Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

# КВАЛІФІКАЦІЙНА РОБОТА

## Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Модель штучної імунної мережі для обробки даних ігрових персонажів

(тема)

Виконав:

студент II курсу, групи СПм-22-3

Стефанюк Я.Г.
(прізвище, ініціали)

Спеціальність

123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: ст. викл. Фомічов О.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ _____ Коваленко А.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____"_____ 20___ р.

# ЗАВДАННЯ

## НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Стефанюку Ярославу Геннадійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Модель штучної імунної мережі для обробки даних ігрових персонажів

затверджена наказом по університету від "01" квітня 2024 р. № 257 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 15 червня 2024 р.

3. Вхідні дані до роботи
штучна імунна мережа
обробка даних
архівація даних
ігровий персонаж

4. Перелік питань, що потрібно опрацювати у роботі
Загальні теоретичні відомості
Методи архівації даних на прикладі стиснення зображень для ігрових персонажів
Програмна реалізація

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 15 слайдів

6. Консультанти розділів роботи (заповнюється 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**

| № | Назва етапів роботи | Термін виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Отримання завдання та аналіз літератури | 01.04.2024 – 06.04.2024 | |
| 2 | Огляд існуючих рішень та методів | 07.04.2024 – 12.04.2024 | |
| 3 | Розробка моделі | 13.04.2024 – 18.04.2024 | |
| 4 | Вибір програмних засобів | 19.04.2024 – 25.04.2024 | |
| 5 | Програмна реалізація | 26.04.2024 – 02.05.2024 | |
| 6 | Аналіз отриманих результатів | 03.05.2024 – 16.05.2024 | |
| 7 | Оформлення записки | 17.05.2024 – 12.06.2024 | |
| 8 | Представлення роботи в ЕК | 15.06.2024 | |
| | | | |
| | | | |
| | | | |
| | | | |

Дата видачі завдання 01 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ ст. викл. Фомічов О.О.
(підпис) (посада, прізвище, ініціали)

# РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 65 с., 16 рис., 1 табл., 2 дод., 14 джерел.

ШТУЧНА ІМУННА СИСТЕМА, СТИСНЕННЯ ДАНИХ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, АРХІВАЦІЯ, ІГРОВИЙ ПЕРСОНАЖ.

Метою кваліфікаційної роботи є дослідження моделей та методів для архівації даних з використанням штучних нейронних та імунних мереж.

У ході виконання кваліфікаційної роботи розглянуто методи обробки та архівації даних ігрових персонажів. Розглянуто існуючі види штучних нейронних та імунних систем. Запропоновано модель штучної імунної мережі для архівації даних. Представлені підходи та процедури розв’язання завдання архівації даних з використанням штучних імунних систем та штучних нейронних мереж типу радіально-базисних функцій. Обробка та архівація даних ігрових персонажів з використанням штучних імунних мереж – це передова технологія, яка застосовується для забезпечення безпеки та збереження важливої інформації в ігрових системах.

ABSTRACT

Master’s thesis: 65 pages, 16 figures, 1 tables, 2 appendices, 14 sources.

ARTIFICIAL IMMUNE SYSTEM, DATA COMPRESSION, SOFTWARE, ARCHIVING, GAME CHARACTER.

The major goal of this thesis is to research models and methods for data archiving using artificial neural and immune networks.

