖ, ДЕТЕКТОР ORB, ДЕСКРИПТОР.

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

Метою дослідження є впровадження моделі комітету мереж векторного квантування Кохонена у процес класифікації, що базується на опису у формі множини бінарних дескрипторів ключових точок.

Класифікація зображення здійснюється на підставі процедури голосування дескрипторів вхідного зображення та на множині опису еталонів.

У результаті дослідження здійснена програмна реалізація системи для класифікації зображень за дескрипторами ключових точок.

COMPUTER VISION, IMAGE CLASSIFICATION, ARTIFICIAL NEURAL NETWORK, KOHONEN NETWORK, VECTOR QUANTIZATION, ENSEMBLE LEARNING, NEURAL NETWORK COMMITTEE, ORB DETECTOR, DESCRIPTOR.

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

The purpose of the research is to introduce the neural network committee model into the classification process, which is based on the description in the form of a set of binary descriptors of keypoints.

The classification of the image is carried out on the basis of the voting procedure of the descriptors of the input image and on the set of description etalons.

As a result of implemented software implementation of the system for image classification by descriptors of keypoints.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ.....	7
1 Аналіз застосувань нейронних мереж для класифікації зображень	9
1.1 Згорткові нейронні мережі.....	9
1.2 Моделі на основі ансамблю нейронних мереж.....	13
1.3 Класифікація на базі структурного опису зображення.....	21
1.4 Постановка задачі дослідження.....	24
2 Математична модель комітету мереж Кохонена	25
2.1 Вибір архітектури базового класифікатора ансамблю	25
2.2 Математичний апарат навчання мережі Кохонена	28
2.3 Конструкція ансамблю нейронних мереж Кохонена	31
2.4 Оцінювання ефективності класифікації зображень	34
3 Результати дослідження комп'ютерної моделі комітету мереж Кохонена	36
3.1 Обґрунтування вибору середовища програмної реалізації	36
3.2 Особливості програмної реалізації	37
3.3 Аналіз результатів тестування.....	44
Висновки	55
Перелік джерел посилання	56
Додаток А Таблиці точності класифікації тестових зображень для моделей комітету мереж Кохонена.....	62

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

ШНМ (НМ) – штучна нейронна мережа

ЗШНМ (ЗНМ) – згорткова штучна нейронна мережа

КТ – ключова точка

ORB – Oriented FAST and Rotated BRIEF

ВК – векторне квантування

ВСТУП

Розвиток комп'ютерного зору в сучасному світі є критично важливим у забезпеченні автоматизації процесів, підвищенні продуктивності та точності прийняття рішень. У сучасних умовах розвитку штучного інтелекту та росту графічного контенту, оптимізація існуючих методів рішень задач комп'ютерного зору, зокрема класифікації зображень, стала ключовою складовою для прогресу в різноманітних галузях, включаючи торгівлю, науку, медицину, технології безпеки та інші. Розвиток комп'ютерного зору стимулює інновації, сприяючи появі нових технологій та способів впровадження передових рішень у різних галузях людської діяльності.

Актуальність застосування штучних нейронних мереж, як універсальних апроксиматорів, в задачі класифікації зображень полягає у пошуку нових підходів для поліпшення точності та швидкодії процесу аналізу візуальної інформації.

Хоча сучасні архітектури ШНМ відрізняються високою точністю, вони є досить громіздкими та вимагають великих обсягів обчислювальних потужностей. Натомість застосування об'єднаних в ансамбль простих моделей може становити конкуренцію глибоким згортковим мережам.

Однією з ефективних моделей, застосованих в задачі розпізнавання образів, є мережа Кохонена. Даний алгоритм дозволяє вивчити внутрішні закономірності в даних, застосовуючи засоби самоорганізації. Ця мережа здатна ефективно структурувати вхідні дані, створюючи топологію, що відображає внутрішню структуру вхідних даних без потреби в учителі, що робить її ефективними для кластеризації та розпізнавання. Такий підхід дозволяє мережі Кохонена ефективно використовувати при аналізі зображень та розпізнаванні патернів без потреби у великих обсягах навчальних даних.

Задля ефективного масштабування кількості запам'ятованих образів мережі Кохонена можна об'єднати у моделі комітету. Основною перевагою такої моделі є її здатність працювати зі складними даними, враховуючи

різноманітні характеристики зображень і виявляти їхні закономірності у вигляді структурного опису, працюючи незалежно один від одного. Це дозволяє збільшити точність класифікації та підвищити рівень надійності у розпізнаванні відмінностей класів зображень.

Популярним методом задля забезпечення більш стислого подання даних для навчання моделі є застосування структурно опису зображення. Прикладом структурного опису можна вважати кути, краї, текстури, форми та ключові точки. Це дозволяє моделям класифікації зосередитися на суттєвих аспектах зображення, роблячи їх менш чутливими до шуму та деталей, що можуть бути менш важливими для процесу розпізнавання.

У рамках даної кваліфікаційної роботи проведено дослідження методів класифікації з використанням структурного опису зображення на основі дескрипторів ключових точок бази еталонів разом із моделлю комітету мереж векторного квантування Кохонена. Оцінено і проаналізовано отримані результати моделей різної навчальної конфігурації, а також проведено порівняння з іншими методами методом класифікації на базі одиничної моделі.

Актуальність дослідження полягає у розвитку методів класифікації зображень з використанням засобів самонавчання у формі ансамблевої моделі за множиною дескрипторів ключових точок та пошуку оптимального методу класифікації з урахуванням таких критеріїв, як точність та швидкодія класифікації.

1 АНАЛІЗ ЗАСТОСУВАНЬ НЕЙРОННИХ МЕРЕЖ ДЛЯ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

1.1 Згорткові нейронні мережі

У другій половині ХХ століття багато вчених і дослідників працювали над розумінням структури та функцій нейронів, а також вивченням обробки інформації в мозку. Ці дослідження створили основу для подальших розробок у галузі штучного інтелекту, зокрема для розвитку згорткових нейронних мереж (Convolutional Neural Networks) [1-4].

Першою важливою умовою були дослідження візуальної обробки мозку. Роботи Вільяма Джеймса та Губеля-Візеля в 1960-1970 роках розкрили принципи обробки візуальної інформації у мозку мавп. Вони вивчали відповіді нейронів на різні візуальні стимули та виявили структуру рецептивних полів нейронів. Клітини в цьому контексті функціонують як малий регіональний фільтр для вхідних даних, реагуючи на наявність певних базових образів у своєму рецептивному полі, таких як контури і межі. Ці дослідження надали важливі вказівки щодо організації обробки візуальної інформації в мозку, що стало основою для згорткових шарів у нейронних мережах.

Подальші дослідження анатомії та фізіології кори головного мозку, проведені С. Куісом та В. Розенблатом, виявили важливі принципи організації нейронних з'єднань, такі як ієрархічність та рекурсивність, що стали основою для розробки згорткових нейронних мереж з багат шаровою архітектурою. Роботи Н. Віннера та К. Шеннона з обробки сигналів і інформації сприяли розвитку теоретичних основ і математичних методів для згорткових нейронних мереж.

Таким чином згорткові нейронні мережі представляють собою варіацію багат шарової архітектури нейронних мереж, які включають в себе згортку вхідних даних (зазвичай двовимірних, такі як зображення) із матрицями

фільтрів для виявлення певних примітивів у даних. У результаті отримуються карти ознак, які дозволяють знаходити ознаки певних об'єктів на зображенні незалежно від їх розташування у полі зору. Для зменшення надмірності в даних застосовуються шари субдискретизації (пулінгу), які агрегують значення карт ознак в межах вікна заданого розміру із певним кроком. Таким чином більшість інформації про локальні ознаки зберігається при значному зниженні об'ємів даних для обробки.

Згорткові мережі набули популярності на початку XXI століття через зростання доступності великих обсягів даних, покращення обчислювальних потужностей графічних процесорів (GPU) та появу спеціалізованих архітектур для швидкого навчання глибоких нейронних мереж. Розглянемо деякі з них.

Початок нової ери визначила поява мережі AlexNet (рис. 1.1) [5] в рамках міжнародного змагання з розпізнавання зображень ImageNet, де вона зайняла перше місце, значно зменшивши кількість помилок у класифікації (до рівня близько 16%). Через це до згорткових нейромереж було привернуто багато уваги.

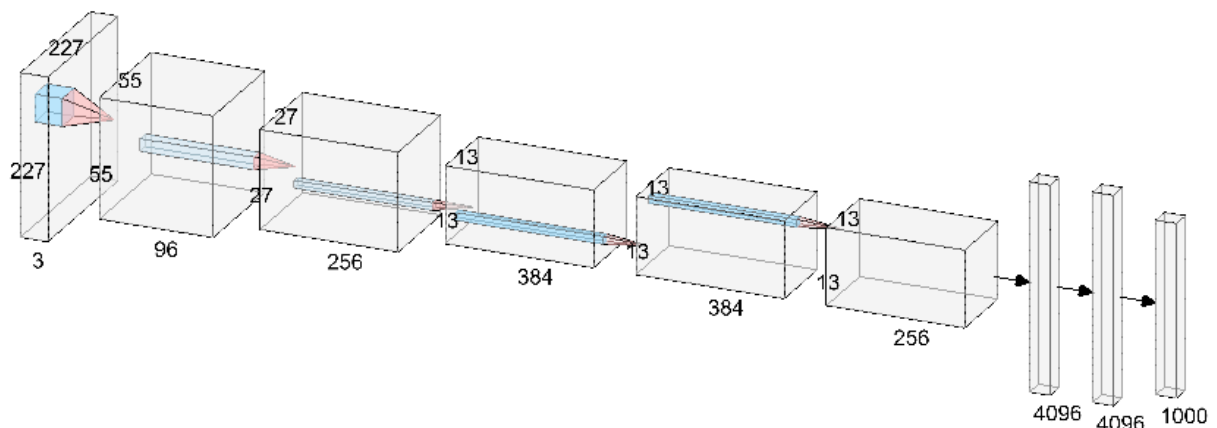


Рисунок 1.1 – Схема архітектури мережі AlexNet

Особливістю AlexNet є те, що згорткові шари розташовані один за одним. До цього, зазвичай, використовували згорткові шари, за якими одразу

йшли агрегуючі шари (max-пулінг або серединний пулінг). Мережа складалась з 60 мільйонів параметрів, 500000 нейронів, п'яти згорткових шарів, деякі з яких передували шарам з max-пулінгом, двох повнозв'язних шарів та фінального 1000-вимірний softmax шару. Ця архітектура ефективно реалізована на GPU з застосуванням регуляризації для уникнення перенавчання. Також AlexNet вперше використовувала функцію активації ReLU (Rectified Linear Unit) в своїх шарах. Це дозволило моделі більше ефективно вивчати та розпізнавати нелінійні залежності у даних.

Наступною визначною архітектурою серед ЗНМ стала VGGNet [6] – мережа, основним здобутком якої є доведення емпіричним шляхом того, що глибина мережі критично важливий показник хороших результатів. Значного покращення можна досягнути, використовуючи 16-19 згорткових або повнозв'язних шарів з дуже однорідною архітектурою, яка виконує лише 3×3 згортки та 2×2 агрегацію. Недоліками є те, що вона повільна та використовує багато пам'яті та параметрів. VGGNet досі може використовуватися в якості мережі для формування карти ознак (feature extraction) на етапі попередньої обробки.

GoogLeNet (Inception-v1) [7] – вперше представила архітектуру Inception, що значно зменшила кількість параметрів в мережі до 4 мільйонів. Мережа складається з дев'яти блоків або Inception модулів, схожих за своєю архітектурою. Кожен з блоків робить згортки 1×1 , 3×3 , 5×5 та max-пулінг. Таким чином на кожному рівні вилучаються ознаки різного масштабу. Якщо масштаб занадто великий для поточного рівня – він розпізнається на наступному рівні. Після Inception модулів йде агрегація за середнім та softmax шар. Для пришвидшення тренування використовуються ще допоміжні так звані training heads, тобто на проміжних рівнях додається вихід, який дозволяє сигналу швидше доходити до нижніх рівнів та є додатковою регуляризацією.

Архітектура ResNet [8] – основна ідея таких мереж полягає у введенні в архітектуру так званих обхідних з'єднань, які дозволяють просто пропустити

сигнал далі. Такий прийом дозволяє посилювати сигнал на шляху до більш глибоких шарів та також покращити розповсюдження градієнта. Мережа також використовує *batch normalization*, тобто нормалізацію значень всередині кожної підгрупи надходжень даних. Наявність обхідних шарів дозволила натренувати модель зі 152 шарів, при цьому зберігаючи меншу складність ніж VGGNet.

Алгоритми глибокого навчання пропонуються безперервно, і в багатьох наборах даних постійно створюються нові записи. Наприклад, у задачі класифікації зображень для різних видів зображень, за п'ять років, завдяки постійному вдосконаленню моделі глибокого навчання рівень помилок знизився до 3,5%, що вище, ніж точність звичайних людей. Однак, важливо відзначити, що згорткові мережі можуть мати труднощі у вирішенні завдань обробки малих об'єктів та у боротьбі зі спотвореннями, такими як розмив фільтрів або інтенсивний шум, які існують у реальному світі.

Через те, що успішність навчання та роботи моделі залежить від якості даних та обчислювальних ресурсів, поява нових архітектур значно загальмована. Експериментальні результати [9] показують, що глибоке машинне навчання, зокрема ЗНМ, має гірший ефект на невеликих вибірках даних, ніж традиційні алгоритми машинного навчання, такі як машини опорних векторів. Відсутність великих обсягів маркованих даних може призвести до перенавчання моделі, коли вона добре пристосовується до навчальних даних, але не може узагальнити знання до нових прикладів. Модель вивчає шум або нерепрезентативні особливості даних тренувального набору.