In order to the qualification work methods of data processing and archiving of game characters were considered. Existing types of artificial neural and immune systems are considered. An artificial immune network model for data archiving is proposed. Approaches and procedures for solving the task of data archiving using artificial immune systems and artificial neural networks of the type of radial-basis functions are presented. The processing and archiving of data of game characters using artificial immune networks is an advanced technology that is used to ensure the safety and preservation of important information in game systems.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ ..... 7
ВСТУП ..... 8
1 ЗАГАЛЬНА ТЕОРІЯ ..... 10
1.1 Біологічні основи штучних імунних систем ..... 10
1.2 Аналіз ШНМ ..... 13
1.3 Обчислювальні моделі ШНМ ..... 16
1.3.1 Модель негативного відбору ..... 16
1.3.2 Модель позитивного відбору ..... 17
1.3.3 Модель клонального відбору ..... 18
1.3.4 Модель мережевої взаємодії ..... 19
1.4 Програмне забезпечення для роботи з ШНМ ..... 21
2 МЕТОДИ АРХІВАЦІЇ ДАНИХ НА ПРИКЛАДІ СТИСНЕННЯ ЗОБРАЖЕНЬ ДЛЯ ІГРОВИХ ПЕРСОНАЖІВ ..... 24
2.1 Класифікація методів кодування зображень ..... 35
3 РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ ..... 45
ВИСНОВКИ ..... 52
ДОДАТОК А Графічний матеріал кваліфікаційної роботи ..... 55
ДОДАТОК Б Апробація ..... 64

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

ОС – операційна система

ПЗ – програмні засоби

ШІС – штучна імунна система

ШНМ – штучна нейронна мережа

# ВСТУП

З розвитком технологій комп'ютерні ігри стають все більш складними, а кількість даних, що обробляються у цих іграх, стрімко зростає. Однією з важливих задач є забезпечення цілісності та безпеки даних ігрових персонажів. Для досягнення цієї мети можуть використовуватися різноманітні підходи, одним з яких є модель штучної імунної мережі. Перспектива використання ШІМ для обробки даних ігрових персонажів є надзвичайно обнадійливою. Завдяки постійному розвитку технологій, такі системи можуть стати ключовим елементом забезпечення безпеки та надійності в сучасних комп'ютерних іграх.

В ШНМ розглядаються різні механізми імунітету і їх відношення до обробки інформації і вирішення завдань. У порівнянні з іншими моделями і методами, створеними на основі біологічних уявлень, такими як нейронні мережі та еволюційні алгоритми, штучні імунні системи до теперішнього часу залучали дуже мало уваги. Імунна система представляє великий інтерес як система, здатна ефективно обробляти значні обсяги даних. Зокрема, вона виконує великий обсяг складних високопараллельних розподілених обчислень [1].

Поведінка імунної системи в цілому визначається всією сукупністю локальних взаємодій. Відповідні методи з успіхом застосовуються для вирішення завдань розпізнавання образів, діагностики та виявлення несправностей, а також для створення систем комп'ютерної безпеки і в деяких інших додатках. В ШНМ використовуються обчислювальні моделі, засновані на принципі функціонування ШНМ, такі як модель клонального відбору, мережевої взаємодії, негативного відбору та ін.

Модель негативного відбору є однією з найбільш широко відомих моделей ШНМ. Її принцип роботи полягає в генерації детекторів, відповідних аномальної поведінки досліджуваного об'єкта. Метод

негативного відбору складається з двох фаз: навчання та класифікації. Огляд існуючого програмного забезпечення показує наявність кількох окремих розрізнених програм по роботі з ШНМ, тому актуальною є задача розробки програмних засобів реалізації ШНМ для виявлення аномалій у вихідних даних на основі обчислювальної моделі негативного відбору.

Метою кваліфікаційної роботи є дослідження моделей та методів для архівації даних з використанням штучних нейронних та імунних мереж.

Завдання:

- аналіз принципів та методів побудови штучних імунних і гібридних систем інтелектуального аналізу даних;
- огляд існуючих методів архівації графічних даних ігрових персонажів;
- розробка моделі і засобів структурно-параметричного синтезу нейронних мереж для обробки та архівації даних.

# 1 ЗАГАЛЬНА ТЕОРІЯ

## 1.1 Біологічні основи штучних імунних систем

Природна імунна система являє собою складну систему, що складається з нескольких функціонально різних компонентів. Імунна система використовує багаторівневий захист проти зовнішніх антигенів, включаючи дію неспецифічних (вроджених) і специфічних (придбаних) захисних механізмів. В процесі еволюції імунна система очучается розрізняти зовнішні антигени (бактерії і віруси) і совственние клітини або молекули організму. Встановлено, що в організмі людини підтримується велика кількість імунокомпетентних клітин, які циркулюють по всьому тілу.

Основним типом клітин, що беруть участь в імунній відповіді і володіють властивостями специфічності, різноманітності, пам'яті і адаптивності, є лімфоцити. В організмі є два основних типи лімфоцитів - Т- (рисунок 1.1) і В-лімфоцити (рисунок 1.2). Основною їх функцією є захоплення антигенів і забезпечення взаємодії з ними лімфоцитів для стимуляції імунної відповіді.

Зазначені два типи лімфоцитів грають в імунній відповіді різну роль, хоча і можуть взаємодіяти, контролюючи таким чином свої функції. При попаданні антигену в організм лише мала частина клітин імунної системи здатна до розпізнавання його пептидів. Таке розпізнавання стимулює процеси розмноження і диференціювання лімфоцитів, що призводять до утворення клонів ідентичних клітин (або антитіл).

Цей процес, званий розмноженням клону формує численну популяцію специфічних до антигену антителопродуцирующих клітин. Розмноження клону імунокомпетентних клітин призводить до руйнування або нейтралізації антигену. Антиген - це ініціатор і рушійна сила всіх реакцій набутого імунітету. Імунна система виникла для розпізнавання і руйнування

чужорідних антигенів, а також усунення джерела їх утворення - бактерій, інфікованих вірусом клітин і т.п. Коли антиген елімінувати, імунна відповідь припиняється. Частина утворилися клітин зберігається для імунної пам'яті. В результаті, наступне вплив схожого антигену призводить до більш швидкої імунної реакції (вторинного відповіді). Реакція В-лімфоцита на антиген представлена на рисунку 1.1.

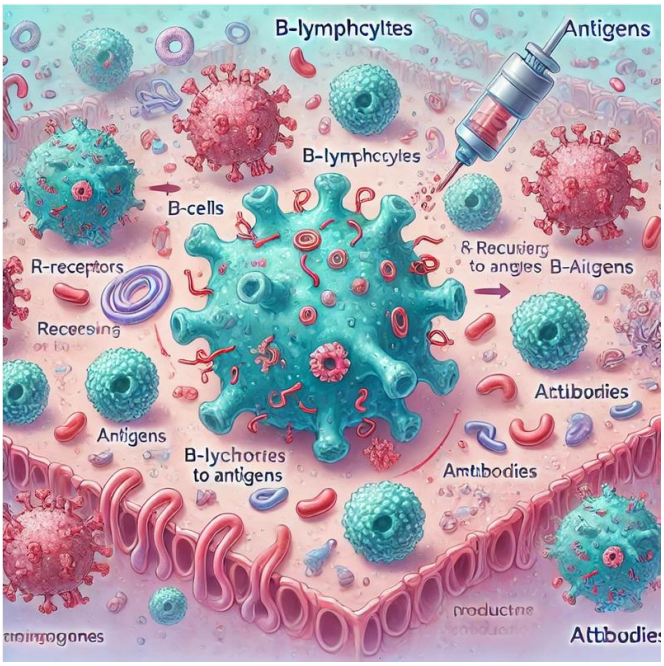


Рисунок 1.1 – Взаємодія В-лімфоцитів та антигенів

Процес циркуляції В- і Т-лімфоцитів в первинних і вторинних лімфоїдних органах ретельно контролюється, що забезпечує потрапляння відповідних клітинних популяцій -необученних клітин, а також клітин пам'яті - в різні місця призначення. Клітини пам'яті роблять виборчу перевагу тому типу тканин, в якому вони вперше зустрілися з антигеном. Ймовірно, це забезпечує повернення кожної конкретної клітини пам'яті в ту ділянку тіла, де вона, швидше за все, знову зустрінеться з антигеном. Реакція Т-лімфоцити на патоген представлена на рисунку 1.2.