Друга важлива проблема – це вибір гіперпараметрів. Вибір правильних гіперпараметрів, таких як розмір шарів, кількість фільтрів, крок згортки та інші, може бути нетривіальним завданням і вимагає експериментів та налаштувань. Цей факт обмежує швидкість створення ефективної моделі згорткових мереж для спеціалізованих застосунків в рамках заданої предметної області.

У [10] було наведено структурований огляд сучасних методів розв'язання задач комп'ютерного зору, їх переваг та недоліків, і визначення невирішених проблем. В роботі зазначено перелік рішень для розв'язання задачі класифікації та ідентифікації зображень, до яких відносяться штучні нейронні мережі. Автори завдяки аналізу багатьох джерел визнали, що наразі пошук нових архітектур нейронних мереж для роботи з відеоданими вимагає наявності великої кількості репрезентативних даних та нарощування складності моделей. До найбільш ефективних методів класифікації зображень відносяться глибокі згорткові нейронні мережі. Але велика кількість параметрів налаштування потребує великих розрахункових потужностей, що робить неефективним та практично неможливим розробку ефективної моделі у малих лабораторних умовах. Натомість швидким та невеликим моделям бракує точності.

1.2 Моделі на основі ансамблю нейронних мереж

Ансамблеве навчання сумісно із глибоким навчанням є двома підходами, які домінують у галузі розпізнавання образів та класифікації зображень [11-13]. Ансамблеве навчання – це метод, який використовується для об'єднання двох або більше алгоритмів ML для отримання вищої продуктивності порівняно з тим, коли складові алгоритми використовуються окремо. Існує три основні методи ансамблевого навчання: пакетування (bootstrap aggregating, bagging), підсилювання (бустінг, boosting) та стекування (stacking, stacking aggregation).

Пакетування (рис. 1.2) – це метод машинного навчання, який передбачає розділення навчальних даних для кожного базового класифікатора за допомогою випадкової вибірки (бутстрап) для генерації різних підмножин (пакетів), кожна з яких використовується для навчання окремого базового класифікатора. Потім результат роботи базових

класифікаторів об'єднуються за допомогою мажоритарного голосування для отримання більш сильного класифікатора.

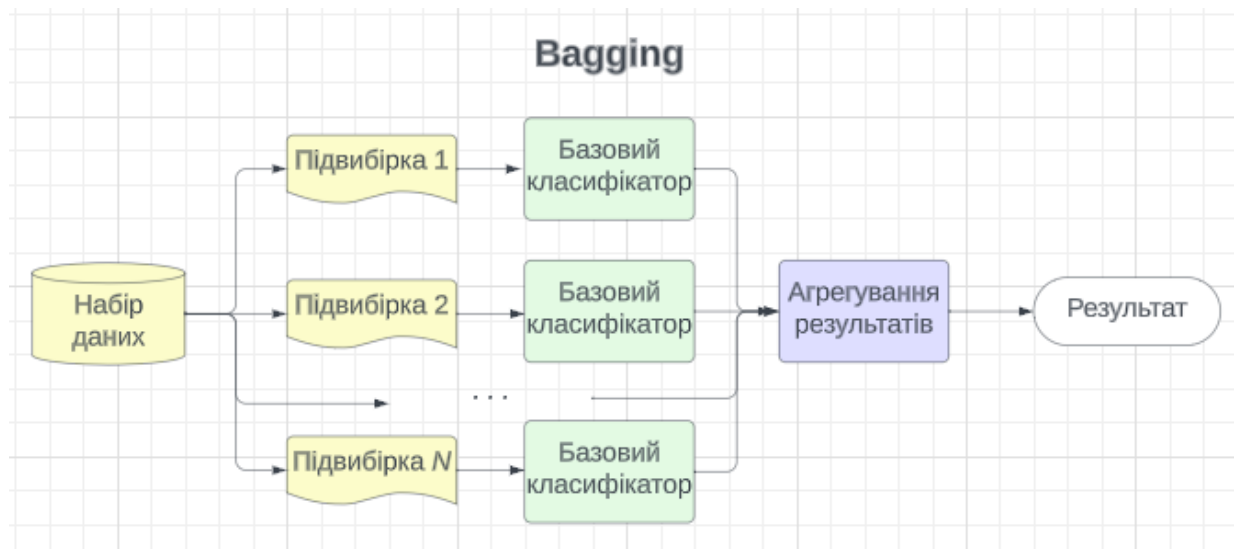


Рисунок 1.2 – Схема методу пакування

Підсилювання (рис. 1.3) – це техніка машинного навчання, здатна перетворити набір слабких класифікаторів на сильний. При цьому, слабким класифікатором вважається такий, що працює трохи краще, ніж випадкове вгадування. У свій час сильні класифікатори – це ті, що досягають хорошої точності і які є ядром, на якому будуються алгоритми бустінгу. Зокрема, бустінг-методи використовують вхідні дані для навчання слабкого класифікатора, обчислюють його передбачення, відбирають неправильно класифіковані елементи навчальної вибірки і навчають наступний слабкий класифікатор з акцентом на помилках попереднього.

Суть методу стекового узагальнення (рис. 1.4) складається у навчанні окремого алгоритму ML (мета-модель або модель першого рівня) об'єднувати прогнози від двох або більше базових класифікаторів (моделей нульового рівня), навчених на безпосередньо вхідному наборі даних. На вхід базовим класифікаторам поступають нерозмічені дані, а отримані від них мітки разом із істинними застосовуються в якості навчального набору для мета-моделі. Стекінг корисний у ситуаціях, коли кілька моделей ML є

унікально вправними у виконанні певного завдання. Тоді підхід стекінгу використовує окрему модель ML, щоб дізнатися, коли використовувати прогнози від різних моделей.

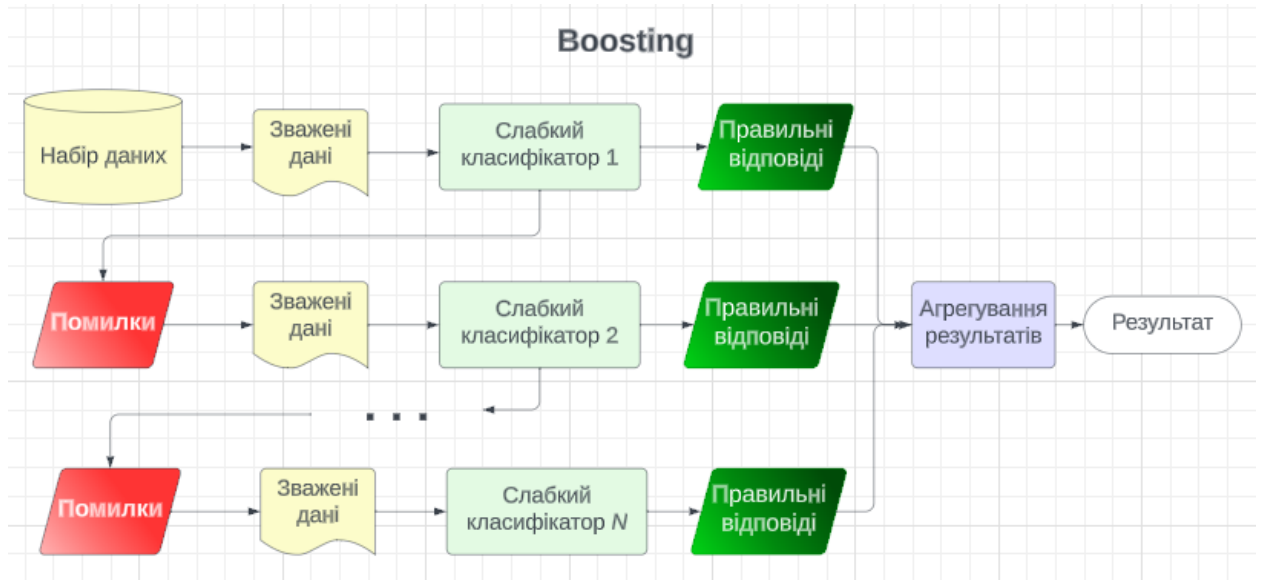


Рисунок 1.3 – Схема методу підсилення

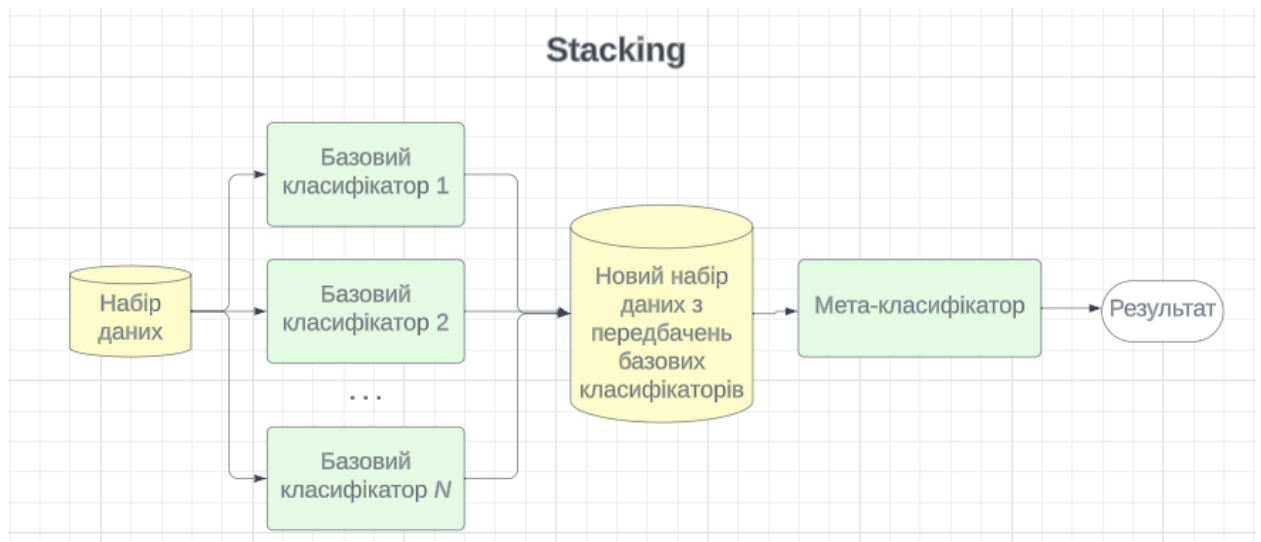


Рисунок 1.4 – Схема методу стекового узагальнення

Деякі обмеження алгоритмів ML включають: те, що вони призводять до моделей з високою дисперсією, високим упередженням і низькою точністю [14, 15]. Однак кілька досліджень показали, що ансамблеві моделі часто досягають вищої точності, ніж поодинокі моделі ML [16]. Ансамблеві методи

можуть обмежити помилки дисперсії та зміщення, пов'язані з одиночними моделями ML; наприклад, пакування зменшує дисперсію без збільшення зміщення, тоді як підсилення зменшує зміщення [17-19]. Загалом, ансамблеві класифікатори є більш надійними і працюють краще, ніж окремі.

Загалом, ансамблеві методи можна класифікувати на паралельні та послідовні ансамблі. Паралельні методи навчають різні базові класифікатори незалежно один від одного і об'єднують їхні прогнози за допомогою комбінатора. Популярним паралельним ансамблевим є метод bagging та його розширення – алгоритм випадкового лісу [20].

Паралельні ансамблеві алгоритми використовують паралельну генерацію базових класифікаторів для заохочення різноманітності членів ансамблю. Тоді як послідовні ансамблі не підбирають базові моделі самостійно. Вони навчаються ітеративно, так що моделі на кожній ітерації вчаться виправляти помилки, зроблені попередньою моделлю. Популярним типом послідовного ансамблю є алгоритм бустінгу [21].

Крім того, паралельні ансамблі можна класифікувати на гомогенні та гетерогенні, залежно від однорідності базових моделей, що навчаються. Гомогенні ансамблі складаються з моделей, побудованих за допомогою одного алгоритму ML, тоді як гетерогенні ансамблі складаються з моделей, побудованих за допомогою різних алгоритмів [22-24]. Основною проблемою в гетерогенних ансамблях є отримання найефективнішого методу для об'єднання прогнозів різних базових учнів [22]. Натомість головна проблема методів однорідних ансамблів полягає в тому, щоб забезпечити різноманітність базових учнів, навіть якщо вони використовують один і той самий алгоритм навчання. Тому для досягнення різноманітності в ансамблі були розроблені методи бутстрапу, такі як випадковий ліс [25], та методи бустінгу, такі як AdaBoost [26].

Ансамблеве навчання застосовувалося переважно до традиційних алгоритмів ML, натомість дана стратегія може бути впроваджена до моделей неглибоких нейронних мереж, складаючи конкуренцію глибоким згортковим

архітектурам. Авторами [27] було виявлено, що глибока нейронна мережа Residual Network (ResNet) поводитья подібно ансамблю неглибоких нейронних мереж. Було запропоновано перероблення даної моделі як набору шляхів, які поводитья як ансамбль, в тому сенсі, що вони не сильно залежать один від одного. Більшість градієнтів в мережі зі 110 шарів можна отримати зі шляхів, які мають глибину всього 10-34. Таким чином застосування ансамблевого методу спрощує навчання моделі загалом без втрати ефективності

Той же підхід було досліджено у [28] і доведено, що його застосування не тільки покращує точність, але й дає значний приріст в швидкодії в тому сенсі, що її можна обчислювати паралельно на декількох процесорах.