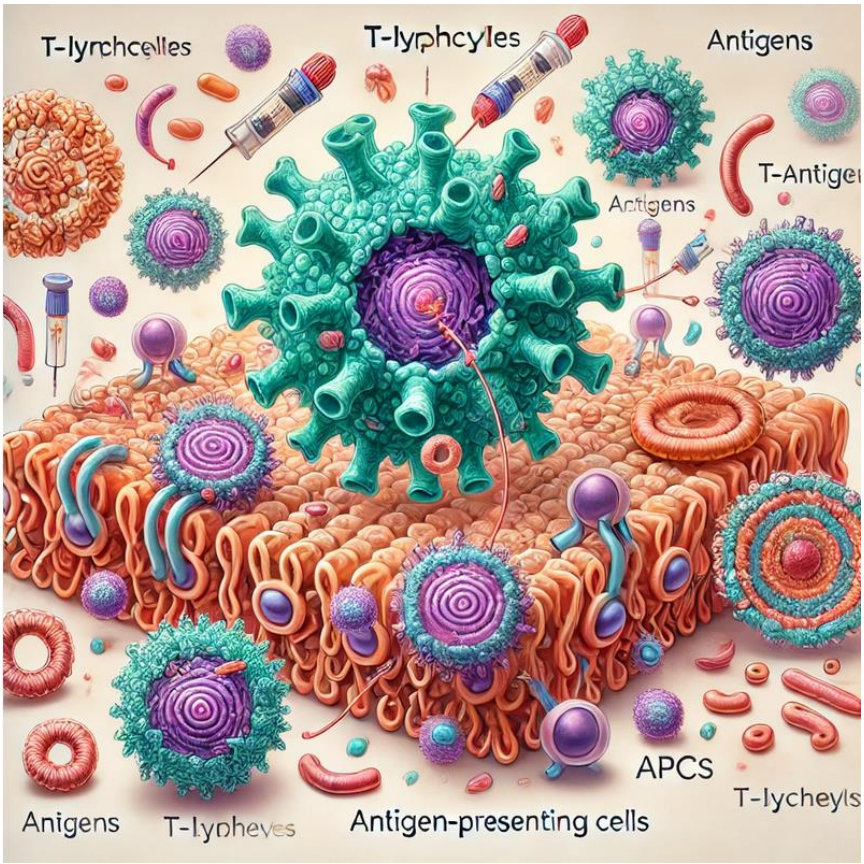


Рисунок 1.2 – Взаємодія Т-лімфоцитів і антигенів

Існує два основні варіанти імунної відповіді: гуморальний, опосередкований В-клітинами і їх продуктами, і клітинний, в якому беруть участь Т-клітини. Обома варіантами імунних реакцій відповідають подібні послідовності етапів захисту організму: активація, розподіл, діфференіровка, секреція, імунна атака, супрессия і пам'ять, проте вони осуществяется різними способами.

У регуляції як гумарального, так і клітинного імунітету беруть участь популяції Т-клітин, які називаються хелперно або супресорних клітинами, які посилюють або, відповідно, пригнічують імунну відповідь. Таким чином, основна роль імунної системи полягає у розпізнаванні всіх клітин організму і класифікації їх як «своїх» або «чужих». Чужорідні клітини піддаються далі класифікації з метою стимуляції захисного механізму відповідного типу.

## 1.2 Аналіз ШНМ

Штучні імунні системи - це адаптивні системи, в основі яких лежить теоретична імунологія та спостережувані принципи, функції і моделі імунітету, що застосовуються для вирішення завдань в різних проблемних областях. ШНМ являють собою напрямок штучного інтелекту, що емулює природну імунну систему.

У ШНМ використовується здатність природної імунної системи виробляти нові типи антитіл і відбирати найбільш підходящі з них для взаємодії з потрапили ШНМ використовують обчислювальні моделі обробки інформації в імунологічних взаємодіях з практичними застосуваннями для вирішення багатьох проблем, таких як оптимізація та відновлення функцій, розпізнавання образів, пошук даних, комп'ютерна безпека, виявлення помилок, класифікація, оптимізація та ін. [7].