Розвиток цієї ідеї у дослідженні [29] на базі зображень CIFAR-10 показав, що застосування широкої, у сенсі розпаралеленої, нейронної мережі навіть при глибині у 16 шарів надає можливість отримати точність та швидкість, яка перевищує інші глибокі «residual networks».

У джерелі [30] досліджено застосування вейвлет-перетворення як засобу попередньої обробки зображення у задачі класифікації зображень із застосуванням ансамблю однотипних штучних нейронних мереж. В роботі було застосовано бази даних CIFAR-10 для класифікації сцени та KDEF для розпізнавання емоцій. Основною метою дослідження є розробка методу, що підвищує точність класифікації традиційних моделей глибокого машинного навчання, та порівняння ефективності різних архітектур ансамблів штучних нейронних мереж відповідно до запропонованого методу.

Автор розділяє вхідне зображення на високо- та низькочастотні ознаки та застосовує кожену матрицю ознак (вертикальних, горизонтальних, діагональних ознак та зменшену копію вхідного зображення) в якості входу до ансамблю нейронних мереж – або згорткових штучних нейронних мереж, або глибоких автокодувальників (Stacked Denoising Autoencoders – SDA). Обробка високочастотних ознак здійснюється або у паралельних окремих ШНМ CNN-WAV4, або у блоці об'єднаної обробки високочастотних ознак

(вертикальних, горизонтальних, діагональних) CNN-WAV-2. Також порівнюється застосування різних методів прийняття фінального рішення про клас об'єктів: із застосуванням багат шарового перцептронну, ELM або логічного OR-блоку.

Результати дослідження показали, що застосування запропонованих архітектур ансамблів паралельних згорткових нейронних мереж із об'єднуючим OR-блоком призводить до максимальної точності у 93,9% (KDEF) та 86,11% (CIFAR-10) у разі застосування окремої обробки кожної матриці деталей вейвлетів CNN-WAV4, що перевищує результати застосування окремо ЗШНМ (82,5 % та 81,95% відповідно) або SDA (48,64% та 14,5% відповідно). Висока точність моделі спричиняє підвищену вартість обчислень – значення складності $O(N)$ становить $1,12 \cdot 10^{11}$, тоді як складність другої за точністю моделі CNN-WAV2 із логічним OR-блоком становить $5,58 \cdot 10^{10}$ при точності 86,1%.

У роботі [31] запропоновано алгоритм, заснований на базі ансамблевого навчання із застосуванням структури з різних новітніх архітектур згорткових нейронних мереж для задачі розпізнавання зображень. Модель застосовує беггінг класифікатор для прийняття остаточного рішення про клас об'єкту, опираючись на вихідну інформацію з сукупності незалежних паралельних нейронних мереж різних конфігурацій: ResNet, DenseNet, DenseNet-BC та Inception-ResNet-v2. Для блоку комбінації виходів всіх ШНМ випробовуються різні правила голосування: жорстке голосування (Hard Voting – HV), в якому меншість голосів підкоряється більшості; жорстке голосування із порогом (HV_n), в якому необхідно як мінімум n голосів для того, щоб приймати рішення, а також зважене (Weight Vote) та м'яке (Soft Vote) імовірнісне голосування.

Експерименти на базі даних CIFAR-10 показали, що застосування запропонованого алгоритму разом із жорстким голосуванням HV₁₂ дає точність класифікації у 98,89%, що перевищує точність окремо усіх застосованих нейронних мереж на 3-16%. Результати даної роботи

засвідчують, що застосування ансамблевої моделі з глибоких нейронних мереж є більш ефективним рішенням з точки зору точності, але при цьому час, витрачений на навчання кожної внутрішньої ШНМ, та час на виконання роботи всієї моделі є досить великим, що робить неможливим застосування даного рішення в умовах реального часу.

У джерелі [32] було використано ансамбль дрібних згорткових нейронних мереж для класифікації статі людини у відеопотоці. Навчання проводилось на наборі даних Labeled Face in the Wild, а тестування на об'єднанні наборів CelebA і Adience. Одним з розглянутих підходів до розв'язання задачі була незалежна класифікація кадрів без урахування історії під час процесу остаточного вибору класу, що зводиться до простої задачі класифікації зображень. В результаті дослідження було виявлено, що точність класифікації кадрів (зображень) ансамблевої моделі порівняно з окремо застосованими згортковими мережами вища на 1,77% і становила 86,71%. При цьому швидкість роботи моделі, вказана авторами статті, дозволяє використовувати цей метод для онлайн обробки потоку даних.

У статті [33] описано розробку ансамблів з двошарових нейронних мереж прямого поширення помилки для розв'язання задачі класифікації зображень рукописних цифр з набору MNIST із застосуванням бібліотек Tensorflow та Keras.

Було розглянуто різні архітектури ансамблів з 3-9 нейронних мереж та двома різними способами агрегації рішень: за допомогою усереднення або керуючої нейронної мережі. В ході дослідження було виявлено, що найкращі результати досягаються при використанні усереднюючого ансамблю. До подібного висновку дійшли автори у [30]. Точність класифікації такого ансамблю з 5-9 ШНМ збільшувалась в межах 1% у порівнянні з результатами окремої двошарової нейронної мережі. Отже, усереднення результатів є вигідним кроком, бо різні нейронні мережі допускають різні помилки, спричинені випадковою ініціалізацією початкових ваг нейронних зв'язків.

Було також встановлено, що загальний час навчання усього ансамблю з двошарових та тришарових нейронних мереж значно менший, ніж час навчання однієї неглибокої згорткової нейронної мережі: 16,5 хвилин для ансамблю з 9 мереж проти більше ніж 2 годин для згорткової. При таких витратах часу, ансамблева модель досягла 98,15% точності, а ЗШНМ – 99,02% точності класифікації. Тобто при відносно невеликому виграші в точності згорткова мережа займає у 7,75 разів більше часу при навчанні, що віддає перевагу у застосуванні ансамблевої моделі.

Робота [34] присвячена вибору архітектури згорткової неронної мережі для класифікації раку шкіри та порівнянню продуктивності між моделями ансамблю та окремою моделлю. Використовуючи дев'ять різних архітектур згорткових нейронних мереж було досягнуто висновку, що найкращим підходом до класифікації зображень меланоми є створення ансамблів із кількох моделей. Крім того, було досліджено ефективність між двома різними стратегіями вибору базових моделей для формування ансамблю: випадковим вибором і за значенням функції втрат на валідаційній (перевірочній) вибірці. Було виявлено, що ансамбль, сформований за другим методом, має трохи краще середнє значення показників ефективності та трохи меншу мінливість. Цей метод має більше переваг скоріш для менших моделей ансамблю, натомість для більших моделей вибір міг бути довільним без значних втрат результативності.

Аналогічно у [35] запропоновано два різні методи класифікації раку шкіри зі зменшенням складності моделі за допомогою стратегії один проти всіх (ОПВ): використання єдиної ЗШНМ проти ансамблю з семи моделей бінарних класифікаторів. В результаті окрема згорткова нейронна мережа отримала 77% точності класифікації у виявленні захворювань шкіри за допомогою семи класів, поєднання комітету нейромереж та підходу один проти всіх показало 92,90% точності. Дослідження проводились на наборі даних HAM10000 із зображень дермоскопії.

Таким чином застосування ансамблевого навчання дозволяє отримати перевагу у швидкості та обчисленнях зі збереженням точності аналогічної більш складним моделям.

1.3 Класифікація на базі структурного опису зображення

Навчання згорткових нейронних мереж для зображень великого розміру зі збереженням точності при виявленні об'єктів різних масштабів є громіздкою задачею. На вхід моделі подається зображення цілком, яке містить надлишкову інформацію. Поступове сканування зображення згортковими фільтрами та масштабування карт ознак шарами пулінгу потребує великої глибини моделі, що відображається на складності навчання такої моделі. Разом із цим інтерпретація роботи самої моделі залишається чорним ящиком. Альтернативним підходом до цього можна вважати застосування замість цілого зображення його структурного опису [36, 37].

Структурний опис зображення є набором характерних особливостей, вилучених засобами комп'ютерного зору, і відображає властивості об'єкту, що необхідно розпізнати або класифікувати. Прикладами елементів структурного опису можуть виступати межі об'єктів або сегментація зображення, кути, перепади яскравості та точки неоднорідності, ключові точки, виявлені за допомогою спеціальних алгоритмів-детекторів.

Класифікація зображень на основі структурного опису має багато переваг, серед яких: пришвидшення роботи алгоритму обробки зображень, скорочення розмірності простору ознак, універсальна можливість структурного аналізу об'єктів, стійкість до фонових впливів, геометричних перетворень та помилкових об'єктів.

Так як сегментація зображення сама по собі є нетривіальною задачею, а виявлення меж об'єктів є неефективним для зображень зі складною текстурою, ці способи використовують нечасто. Кути або точки

неоднорідності інтенсивності на зображенні в чистому вигляді непридатні для задачі розпізнавання в силу своєї недискримінативності – об'єкти на зображенні можуть підлягати афінним перетворенням, через що інформація про розташування кутів не буде інформативним фактором (схожі кути можуть бути присутні на різних об'єктах, позиції об'єктів не зафіксовані на одному місці). Натомість зазначені обмеження можливо обійти при застосування ключових точок та їх локальних дескрипторів.

Ключова точка – це точка на зображенні, яка визначається певною інтенсивністю навколо неї через досягнення значення екстремуму локальної особливості. Ключова точка обирається відповідно до області навколо неї, яка описується дескриптором, представленим чисельним вектором. Ключова точка і дескриптор разом визначають ознаку. До найпоширеніших дескрипторів ключових точок відносяться SIFT, SURF, ORB, BRISK, KAZE.

Так як у більшість новітніх алгоритмів розрахунку дескрипторів включені параметри орієнтації та масштабу, то такі ознаки виявляються інваріантними по відношенню до обертання зображення, зміщення, зміни масштабу або яскравості. Такі алгоритми як AKAZE і ORB, крім того, забезпечують можливість зіставлення зображень, співвідносячи однакові локальні ознаки один з одним.

Порівняльний аналіз [10] низки детекторів показав значну перевагу у застосуванні бінарних дескрипторів, таких як ORB та BRISK, в сенсі швидкості розрахунку, стійкості до повороту, розмиття зображення, впливу стиснення. Також такий алгоритм, як AKAZE, може скласти конкуренцію останнім двом за кількістю знайдених збігів та викидів, поступаючись у кількості знайдених особливостей та середньою швидкістю обчислень на одну точку.

Дослідження [38, 39] показали, що застосування бінарних дескрипторів як структурного простору ознак, в якому формуються описи окремих об'єктів, разом із підходом до класифікації bag-of-words дозволяє домогтися

суттєвих покращень у швидкості при обчисленні релевантності описів зображень в задачі класифікації зображень бази еталонів.

Також застосування бінарних дескрипторів разом із моделями нейронних мереж, що самонавчаються (наприклад, мережа Кохонена), дозволяє отримати систему, швидкість роботи якої співставна з потоковою обробкою зображень [40-42]. Кожне зображення бази еталонів представляється кластером дескрипторів КТ. Такий підхід дозволяє зменшити розмір структурного опису до одного представника – центроїду кластеру.

Однак можлива наявність великої дисперсії всередині кластеру може спричинити помилкові класифікації, тому розглядається можливість представлення опису кожного зображення у вигляді системи кластерів і, відповідно, їх центрів. Експерименти [43-45] показали, що проведення кластеризації на даних структурного опису зображення забезпечує необхідну результативність класифікації, дає можливість суттєво зменшити обсяг обчислень (у порівнянні з класичним голосуванням дескрипторів) та приймати рішення про належність до того чи іншого класу за окремими представниками множини КТ, що важливо у прикладному сенсі.

До переваг застосування класифікаторів на базі структурного опису з дескрипторів ключових точок відноситься: швидкість навчання та виконання, інваріантність до афінних перетворень зображень, розрахункова невибагливість (порівняно з моделями глибокого навчання), можливість врахувати ієрархічну або геометричну структуру об'єкту, універсальність стосовно предметної області застосування.

Недоліками можуть виступити залежність результативності від алгоритму детектування ключових точок та обчислення їх дескрипторів, кількості та репрезентативності обраних дескрипторів, обсягу бази еталонних зображень, а також їх характеру та розрізненості між собою.

В ході дослідження в якості базового класифікатора для ансамблевого навчання буде застосовуватись мережа Кохонена в якості методу

формування системи центрів кластерів з дескрипторів кожного окремого еталону. Вибір саме цього алгоритму самонавчання обумовлений його швидкістю обчислень та простотою. Детальний огляд математичного апарату мережі Кохонена буде розглянуто в розділі 2.

1.4 Постановка задачі дослідження

Застосування ансамблевих моделей разом із апаратом структурного опису зображень за множиною дескрипторів КТ є актуальним завданням у сфері розпізнавання зображень для комп'ютерного зору. Практичний інтерес викликає розроблення методів класифікації та отримання прийнятної результативності за рахунок застосування комітету нейронних мереж. Для формування множини дескрипторів КТ обрано детектор ORB. Базовим класифікатором комітету обрано мережу векторного квантування Кохонена.

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

Метою дослідження є впровадження моделі комітету мереж векторного квантування Кохонена у процес класифікації, що базується на опису у формі множини бінарних дескрипторів ключових точок.

Для досягнення мети необхідно вирішити такі завдання:

- розробити модель комітету (ансамблю) мереж Кохонена;
- визначити процедури попередньої обробки множин дескрипторів та провести аналіз їх впливу на результативність;
- реалізувати програмну модель комітету нейронних мереж за множиною еталонних описів класів;
- експериментально дослідити результативність розробленого методу класифікації для прикладної бази зображень.

2 МАТЕМАТИЧНА МОДЕЛЬ КОМІТЕТУ МЕРЕЖ КОХОНЕНА

2.1 Вибір архітектури базового класифікатора ансамблю

Мережа Кохонена (мережа векторного квантування Кохонена) – це різновид нейронних мереж прямого поширення, які застосовуються в задачах кластерування, кодування, зменшення розмірності. Особливо ефективна в умовах відсутності інформації про статистичний розподіл досліджуваного процесу, наприклад, при потоковому надходженні вхідних образів.

Під векторним квантуванням розуміють представлення деякої сукупності векторних даних x з множини $A \in R^n$ невеликою кількістю опорних (еталонних) або кодуючих даних w з множини $B \in R^m$, де $m \leq n$; оптимальне квантування простору сигналів, яке мінімізує середню очікувану помилку квантування. Інакше кажучи, множина векторів x розбивається на кінцеве число класів, кожному з яких призначається опорний представник w_i . Множина всіх w_i утворює кодову множину класифікатора.

Векторне квантування подібно до задачі кластеризації даних та здійснюється за допомогою методу «найближчого сусіда», причому під «найближчим» розуміється вектор, що задовольняє різним вимогам, наприклад, мінімізації міри відстані або максимізації міри подібності всередині кластеру.

Навчання мережі Кохонена будується за модифікованим правилом Хебба та може впроваджуватись як у контрольованому (з учителем) так і неконтрольованому (без учителя) форматі. При тренуванні мережі з учителем цей процес називається Learning Vector Quantization (LVQ) – векторне квантування, що навчається – та воно застосовує мітки приналежності вхідного вектору певному класу. При навчанні без учителя застосовуються мапи, що самоорганізуються (Self-Organizing Map, SOM).

В класичному розумінні нейрони в мапах Кохонена утворюють решітку, в якій сусідні нейрони пов'язані латеральними зв'язками, завдяки

чому в процесі конкуренції нейрони навколо переможця можуть також змінювати свої ваги (рис. 2.1). Але мапи Кохонена зазвичай використовують в задачах візуалізації, стиснення даних та зменшення розмірності, тоді як LVQ застосовують для статистичної класифікації та розпізнавання. В тому вигляді, в якому мапи SOM були розроблені, виявити який нейрон представляє який саме клас всередині програми можливо або наочно (вписуючи цю інформацію власноруч після візуального аналізу мапи), або перебором на базі міри близькості чи відстані (в разі вдалої кластеризації).

Зважаючи на перелічене вище, для реалізації поставленої задачі класифікації зображень буде застосовано проміжну архітектуру між цими двома варіантами мереж векторного квантування, а саме таку, в якій нейрони не пов'язані між собою топологічно (не утворюють решітку), є незалежними, не застосовують мітки класів і за такт навчання оновлюються ваги лише одного нейрона, реалізуючи модель «Переможець отримує все» (Winner Takes All - WTA).

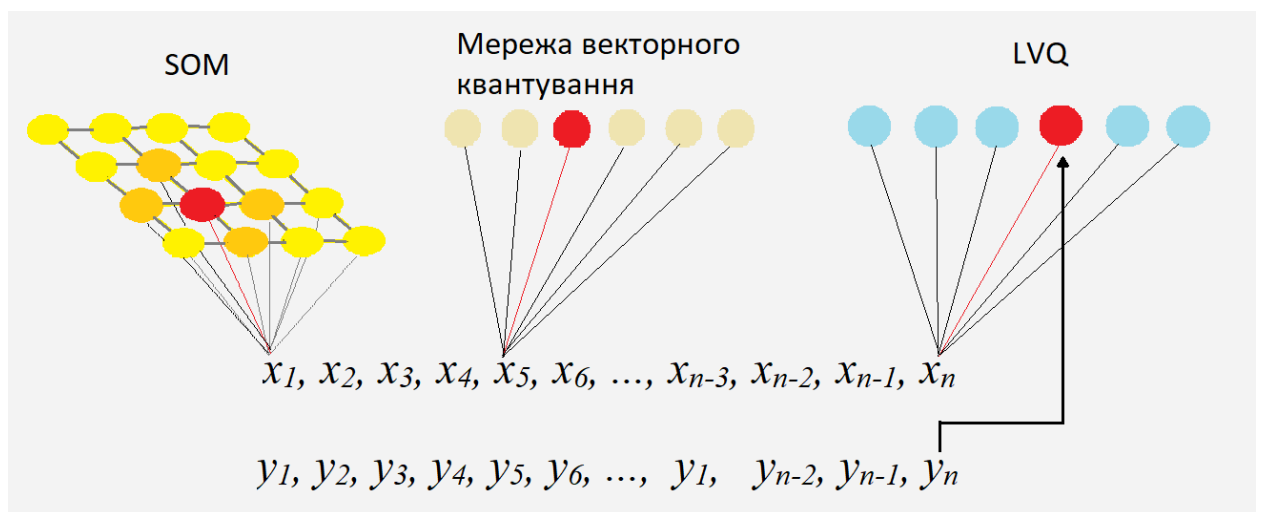


Рисунок 2.1 – Варіанти архітектур мереж Кохонена

Так як модель WTA не має на увазі встановлення певного околу «переможця», то немає ніяких причин очікувати збереження просторової упорядкованості кодуєчих векторів. Так як мережа зведе набір вхідних образів x до набору кластерів, представлених центроїдами (самими

нейронами), то можна казати, що значення самих ваг нейронів w_j будуть відображати певний статистичний розподіл елементів всередині образу (в нашому випадку розподілу дескрипторів КТ на кластери, які утворюють характерні риси певного класу).

Вибір неконтрольованого способу навчання мережі векторного квантування обумовлений тим, що в рамках бази дескрипторів одного класу мітки дескрипторів будуть однаковими, а отже щоб мережа навчилася відрізняти «свої» дані від «чужих» треба буде розширити тренувальний набір даними з інших еталонних описів. До того ж таку процедуру треба буде зробити для кожної мережі в ансамблі. Це призведе до збільшення обчислювальних витрат та часу навчання.

Таким чином, при навчанні нейромережі на наборі дескрипторів окремого класу, буде створено стиснутий структурний опис заданого еталону. Об'єднання нейронних мереж Кохонена, навчених окремо на дескрипторах КТ кожного класу бази еталонів, в ансамблеву структуру дозволяє:

- зробити новий структурний опис класу більш детальним за рахунок збільшення числа нейронів-представників класу (тоді як у звичайних мапах Кохонена передбачається, що один нейрон кодує один клас);

- детермінувати кількість нейронів призначених на клас (тоді як у звичайних мапах число нейронів, які будуть представляти кожен клас, достеменно невизначено);

- визначити які саме нейрони відповідають якому класу за рахунок привласнення кожному нейрону індексу та навчанню його на даних свого класу;

- зменшити кількість невизначених нейронів, які з'являються через вплив на їх навчання вхідних елементів x , розташованих на границі класів.

2.2 Математичний апарат навчання мережі Кохонена

Алгоритм навчання мережі Кохонена складається за наступних кроків:

Крок 1. Ініціалізація синаптичних ваг w_j , наприклад, випадковими значеннями. Ініціалізація початкового значення коефіцієнту швидкості навчання.

Крок 2. Вилучення одного вектору x з тренувального набору. Розрахунок відстаней до всіх нейронів.

Крок 3. Визначення індексу нейрону-переможця з мінімальною відстанню до поточного вектору.

Крок 4. Перерахування значення ваг для заданої околиці нейрону-переможця (для WTA околиця замикається на самому переможці).

Крок 5. Оновлення значення коефіцієнту навчання.

Крок 6. Перевірка умови зупинки: в разі незадоволення умови перейти до Кроку 2, інакше – кінець тренування.

Мережа Кохонена має можливість ефективно навчатись як в потоковому, так і в пакетному режимі. В пакетному режимі з фіксованим значенням коефіцієнту навчання алгоритм зводиться до k -середніх. Натомість навіть за наявності кінцевої множини тренувальних даних цілком, мережу можна навчити потоковим способом за умови циклічного вилучення кожного нового вектору з замкненого тренувального набору та поступового зменшенням коефіцієнту навчання після кожної ітерації. Умовою зупинки алгоритму на поточній ітерації може бути досягнення ліміту тренувальних епох або досягнення різницею помилки квантування між двома ітераціями певного порогового значення.

Якщо сума ваг для кожного нейрона є величиною постійною й $\|\bar{w}_i\|^2 \approx 1$, а вхідні вектори нормалізовані так, що $\|\bar{x}\|^2 = 1$, то як показав Кохонен, час визначення нейрона-переможця може бути скорочено [46]. При цьому нейрон-переможець з індексом c визначається шляхом мінімізації евклідової норми.

$$c = \arg \min_j (\|x - w_j\|^2). \quad (2.1)$$

Так як в якості структурного опису зображення було обрано набір дескрипторів ORB, то вхідні вектори будуть бінарними та довжиною 256 біт. Через це для нормалізації даних буде використано норму Хемінга, яка розраховується як кількість одиничних бітів вектору. Таким чином бінарні вектори переходять у простір векторів чисел з плаваючою комою і для розрахунку «найближчого» нейрона до вхідного необхідно обрати міру близькості або відстані, яка б відповідала евклідовому простору нормалізованих даних.

В якості метрики може бути обрана як L2-норма (евклідова відстань), так і L1-норма (Манхетенська відстань) як більш проста в розрахунку. В даній роботі було обрано Манхетенську відстань в якості міри відстані векторів.

В разі неконтрольованого навчання мережі векторного квантування у дискретному випадку (при поданні на кожному такті навчання нового вхідного образу) відбувається корекція ваг нейрона-переможця за формулою:

$$w_c(t+1) = w_c(t) + \alpha(t)(x - w_c(t)), \quad (2.2)$$

де c – індекс нейрона-переможця;

$d(x, m_i)$ – обрана міра відстані;

w_c – вага нейрона-переможця;

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

t – ітерація в циклі навчання;

x – вектор вхідного сингалу.

Коефіцієнт навчання $\alpha(t)$ може бути обраний постійним значенням або у процесі навчання монотонно зменшуватись, залежно від ітерації навчання. Під ітерацією мається на увазі один такт вибору нейрона-переможця. В даній

роботі буде застосовано обернено пропорційний закон для розрахунку коефіцієнта швидкості навчання. Формула для фіксованої кількості епох та для потокового навчання може приймати наступний вигляд:

$$\alpha^{epochs}_t = \left(1 - \frac{t}{T}\right) * \alpha_0, \quad (2.3)$$

$$\alpha^{stream}_t = \frac{1}{t} \alpha_0, \quad (2.4)$$

де t – поточна ітерація;

T – загальна кількість ітерацій, $T = \text{кількість епох} \times \text{кількість зразків даних}$;

α_0 – початковий коефіцієнт навчання;

α_t – коефіцієнт навчання на ітерації t .

Окрім нормалізації даних в якості попередньої обробки бази дескрипторів КТ еталонів можна застосувати фільтрацію дескрипторів як відбір n перших значень, відсортованих за максимумом метричного коефіцієнта інформативності [47]:

$$V(z, E) = \rho_m(z, \bar{E}_k) - \rho_m(z, E_k), \quad (2.5)$$

де $V(z, E)$ – показник інформативності дескриптора z у базі еталонів E ;

$$\rho_m(z, \bar{E}_k) = \min_{v, i \neq k} \rho(z, e_v(i)) - \text{мінімальна відстань від дескриптору } z \text{ до}$$

елементу бази, що не належить класу E_k ;

$$\rho_m(z, E_k) = \min_{v, i=k} \rho(z, e_v(i)) - \text{відстань від } z \text{ до ближчого елемента класу } E_k$$

(за виключенням нульової відстані до себе).

Таким чином за рахунок деякої втрати інформації (рівень втрат визначається порогом фільтрації) можна прискорити час навчання моделі на наборі найбільш унікальних векторів свого класу. В разі, коли об'єм вихідного та первісного набору дескрипторів в результаті встановлення

порогу фільтрації однаковий, значення всередині опису будуть відсортовані у порядку від найбільш до найменш значущих. Завдяки цьому при найперших оновленнях ваг нейронів, коли значення коефіцієнту швидкості навчання є максимальним, це дозволить одразу наблизити значення ваг w до найбільш імовірних значень центрів кластерів.

2.3 Конструкція ансамблю нейронних мереж Кохонена

Нехай задано деяку базу E описів зображень еталонів розміром N : $E = \{E_1, E_2, \dots, E_N\}$. Кожен еталонний опис E_i представляє окремий клас зображень та має вид скінченної множини дескрипторів КТ зображення $E_i = \{e_\nu(i)\}$, де $\nu \in [1; s]$, s – фіксована кількість дескрипторів КТ в описі еталону E_i [48, 49].