Імунна система представляє великий інтерес як система, здатна ефективно обробляти значні обсяги даних. Зокрема, вона виконує великий обсяг складних високопараллельних розподілених обчислень. Поведінка імунної системи в цілому визначається всією сукупністю локальних взаємодій.

Відносно можливих додатків для обробки інформації перспективні такі властивості імунної системи:

- розпізнавання;
- виділення особливостей;
- різноманітність; - навчання;
- пам'ять; - розподілу пошук;
- саморегуляції;
- динамічний захист;
- розподіл усіх виявлення.

При розпізнаванні імунна система здатна розпізнавати та класифікувати різні молекулярні структури і вибірково на них реагувати.

Розпізнавання відбувається в ході міжклітинних контактів, при цьому сила утворюються зв'язків визначається формою молекул і величиною електростатичного заряду. Розпізнавання свого і чужого є однією з основних задач, яку вирішує імунна система. У властивостях виділення особливостей антиген-презентируют клітини інтерпретують антигенное оточення і виділяють особливості шляхом обробки антигенів та подання антигенних пептидів на своїй поверхні.

При розмаїтті імунна система використовує комбінаторний механізм (генетично-обумовлений процес) для утворення безлічі різних рецепторів лімфоцитів, з тим щоб гарантувати, що хоча б один лімфоцит з усієї сукупності зможе провзаємодіяти з будь-яким наперед заданим (відомим або невідомим) антигеном. У властивостях навчанні імунна система оцінює структуру конкретного антигену, використовуючи його випадкові контакти з складовими цю систему клітинами.

Навчання складається в зміні концентрації лімфоцитів, яке відбувається при первинному відповіді (в результаті першого контакту з антигеном). Отже, здатність імунної системи до навчання закладена, головним чином, в механізмі поповнення клонів, що приводить до утворення нових імунокомпетентних клітин з урахуванням поточного стану системи (цей процес має назву розмноження клону).

У категорії пам'ять невелика частина лімфоцитів, що знаходяться в активованому стані, стає клітинами пам'яті (асоціативна пам'ять). Вважається, що час життя клітин пам'яті є динамічною величиною і визначається частотою стимуляції антигенами. Використовуючи короткочасні і довгострокові механізми імунної пам'яті, імунна система підтримує ідеальний баланс між економією ресурсів і виконанням функції за рахунок збереження мінімально необхідної, але достатньо пам'яті про попередніх контактах з антигеном.

При розподіленому пошуку імунна система - це розподілена система. Клітини імунної системи, головним чином лімфоцити, безперервно

рециркулируют через кров, лімфу, лімфоїдні органи і інші тканини. У разі зустрічі з антигеном вони здійснюють специфічну імунну відповідь.

У саморегуляція імунний захист має властивість саморегуляції. Центрального органу, який контролює функції імунної системи, не існує. Залежно від способу проникнення в організм і інших властивостей антигену, регуляція імунної відповіді може бути як локальної, так і системною.

У властивості пороговий механізм імунну відповідь і розмноження імунокомпетентних клітин відбуваються лише після подолання деякого порогу, що залежить від сили хімічних зв'язків. При динамічної захисту клональное розмноження і соматичне гіпермутірованіе дозволяють імунній системі продукувати високоафінні імунокомпетентні клітини, що створює динамічний баланс між вивчає і захисною функцією адаптивного імунітету. Наявність динамічного захисту поступово призводить до розширення зони спостереження, контрольованої імунною системою.

У імовірнісний виявленні перехресні реакції в ході імунної відповіді - це процес стохастичний. До того ж виявлення антигену завжди неминуче відбувається приблизними чином; отже лімфоцит може взаємодіяти з декількома структурно подібними антигенами.

В імунній відповіді на антиген важливу роль відіграють і інші характеристики імунної системи, такі як адаптованість, специфічність, самотолерантність, диференціювання та інші. Всі ці властивості, що мають відношення до обробки інформації, створюють ряд цікавих можливостей з обчислювальної точки зору.