Кожен окремий дескриптор $e_\nu(i)$ описує деякий окіл КТ зображення та є елементом векторного простору R^n скінченної розмірності n , тобто $e_\nu(i) \in R^n$ з дійсними, цілими або бінарними компонентами. При використанні детектору ORB $n = 256$ та компоненти вектору є бінарними. Для спрощення аналізу та збалансування загальної вибірки даних серед класів встановимо значення параметру s для кожного опису із множини E однаковим, тобто $\text{card}(E_1) = \text{card}(E_2) = \dots = \text{card}(E_N) = s$.

Застосуємо відображення $E \rightarrow T$ із простору образів (множини дескрипторів) у множину T непересічних кластерів, сформованих мережами векторного квантування Кохонена. У результаті кожний образ E_i еталону трансформується до виду фіксованого числа M його непересічних підмножин $T_k(E_i)$:

$$E_i = T(E_i) = \bigcup_{k=1}^M T_k(E_i), T_k(E_i) \cap T_j(E_i) = \emptyset. \quad (2.6)$$

можливої дії завад на зображенні та фільтрації у процесі класифікації появи хибних КТ.

У процесі класифікації набору дескрипторів Z вхідного зображення для кожного вектору z_v розраховується індекс нейрона-переможця із множини W у комітеті за конкурентним правилом (2.1) та накопичуються голоси r_i на користь цього класу за формулою:

$$r_i = \sum_{k=1}^M f_k [z_v \rightarrow \{w_j(E_i)\}], \quad (2.7)$$

де f_k – логічна функція, що визначає віднесення елемента z_v до відповідного центру з номером j кластера еталону E_i .

Прийняття кінцевого рішення про клас зображення здійснюється за мажоритарним голосуванням на базі збережених міток класу r_1, r_2, \dots, r_N голосів для усіх z_v із встановленням порогу, нижче якого вважається, що клас об'єкту є невизначеним:

$$Z \rightarrow E_j: j = \arg \max_i (r_i) \text{ if } \max(r_i) > \delta, \quad (2.8)$$

де δ – поріг прийняття рішень;

r_i – множина голосів за i -й еталон.

Якщо максимальний відсоток голосів не перевищує хоча б $100/m\%$, тобто розподіл голосів між класами приблизно рівномірний, то можна вважати, що клас зображення є невизначеним. Для зображень, які мають характерні особливості та добре відрізняються один від одного значення порогу може бути встановлено на рівні 40-50% (або 0,4-0,5 у відносному вираженні).

2.4 Оцінювання ефективності класифікації зображень

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

$$E_q = \frac{1}{n} \sum_{i=1}^n \|x_i - w_c(x_i)\|^2, \quad (2.9)$$

де E_q – помилка квантування;

$w_c(x_i)$ – вага нейрона-переможця для заданого вектору x_i ;

n – загальна кількість нейронів у комітеті.

В разі використання замість евклідової метрики Манхеттенську відстань та застосовувати нормалізовані вхідні вектори та вектори ваг нейронів, тоді значення E_q також буде нормалізоване на проміжку $[0;1]$. В результаті класифікації отримано вектор $E_q(Z) = [E_{q,1}, E_{q,2}, \dots, E_{q,N}]$, де Z – це набір дескрипторів КТ вхідного зображення, а N – загальна кількість класів у базі еталонів. Мінімальне значення помилки квантування в комітеті буде відповідати мережі ВК, яка представляє клас найближчий до вхідного образу. Якщо найближчий клас відповідає правильному, тоді даний результат класифікації можна вважати успішним.

Альтернативним показником оцінки може бути частка голосів відданих за правильний клас. В такому разі вже максимізація цього показника буде відповідати більш ефективному визначенню класу об'єкта.

В обох випадка має сенс встановлення порогу прийняття рішень в разі, якщо розподіл голосів або значення помилки квантування приблизно однорідні.

Оцінювання класифікації бази тестових зображень можна проводити за часткою правильно класифікованих зображень за одним з перелічених вище критеріїв:

$$Pr = \frac{n_correct(test_base, criteria)}{n_test}, \quad (2.10)$$

де Pr (precision) – коефіцієнт точності класифікації бази зображень;

$test_base$ – набір тестових зображень;

$n_correct$ – кількість успішно класифікованих зображень з $test_base$;

n_test – кількість зображень тестової бази;

$criteria$ – критерій оцінювання успішності класифікації (помилка квантування або відсоток голосів за правильний клас).

3 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ КОМП'ЮТЕРНОЇ МОДЕЛІ КОМІТЕТУ МЕРЕЖ КОХОНЕНА

3.1 Обґрунтування вибору середовища програмної реалізації

У рамках кваліфікаційної роботи проведено детальне дослідження методу із застосуванням комітету нейронних мереж в задачі класифікації зображень. Моделювання виконано на мові програмування Python у програмному середовищі Google Colaboratory із використанням бібліотеки NumPy для побудови моделі та швидкого обчислення, а також засобів бібліотеки комп'ютерного зору OpenCV [50].

Мову програмування Python було обрано через її високий рівень абстракції, зручний та зрозумілий синтаксис та наявність великої кількості готових бібліотек з машинного навчання та інтелектуальної обробки даних, таких як NumPy, SciPy, Scikit-Learn, Tensorflow та інших. Хоча Python і має повільну продуктивність (через те, що це інтерпретаторною мовою), але це компенсує наявність багатьох бібліотек, розроблених на компіляторних мовах програмування, таких як C та C++.

Середовище Google Colab дозволяє застосовувати хмарні обчислення та виділяє для цього користувачеві окрему ЕВМ із під'єднаною графічною картою, що дозволяє прискорити роботу програми. Дане середовище розроблено цілеспрямовано для дослідження даних на мові програмування Python із застосуванням так званих «ноутбуків» Jupyter, в яких виконується код окремо в кожній комірці. Це дозволяє зробити виконання та тестування програм більш контрольованим.

NumPy (Numerical Python) – це бібліотека для роботи з масивами та математичними операціями мовою Python. Вона надає ефективні з точки зору керування пам'яттю та гнучкі за використанням структури даних (такі як багатовимірний масив ndarray) і функції для обчислень, що робить її потужним інструментом для аналізу даних. У NumPy реалізовано широкий

спектр математичних і статистичних функцій, багато з яких підтримують векторну обробку даних, що дозволяє одночасно швидко обробляти великі масиви даних. Вихідний код NumPy написано на C++, що компенсує розрахункову повільність Python.

SciPy (Scientific Python) – бібліотека для наукових обчислень на мові Python. Вона надає широкий спектр функцій для наукових та інженерних застосувань, включаючи чисельне інтегрування, оптимізацію, обробку сигналів тощо. SciPy була створена наприкінці 1990-х років як відгалуження бібліотеки NumPy з метою надання набору інструментів для наукових обчислень, які не були доступні в NumPy. В даній роботі функції SciPy переважно застосовуються для ефективного обчислення відстаней між наборами дійсних векторів.

Бібліотека комп'ютерного зору з відкритим кодом OpenCV надає широкий спектр функцій для обробки зображень і відео, виявлення особливостей, об'єктів, відстеження руху тощо. Вона написана на C++ і має інтерфейс до Python, що дозволяє легко інтегрувати у широкий спектр програмних засобів.

Також OpenCV можна легко інтегрувати з іншими бібліотеками та фреймворками Python, такими як NumPy, SciPy та Scikit-Learn, що робить його чудовим вибором для машинного навчання та додатків глибокого навчання.

В даній роботі OpenCV застосовано для зчитування зображень, їх обробки, обчислення дескрипторів ключових точок та відображення [51].

3.2 Особливості програмної реалізації

У рамках кваліфікаційної роботи розроблено та порівняно результативність наступних методів класифікації зображень: метод класифікації на базі простої мережі Кохонена, метод класифікації на базі

комітету мереж Кохонена з одним нейроном (кожна мережа навчається на відокремленому наборі дескрипторів КТ свого еталону) та метод класифікації на базі комітету мереж Кохонена з k центрами. Другий метод класифікації є окремим випадком третього, коли $k = 1$.

Навчання першого методу відрізняється від інших тим, що в якості навчального набору даних використовується сукупна база дескрипторів КТ усіх еталонів, які потім розподіляються до найближчих центрів кластерів. Таким чином здійснюється класичний метод кластеризації, подібний до k -середніх, який є пакетною варіацією алгоритму навчання мережи Кохонена.

Нейрони перед початком навчання ініційовані еталонними дескрипторами, які мають найбільше значення інформативності (2.5) для свого еталону, тобто є найбільш унікальними представниками. Після навчання мережі, значення отриманих центрів кластерів відображають структуру даних в рамках бази еталонів. Центри, які далеко знаходяться один від одного у векторному просторі повідомляють про маленький перетин у множинах дескрипторів цих образів, тобто їх відмінність.

Перевагою такого методу є те, що для простих образів з маленькою дисперсією у векторному просторі достатньо одного нейрона для представництва цілого кластеру. Збільшення числа нейронів в рамках сукупної бази навчальних даних може призвести як до більш ретельного опису класу, так і до появи «мертвих» нейронів, які не мали змоги навчитись, через що збільшиться число помилкових класифікацій.

Еталонними зображеннями, які будуть використовуватись для навчання класифікатора, для даної роботи було обрано зображення колекційних монет із міфічними богами (рис. 3.1) [52]. Розмір зображень становить 770×770 пікселів. Детектором дескрипторів КТ було обрано алгоритм ORB, вектори значень якого є бінарними векторами довжиною 256 біт. В якості стандартної кількості дескрипторів КТ на еталон обрано $s=500$.

Ключові точки зображень та їх дескриптори обчислено за допомогою методу *detectAndCompute()* об'єкту детектора, створеного функцією *ORB_create()* з бібліотеки OpenCV. Відобразимо ключові точки на еталонних зображеннях за допомогою функції *drawKeypoints()*. На рисунку 3.2 координати ключової точки позначені зеленим колом.

Помітно, що в четвертого та п'ятого еталона є низка спільних ключових точок, виділених на літерах по периметру монет. Отже можна в якості попередньої обробки даних застосувати фільтрацію дескрипторів навчальних наборів за максимумом значень (2.5) та порівняти результат класифікації із застосуванням цієї процедури та без.



Рисунок 3.1 – Еталонні зображення міфічних богів [52]

У результаті сформовано тренувальний набір дескрипторів сукупною кількістю 2500 бінарних векторів. Для методів класифікації на базі навчання ансамблю сукупний тренувальний набір розділяється по $s=500$ елементів на

кожний базовий класифікатор. Задля застосування бінарних векторів у процесі навчання відповідно до (2.2) необхідно перетворити тип даних з `uint8` (ціле число розміром 1 байт) до `float` (32- або 64-байтове дійсне число).



Рисунок 3.2 – Ключові точки на еталонних зображеннях, виділені детектором ORB

Для швидкого обчислення матриць відстані між наборами векторів (дескрипторів та центрів кластерів) замість стандартних циклів Python застосовується функція `cdist()` бібліотеки SciPy, яка обчислює значення зазначеної метрики (по замовчуванню евклідової) між двома наборами векторів однієї довжини.

Для реалізації зазначених методів класифікації за парадигмою ООП було розроблено класи: *KohNet* (для роботи алгоритму векторного квантування Кохонена), *SelfLearningEnsemble* і *ParallelSelfLearningEnsemble* (для відтворення навчання та класифікації комітету нейронних мереж),

DescriptorsReduction (для фільтрації наборів дескрипторів) та *EnsembleImageClassifier* (клас-інтерфейс, об'єднуючий попередні). Зв'язки між класами зображені на UML-діаграмі (рис. 3.3).

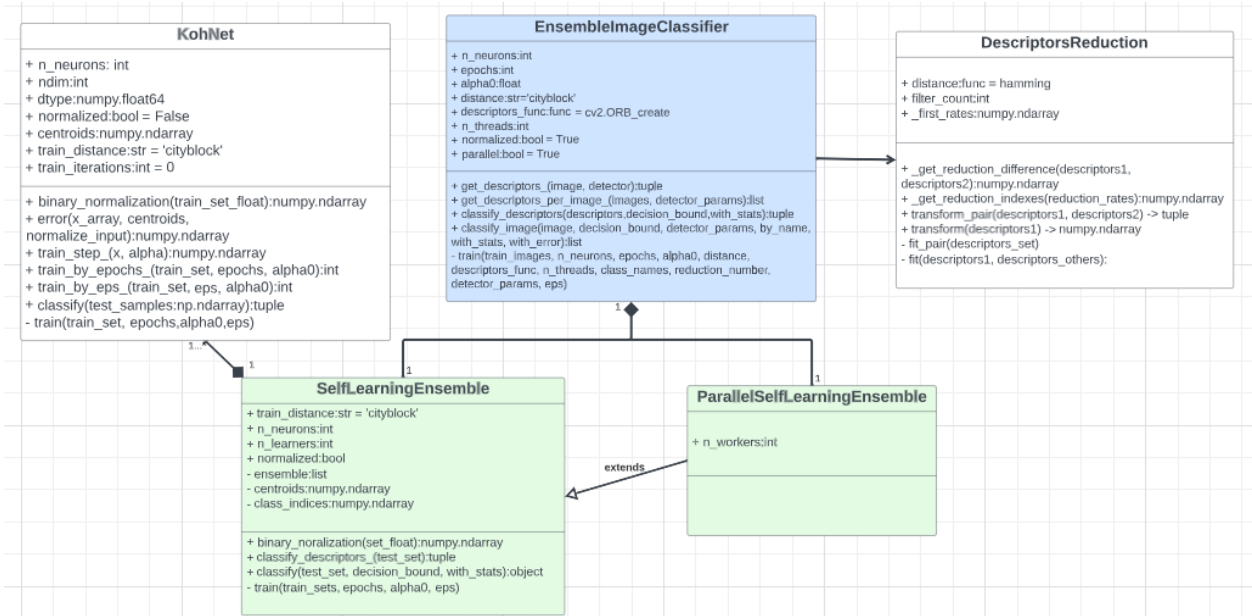


Рисунок 3.3 – UML-діаграма застосованих класів

Як видно на рисунку 3.3, клас *KohNet* є невід'ємною складовою ансамблевого класифікатора, навчання якого зводиться до навчання кожного об'єкту *KohNet*. Програмний код методу для тренування мережі *KohNet* на мові Python представлено в лістингу 3.1.

Лістинг 3.1 Програмний код функції навчання мережі векторного квантування:

```

def train(self, train_set: np.ndarray, epochs: int, alpha0: float,
eps: float=None):
    train_set_float = train_set.astype(self.centroids.dtype)
    train_set_float = self.binary_normalization(train_set_float) if
self.normalized else train_set_float
    self.centroids = train_set_float[np.random.randint(train_set.shape[0],
size=(self.n_neurons))]
  
```

```
self.train_iterations = self.train_by_epochs_(train_set_float, epochs,
alpha0) - 1
```

```
if eps is not None:
```

```
self.train_iterations = self.train_by_eps_(train_set_float, eps,
alpha0) - 1
```

Відповідно до лістингу 3.1 для навчання мережі з фіксованою кількістю епох та правилом оновлення коефіцієнту навчання (2.3) застосовується функція *train_by_epochs_()*, при зацикленому потоковому навчанні із використанням (2.4) – функція *train_by_eps_()*. Порівняння ефективності застосування кожного методу навчання досліджено у розділі 3.3.

Задля прискорення навчання комітету на базі стандартного класу *SelfLearningEnsemble* було створено клас-нащадок паралельного навчання *ParallelSelfLearningEnsemble*, який створює декілька потоків (threads) виконання, в кожному з яких окремо навчаються базові класифікатори *KohNet*. Після закінчення процесу тренування, потік повертає навчену мережу ВК. Таким чином можна значно скоротити загальний час навчання моделі. Мультипотокowe (multi-threading) програмування здійснено за допомогою засобів бібліотеки *concurrent*. Програмний код розпаралеленого тренування комітету представлено у лістингу 3.2.

Лістинг 3.2 Програмний код навчання розпаралеленого комітету мереж векторного квантування:

```
def train(self, train_sets: np.ndarray, epochs:int, alpha0: float,
eps:float=None):
    with
concurrent.futures.ThreadPoolExecutor(max_workers=self.n_workers) as
executor:
    threads = {}
    for i, (learner, train_set) in enumerate(zip(self.ensemble, train_sets)):
```

```

        threads[executor.submit(train_kohnet, learner, train_set, epochs,
alpha0, smart_init=smart_init, eps=eps)] = i
    for future in concurrent.futures.as_completed(threads):
        try:
            class_id = threads[future]
            learner = future.result()
            self.centroids[class_id*self.n_neurons:
(class_id+1)*self.n_neurons] = learner.centroids
        except Exception as exc:
            print('Generated an exception while learning KohNet: %s' % (exc))
            print(class_id, class_id*self.n_neurons,
(class_id+1)*self.n_neurons)
    return

```

У функції *train* створюється набір потоків, контрольований об'єктом *executor* класу *ThreadPoolExecutor*. Цей об'єкт запускає новий потік виконання за допомогою функції *submit*, яка приймає в якості першого аргументу функцію для виконання (в нашому випадку функцію-оболонку *train_kohnet*, в якій здійснюється запуск методу навчання класу *KohNet*), а інші аргументи є аргументами функції для виконання. Після закінчення роботи, потік повідомляє об'єкту *executor* про це, що робить можливим отримання натренованої мережі.

Клас *EnsembleImageClassifier* створено для зручного застосування комітету нейромереж при класифікації зображень. Цей клас здійснює ініціалізацію об'єктів *SelfLearningEnsemble* або *ParallelSelfLearningEnsemble* за вказаними параметрами. Всередині функції навчання та класифікації об'єкта *EnsembleImageClassifier* проводиться розрахунок дескрипторів КТ вхідних зображень, їх фільтрація (за необхідності) та повертається необхідна для аналізу інформація: мітка класу зображення, розподіл голосів між еталонами, помилка квантування.

3.3 Аналіз результатів тестування

Для перевірки точності класифікації кожного методу буде застосовано тестову вибірку зображень, яка складається із еталонних зображень та таких, що зазнають наступний перелік афінних перетворень у різних комбінаціях (рис. 3.4):

- поворот проти годинникової стрілки на 30, 45, 60, 90, 135, 180, 270 градусів;
- масштабування зображення до 50% та 75% від оригінального розміру;
- зміщення по горизонталі вліво та вправо на 100 пікселів.

Загальна кількість тестових зображень становить 360 екземплярів. Оцінювання точності буде здійснено за показником precision (2.10). Клас зображення буде визначатись за відповідністю максимальної частки голосів до правильного класу.

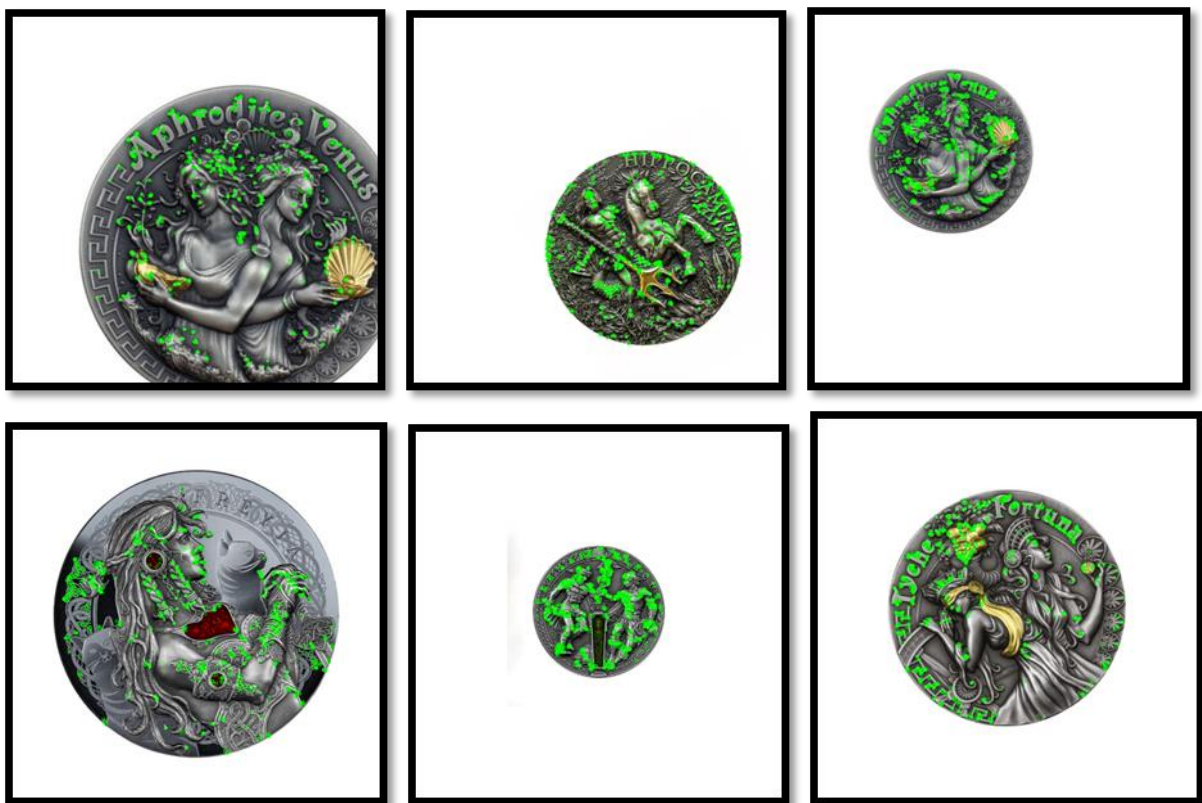


Рисунок 3.4 – Приклад зображень вибірки для тестування із позначеними КТ

Для початку порівняємо середню точність класифікації тестової вибірки із застосуванням запропонованих трьох методів: одинична мережа Кохонена, ансамбль мереж з 1 центром та ансамбль мереж із довільною кількістю центрів.

Тестування буде проводитись із врахуванням впливу кількості епох навчання та початкового коефіцієнту навчання на результуючу ефективність моделі. Кількість епох буде варіюватись між 5 та 25, а коефіцієнт навчання між 0,1 та 0,9. На період даного експерименту кількість дескрипторів в описі еталонів для тренування та в описі вхідних (тестових) зображень будемо вважати зафіксованою на $s=500$. Жирним виділено три найбільш ефективні конфігурації навчання за критерієм точності класифікації на тестовому наборі. Результати експериментів наведену у таблицях 3.1-3.3.

Таблиця 3.1 – Залежність точності класифікації мережі Кохонена з 1 нейроном при різних конфігураціях навчання

Число епох	5	10	25	50
Коефіцієнт навчання				
0,1	0,40	0,002	0,3	0,2
0,25	0,13	0,15	0	0,119
0,5	0,01	0,12	0	0,002
0,75	0	0,2	0,39	0
0,9	0,036	0,002	0	0,175

Таблиця 3.2 – Залежність точності класифікації комітету мереж Кохонена з 1 нейроном на еталон при різних конфігураціях навчання

Число епох	5	10	25	50
Коефіцієнт навчання				
1	2	3	4	5
0,1	0,791	0,758	0,65	0,641
0,25	0,719	0,8	0,758	0,658
0,5	0,60	0,719	0,791	0,758

Продовження таблиці 3.2

1	2	3	4	5
0,75	0,591	0,644	0,8	0,80
0,9	0,586	0,616	0,792	0,79

Таблиця 3.3 – Залежність точності класифікації комітету мереж Кохонена з 3 нейронами на еталон при різних конфігураціях навчання

Число епох	5	10	25	50
Коефіцієнт навчання				
0,1	0,769	0,80	0,811	0,788
0,25	0,636	0,658	0,802	0,755
0,5	0,580	0,627	0,888	0,802
0,75	0,619	0,66	0,788	0,75
0,9	0,630	0,602	0,683	0,722

Як видно з таблиці 3.1 та 3.2, навіть при встановленні лише одного вектора-представника для класу застосування комітету нейронних мереж збільшує середнє значення точності (2.10) з 0,11 для першого методу до 0,713 для другого. Натомість при застосуванні більшого числа нейронів 3, 5, 10 і 25 на еталон середня точність зростає до рівня 0,718, 0,665, 0,76 та 0,908 відповідно за результатами табл. 3.3, А.1-А.3.

Також розглянемо розподіл голосів при класифікації еталонних зображень для одиничної мережі Кохонена, та ансамблей з 1 та 3 нейронами на еталон (табл. 3.4-3.6). Для моделі одиничної мережі Кохонена буде обрана конфігурація за максимальною точністю з таблиці 3.1 (кількість дескрипторів на еталон – 500, кількість епох навчання – 5, коефіцієнт навчання – 0,1), для ансамблевих моделей кількість епох навчання буде однаково рівною 10, а коефіцієнт навчання – 0,25.

Таблиця 3.4 – Розподіл голосів при класифікації еталонних зображень
одиночною мережею Кохонена

Голоса	Эталон 1	Эталон 2	Эталон 3	Эталон 4	Эталон 5
Эталон 1	0,293	0,204	0,159	0,238	0,106
Эталон 2	0,224	0,157	0,169	0,23	0,22
Эталон 3	0,263	0,171	0,122	0,311	0,133
Эталон 4	0,303	0,152	0,119	0,259	0,167
Эталон 5	0,222	0,151	0,177	0,173	0,277

Таблиця 3.5 – Розподіл голосів при класифікації еталонних зображень
моделлю комітету мереж Кохонена з 1 нейроном на еталон

Голоса	Эталон 1	Эталон 2	Эталон 3	Эталон 4	Эталон 5
Эталон 1	0,299	0,136	0,223	0,178	0,164
Эталон 2	0,149	0,207	0,233	0,216	0,195
Эталон 3	0,144	0,126	0,336	0,215	0,179
Эталон 4	0,124	0,123	0,216	0,363	0,174
Эталон 5	0,131	0,108	0,182	0,201	0,378

Таблиця 3.6 – Розподіл голосів при класифікації еталонних зображень
моделлю комітету мереж Кохонена з 3 нейронами на еталон

Голоса	Эталон 1	Эталон 2	Эталон 3	Эталон 4	Эталон 5
Эталон 1	0,383	0,182	0,141	0,196	0,098
Эталон 2	0,134	0,441	0,117	0,187	0,121
Эталон 3	0,184	0,218	0,269	0,21	0,119
Эталон 4	0,157	0,201	0,154	0,367	0,121
Эталон 5	0,193	0,179	0,139	0,22	0,269

По даним таблиць 3.4-3.6 видно, що застосування ансамблю нейронних мереж, починаючи з 3 представників на клас, призводить до правильної класифікації всіх еталонних зображень.

Розглянемо вплив збільшення кількості нейронів на еталон та особливості конфігурації навчання різних комітетів. Для цього побудуємо графіки залежності точності класифікації моделей різної конфігурації навчання від кількості епох навчання: 5 (рис. 3.5), 10 (рис. 3.6), 25 (рис. 3.7). Більш насиченим кольором виділено клас моделей із найбільшою середньою точністю.