Області де застосовуються ШНМ:

- методи обчислень;
- когнітивні моделі;
- розпізнавання образів;
- методи виявлення аномалій і несправностей;
- мультиагентні системи;
- моделі самоорганізації;

- моделі колективного інтелекту;
- системи пошуку і оптимізації;
- моделі автономних розподілених систем;
- моделі штучного життя;
- системи комп'ютерної та інтернет - безпеки;
- моделі навчаються систем;
- методи добування інформації;
- штучні імунні системи для виявлення підробок;
- методи обробки сигналів і зображень;
- оптимізація; - розпізнавання образів;
- комп'ютерна безпека;
- інтелектуальний аналіз і вибірка даних;
- навчання адаптивних систем.

## 1.3 Обчислювальні моделі ШНМ

Основними обчислювальними моделями ШНМ є модель клонального відбору, мережевої взаємодії, негативного і позитивного відбору.

### 1.3.1 Модель негативного відбору

Модель негативного відбору є однією з найбільш широко відомих моделей ШНМ (рисунок 1.3). Її принцип роботи полягає в генерації детекторів, відповідних аномальної поведінки досліджуваного об'єкта. Метод негативного відбору складається з двох фаз: навчання та класифікації.

Під час навчання на вхід моделі надходять «свої» антигени (self-antigen). При цьому в ШНМ виробляються випадкові клітини, які стають детекторами лише в тому випадку, якщо вони не реагують ні на один з антигенів, а також і на вже існуючі імунні клітини (антитіла).

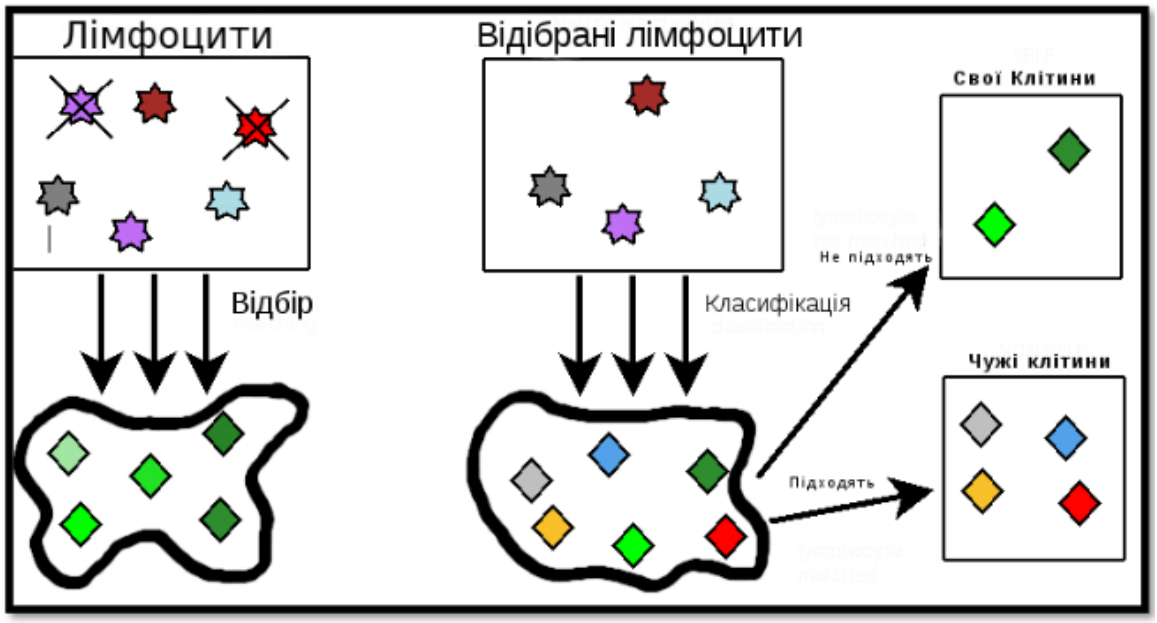


Рисунок 1.3 – Модель негативного відбору

Таким чином, отримана під час навчання популяція клітин в значній мірі визначає простір передбачуваних «чужих» клітин. У процесі розпізнавання клітина вважається патогенної ( «чужий»), якщо її розпізнає хоч один з існуючих детекторів. Як недоліки моделі негативного відбір можна відзначити великі тимчасові затраті на роботу. Однак деякі його модифікації, зокрема V-детектор [7], показали хороші результати і визначили нові шляхи розвитку моделі.