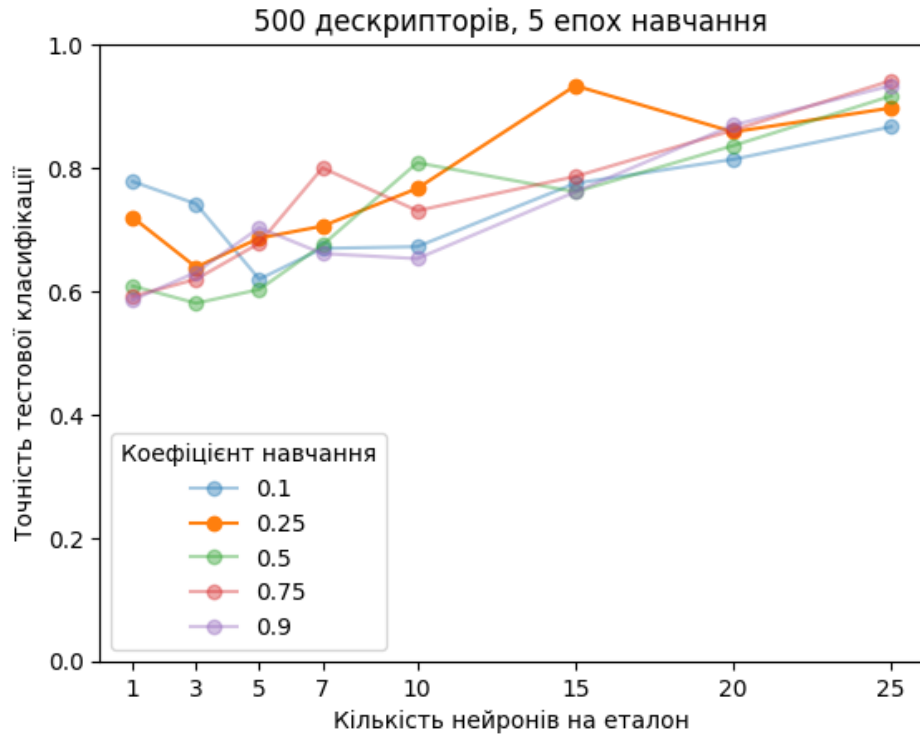


Рисунок 3.5 – Графік точності моделі комітету НМ при 5 епохах навчання

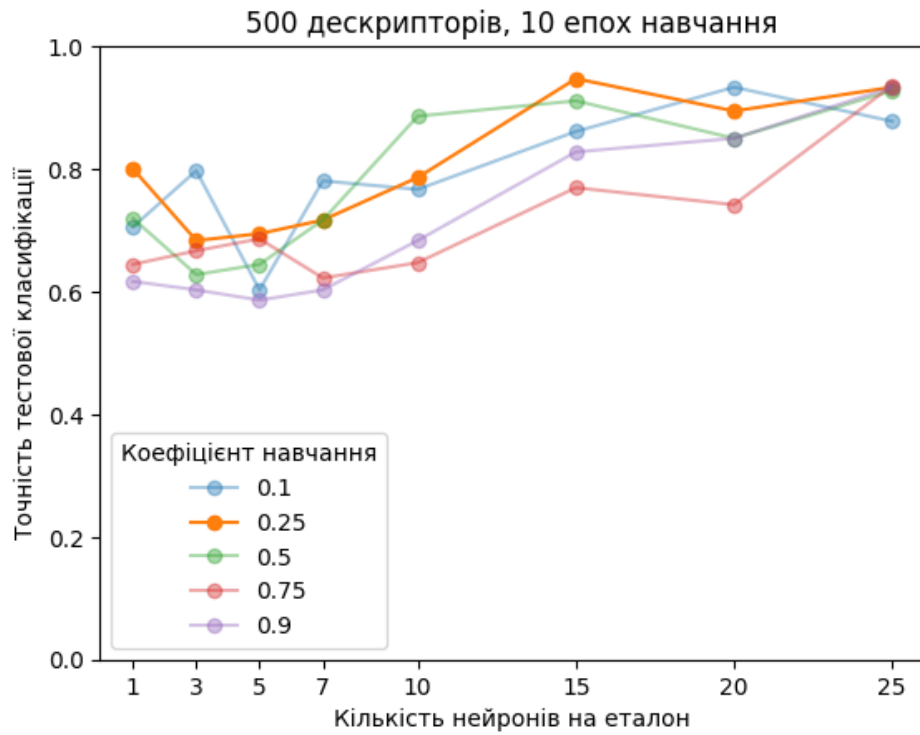


Рисунок 3.6 – Графік точності моделі комітету НМ при 10 епохах навчання

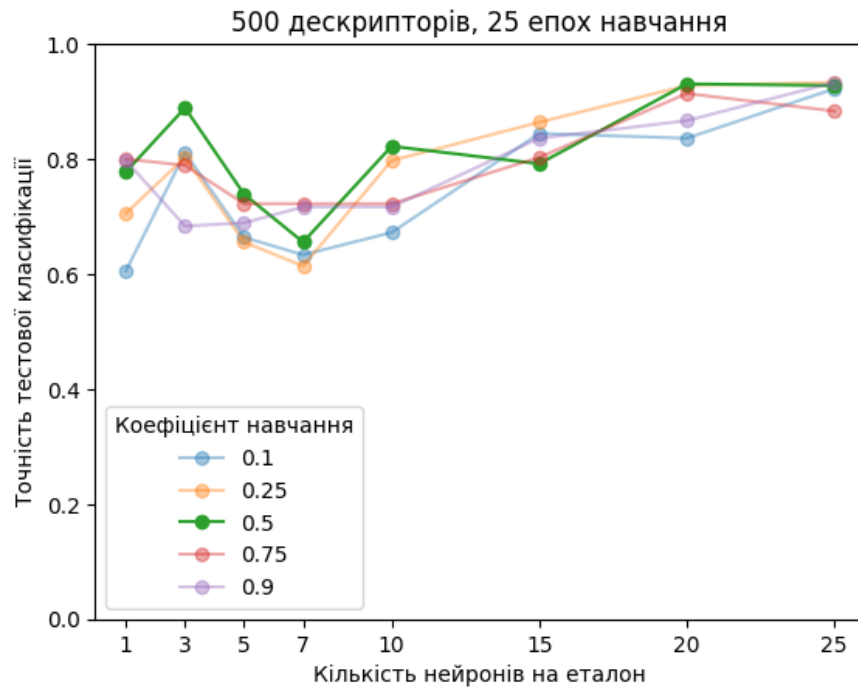


Рисунок 3.7 – Графік точності моделі комітету НМ при 25 епохах навчання

Як видно з графіків на рисунках 3.5-3.7, при збільшенні числа епох для невеликої кількості нейронів на еталон (1-10) більш продуктивною є встановлення високого початкового коефіцієнту навчання (0,5-0,9), тоді як для кількості нейронів більше 10 зі збільшенням числа епох більш ефективно встановлення невеликого початкового коефіцієнту навчання (0,1-0,25).

Можна прослідити, що збільшення кількості нейронів на еталон призводить до покращення точності класифікації (рис. 3.8). Це справедливо як для конфігурацій ансамблю з великою кількістю епох та низьким початковим коефіцієнтом навчання (ретельна конфігурація), так і для низької кількості епох при великому початковому значенні коефіцієнту навчання (швидка конфігурація).

Разом зі зростанням числа нейронів також зростає час на тренування моделі та виконання класифікації, але нелінійним чином. На рисунку 3.9 видно, що до досягнення числа нейронів 5, час на тестування різко знижується, при чому час на навчання мінімальний лише при саме 5 центроїдах, одночасно із мінімальною точністю (рис. 3.8). Після чого разом із перевищенням числа нейронів 5 представників на клас час на тренування

та класифікації монотонно зростає, разом із точністю. Повторні тести підтверджують нелінійну природу цієї залежності. Класифікація проводиться по всій тестовій вибірці з 360 зображень, отриманих афінними перетвореннями еталонів.

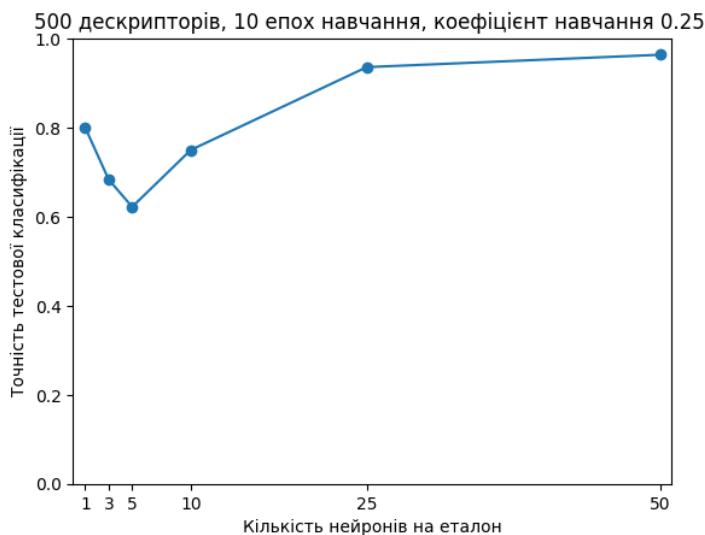


Рисунок 3.8 – Графік залежності точності класифікації від кількості нейронів на еталон

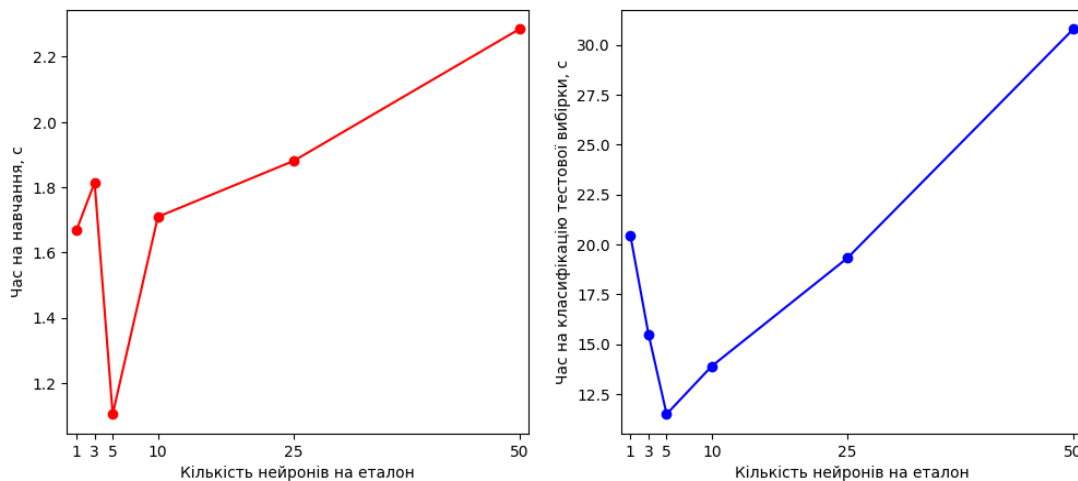


Рисунок 3.9 – Графік зміни часу на навчання та тестування при збільшенні числа нейронів на клас

У розділі 3.2 було розглянуто можливість застосування фільтрації набору дескрипторів за значенням показнику інформативності. Розглянемо

вплив застосування процедури фільтрації на точність класифікації тестової вибірки зображень. Для порівняння візьмемо з рисунку 3.7 дві найкращі моделі серед тих, що мають маленьку кількість нейронів та велику.

Перша модель складається з 3 нейронів на еталон, друга – з 25 нейронів на еталон. Обидві моделі навчаються 25 епох при початковому коефіцієнті навчання 0,5 (найкраща конфігурація серед графіків на рисунках 3.4-3.7). Дані експерименту описані в таблиці 3.7.

Таблиця 3.7. Вплив фільтрації набору дескрипторів навчальної вибірки на точність класифікації

Кількість нейронів	3			25		
	Початкова кількість дескрипторів					
Фільтрована кількість дескрипторів	1000	2500	5000	1000	2500	5000
Без фільтрації	0,813	0,605	0,79	0,983	0,95	0,961
250	0,769	0,777	0,633	0,902	0,875	0,822
500	0,680	0,622	0,6	0,933	0,983	0,833

На прикладі двох еталонів позначимо вплив фільтрації на склад ключових точок. На рисунку 3.10 а) та 3.11 а) зображено четвертий та п'ятий еталон із виділеними 500 ключовими точками без фільтрації. На рисунку 3.10 б) та 3.11 б) зображено ті ж еталони при розрахунку більшої кількості дескрипторів – 2500. Алгоритм ORB не визначає сам по собі, які з КТ є більш, а які менш значущими, а тому їх розподіл можна вважати випадковим. Отже, щоб отримати найбільш інформативний і стиснутий набір дескрипторів КТ, проведемо процедуру фільтрації, результат якої позначено на рисунках 3.10 в) та 3.11 в).

Аналізуючи дані з таблиці 3.7 бачимо, що застосування процедури фільтрації є більш вигідним у випадку 25 представників на клас, тоді як при 3 представниках на клас точність моделі зменшується. Для порівняння,

моделі з 3 та 25 нейронами на еталон, які навчались на 500 дескрипторах при тій же конфігурації без застосування попередньої фільтрації, мали середню точність класифікації тестової вибірки 0,888 та 0,933 відповідно. Тобто у випадку скорочення великого набору дескрипторів задля забезпечення пришвидшення тренування та класифікації є сенс застосовувати фільтрацію суміжно зі збільшенням числа нейронів-представників.



Рисунок 3.10 – Застосування фільтрації до набору дескрипторів четвертого еталону: а) 500 дескрипторів без фільтрації; б) 2500 дескрипторів без фільтрації; в) 500 дескрипторів після фільтрації



Рисунок 3.11 – Застосування фільтрації до набору дескрипторів п'ятого еталону: а) 500 дескрипторів без фільтрації; б) 2500 дескрипторів без фільтрації; в) 500 дескрипторів після фільтрації

Протестуємо також потоковий варіант навчання комітету, згідно з коефіцієнтом навчання (2.4). При такому способі навчання зразки поступають на розподіл до найближчих нейронів, допоки різниця помилки квантування (2.9) не досягне встановленого порогу. Недолік такого підходу може бути в тому, що різниця може бути досягнута на самих ранніх етапах навчання, через що модель залишиться недонавченою (underfitted). Задля запобігання такої ситуації можна ввести короткий процес попереднього ітеративного навчання на невеликій кількості епох, після чого продовжити потокове навчання.

У таблиці 3.8 зведено дані про результат класифікації після стандартного потокового навчання різних моделей при різних коефіцієнтах навчання, у таблиці 3.9 зведено дані про точність класифікації при додатковому етапі попереднього ітеративного навчання. Дослід проводився на обсязі у 500 дескрипторів на зображення (еталонне або тестове), 5 епохах попереднього навчання та порозі різниці помилки векторного квантування рівній 0.0001.

Таблиця 3.8 – Точність класифікації моделі комітету НМ з різною кількістю нейронів на еталон при потоковому навчанні

Число нейронів	1	3	5	10	25	50
Коефіцієнт навчання						
0,1	0,386	0,2	0,285	0,5	0,47	0,92
0,25	0,23	0,208	0,53	0,4	0,67	0,84
0,5	0,29	0,225	0,2	0,38	0,55	0,83
0,75	0,23	0,255	0,19	0,46	0,55	0,82
0,9	0,263	0,44	0,39	0,30	0,59	0,84
Середня точність	0,279	0,265	0,319	0,408	0,566	0,85

Таблиця 3.9 – Точність класифікації моделі комітету НМ з різною кількістю нейронів на еталон при потоковому навчанні з ітеративним попереднім навчанням

Число нейронів	1	3	5	10	25	50
Коефіцієнт навчання						
0,1	0,43	0,613	0,808	0,78	0,95	0,933
0,25	0,202	0,761	0,736	0,77	0,90	0,936
0,5	0,2	0,394	0,713	0,73	0,84	0,914
0,75	0,213	0,202	0,436	0,74	0,86	0,933
0,9	0,2	0,252	0,505	0,69	0,875	0,927
Середня точність	0,249	0,444	0,639	0,742	0,885	0,928

З результатів експерименту видно, що впровадження попереднього навчання покращує класифікацію тестового набору після потокового-зацикленого навчання класифікатора, але загалом середня точність класифікації після потокового навчання програє строго ітеративному.

ВИСНОВКИ

У рамках кваліфікаційної роботи проведено дослідження методу класифікації зображень за допомогою моделі комітету нейронних мереж для векторного квантування.

Здійснено дослідження комітетів штучних нейронних мереж різної конфігурації та оцінено ефективність роботи кожної із них. Результати дослідження показали ефективність застосування ансамблевої моделі проти одиничної мережі Кохонена.

Оцінено залежність ефективності роботи комітету від збільшення кількості нейронів-представників на клас, залежність ефективності навчання від числа дескрипторів опису зображення, епох навчання та початкових коефіцієнтів навчання. Також визначено перевагу застосування ітеративного методу навчання базових класифікаторів ансамблю замість потокового.

Новизною кваліфікаційної роботи є результат детального прикладного дослідження методу класифікації з ансамблем мереж Кохонена у просторі дескрипторів ключових точок.

Практична значущість дослідження полягає у порівняльному аналізі методів класифікації на спільній базі даних; розвитку та просуванні тематики застосування ансамблевих моделей неглибоких нейронних мереж в задачі класифікації зображень; розробленні програмних моделей для впровадження запропонованих методів у системах комп'ютерного зору.

Перспектива розвитку може бути пов'язана із вивченням застосування альтернативних засобів попередньої обробки зображень для застосування спільно з ансамблевими моделями.

Результати дослідження апробовано у тезах доповідей на Міжнародній науково-практичній конференції «Distance learning in universities and modern problems» [53].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), 25–36.
2. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, *International Journal of Academic and Applied Research*, 7(11), pp. 134-145.
3. Gorokhovatskyi, O., Peredrii, O., Gorokhovatskyi, V., Vlasenko, N. (2023) Explanation of CNN Image Classifiers with Hiding Parts. In: J. Benois-Pineau, R. Bourqui, D. Petkovic, G. Quenot (eds), *Explainable Deep Learning Artificial Intelligence*, pp. 125-146, Academic Press, 346 p.
4. Hu, Zh., Bodyanskiy, Ye., Tyshchenko, O. A deep cascade neural network based on extended neo-fuzzy neurons and its adaptive learning algorithm. In: *IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, Kyiv, Ukraine, 29 May–2 June, 2017, 801–805.
5. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
6. Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556.
7. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., & Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1-9).
8. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 770-778).

9. Wang, P., Fan, E., & Wang, P. (2021). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters*, *141*, 61-67.
10. Тимчишин, Р. М., Волков, О. Є., Господарчук, О. Ю., & Богачук, Ю. П. (2018). Сучасні підходи до розв'язання задач комп'ютерного зору. *Управляючі системи і машини*.
11. Mohammed, A., & Kora, R. (2022). An effective ensemble deep learning framework for text classification. *Journal of King Saud University-Computer and Information Sciences*, *34*(10), 8825-8837.
12. Goel, P., Jain, R., Nayyar, A., Singhal, S., & Srivastava, M. (2022). Sarcasm detection using deep learning and ensemble learning. *Multimedia Tools and Applications*, *81*(30), 43229-43252.
13. Zhou, J., Jiang, Z., Chung, F. L., & Wang, S. (2020). Formulating ensemble learning of SVMs into a single SVM formulation by negative agreement learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *51*(10), 6015-6028.
14. Mishra, S., Shaw, K., Mishra, D., Patil, S., Kotecha, K., Kumar, S., & Bajaj, S. (2022). Improving the accuracy of ensemble machine learning classification models using a novel bit-fusion algorithm for healthcare AI systems. *Frontiers in Public Health*, *10*, 858282.
15. Sun, Y., Li, Z., Li, X., & Zhang, J. (2021). Classifier selection and ensemble model for multi-class imbalance learning in education grants prediction. *Applied Artificial Intelligence*, *35*(4), 290-303.
16. Sammut, C., & Webb, G. I. (Eds.). (2011). *Encyclopedia of machine learning*. Springer Science & Business Media.
17. Doroudi, S. (2020). The bias-variance tradeoff: How data science can inform educational debates. *AERA open*, *6*(4), 2332858420977208.
18. Breiman, L. (1998). Arcing classifier (with discussion and a rejoinder by the author). *The annals of statistics*, *26*(3), 801-849.

19. Alelyani, S. (2021). Stable bagging feature selection on medical data. *Journal of Big Data*, 8(1), 1-18.
20. Liu, H., Gegov, A., Cocea, M., Liu, H., Gegov, A., & Cocea, M. (2016). Ensemble learning approaches. *Rule based systems for big data: a machine learning approach*, 63-73.
21. Ravichandran, T., Gavahi, K., Ponnambalam, K., Burtea, V., & Mousavi, S. J. (2021). Ensemble-based machine learning approach for improved leak detection in water mains. *Journal of Hydroinformatics*, 23(2), 307-323.
22. Sabzevari, M., Martínez-Muñoz, G., & Suárez, A. (2022). Building heterogeneous ensembles by pooling homogeneous ensembles. *International Journal of Machine Learning and Cybernetics*, 1-8.
23. Bian, Y., Wang, Y., Yao, Y., & Chen, H. (2019). Ensemble pruning based on objection maximization with a general distributed framework. *IEEE transactions on neural networks and learning systems*, 31(9), 3766-3774.
24. Li, Y., & Chen, W. (2020). A comparative performance assessment of ensemble learning for credit scoring. *Mathematics*, 8(10), 1756.
25. Breiman, L. (2001). Random forests. *Machine learning*, 45, 5-32.
26. Schapire, R. E., & Freund, Y. (2013). Boosting: Foundations and algorithms. *Kybernetes*, 42(1), 164-166.
27. Veit, A., Wilber, M. J., & Belongie, S. (2016). Residual networks behave like ensembles of relatively shallow networks. *Advances in neural information processing systems*, 29. [Online]. Available: <https://arxiv.org/abs/1605.06431v2>. [Accessed 6 November 2018].
28. Abdi, M., & Nahavandi, S. (2016). Multi-residual networks: Improving the speed and accuracy of residual networks. *arXiv preprint arXiv:1609.05672*.
29. Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.
30. Williams, T., & Li, R. (2018). An ensemble of convolutional neural networks using wavelets for image classification. *Journal of Software Engineering and Applications*, 11(2), 69-88.

31. Mo, W., Luo, X., Zhong, Y., & Jiang, W. (2019, June). Image recognition using convolutional neural network combined with ensemble learning algorithm. In *Journal of Physics: Conference Series* (Vol. 1237, No. 2, p. 022026). IOP Publishing.

32. Gorokhovatskyi, O., & Peredrii, O. (2019). АНСАМБЛЬ ДРІБНИХ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ КЛАСИФІКАЦІЇ СТАТИ ЛЮДИНИ У ВІДЕОПОТОЦІ. *Advanced Information Systems*, 3(4), 74-79.

33. Красношлик, Н. О., & Сердюк, М. О. (2019). Застосування ансамблів нейронних мереж для розв'язання задачі класифікації зображень. *Вісник Черкаського університету. Серія Прикладна математика. Інформатика*, (1), 53-60.

34. Perez, F., Avila, S., & Valle, E. (2019). Solo or ensemble? choosing a cnn architecture for melanoma classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

35. Polat, K., & Кос, К. О. (2020). Detection of skin diseases from dermoscopy image using the combination of convolutional neural network and one-versus-all. *Journal of Artificial Intelligence and Systems*, 2(1), 80-97.

36. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, 11, pp. 126938-126949.

37. Gadetska S., Gorokhovatskyi V., Stiahlyk N., Vlasenko N. (2022) Aggregate Parametric Representation of Image Structural Description in Statistical Classification Methods. In *CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2022)*, 3137, pp. 68-77.

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

39. Gorokhovatskyi V., Gadetska S., Stiahlyk N. (2020) Image structural classification technologies based on statistical analysis of descriptions in the form of bit descriptor set. In *CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2020)*. 2608. pp. 1027-1039.

40. Клінов В. О. Моделювання методу класифікації зображень із використанням засобів самонавчання : пояснювальна записка до атестаційної роботи здобувача вищої освіти на першому (бакалаврському) рівні, спеціальність 122 Комп'ютерні науки / В. О. Клінов ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2022. – 42 с.

41. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. International scientific symposium «Intelligent Solutions-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine, pp. 25-27.

42. Gorokhovatskyi V., Tvoroshenko I. (2020) Image Classification Based on the Kohonen Network and the Data Space Modification. In CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2020). 2608. pp. 1013-1026.

43. Гороховатський В., Творошенко І., Сидоренко Д. (2021) Класифікація зображень із використанням кластерного подання, Міжн. наук. симпозіум Інтелектуальні рішення-С. Обчислювальний інтелект. Теорія прийняття рішень: праці міжн. наук. симп. (Вересень 29, 2021). Київ – Ужгород, 44-45.

44. Gadetska, S.V., Gorokhovatskyi, V. O., Stiahlyk, N. I., Vlasenko, N.V. 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, 2021, №4, pp. 58-68.

45. M. A. Ahmad, V. Gorokhovatskyi, I. Tvoroshenko, N. Vlasenko, S. K. Mustafa (2021) The Research of Image Classification Methods Based on the Introducing Cluster Representation Parameters for the Structural Description, International Journal of Engineering Trends and Technology, 69(10), pp. 186-192

46. Руденко, О. Г., & Бодяньський, Є. В. (2006). Штучні нейронні мережі. Харків: Компанія СМІТ.

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

48. Gorokhovatskyi, V., Gadetska, S., & Ponomarenko, R. (2018). СТАТИСТИЧНІ РОЗПОДІЛИ ТА ЛАНЦЮЖКОВЕ ПОДАННЯ ДАНИХ ПРИ ВИЗНАЧЕННІ РЕЛЕВАНТНОСТІ СТРУКТУРНИХ ОПИСІВ ВІЗУАЛЬНИХ ОБ'ЄКТІВ. *Системи управління, навігації та зв'язку. Збірник наукових праць*, 6(52), 87-92.

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

50. OpenCV. URL: <https://docs.opencv.org/> (дата звернення 01.12.2023).

51. OpenCV Compare Images, URL: <https://www.delftstack.com/howto/python/opencv-compare-images/> (дата звернення 01.12.2023).

52. Посилання на сайт зображень монет. URL: <https://fama.ua/podarki/monety> (дата звернення 01.12.2023).

53. Клінов В. (2023) Класифікація зображень із використанням комітету нейронних мереж, *Abstracts of VIII International Scientific and Practical Conference «Distance learning in universities and modern problems»*, (November 07 – 10, 2023). *Budapest, Hungary*, pp. 284-288.