### 1.3.2 Модель позитивного відбору

Позитивний відбір заснований на схемі роботи моделі негативного відбору, за винятком того, що детектором вважають випадкову клітку, що реагує хоча б на один антиген з навчальної вибірки. Очевидно, що негативний відбір в порівнянні з позитивним більш ефективний у випадках, коли область «чужих» клітин істотно менше області «своїх». Таким чином, для правильного вибору моделі необхідно володіти апріорними знаннями про розподіл «своїх» і «чужих» клітин, а це неможливо далеко не завжди.

### 1.3.3 Модель клонального відбору

Клональний відбір є найбільш поширеною моделлю позитивного відбору (рисунок 1.4). Її особливість полягають в тому, що після генерації випадкового позитивного детектора він піддається клонування, а потім здійснюється мутація кожного з клонів. Таким чином, з популяції клонів можна визначити найбільш точно відповідає антигену детектор, який і замінить батьківський детектор в загальній популяції. Цей підхід дозволяє прискорити процес збіжності методу, при цьому розмір популяції істотно не збільшується, що призводить до економії обчислювальних ресурсів.

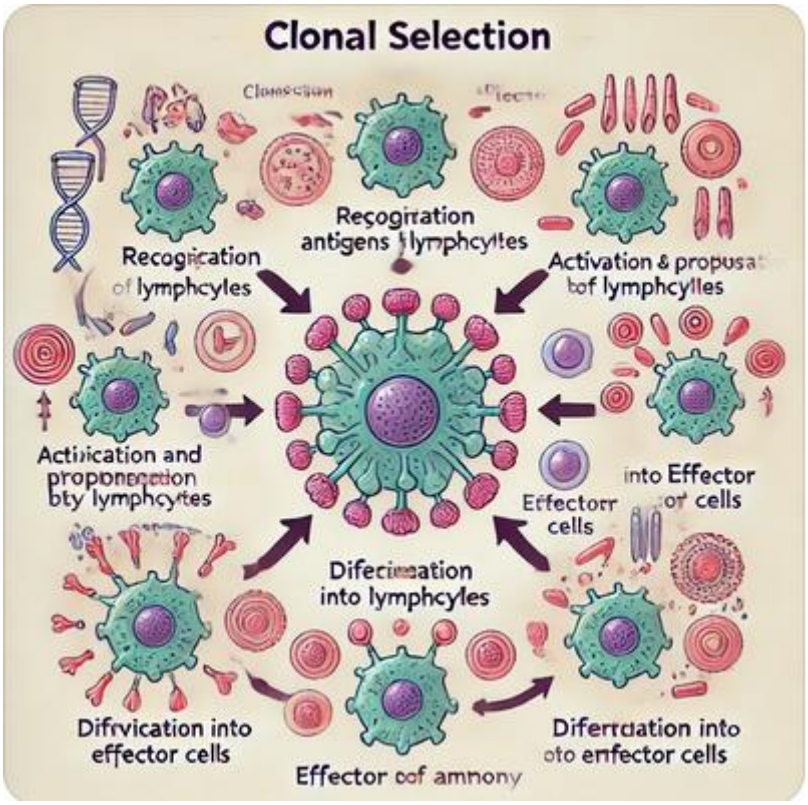


Рисунок 1.4 – Клональний відбір

Характеристики клонального відбору:

- розпізнавання антитілами специфічних антигенів;
- відбір антитіл з високою аффинностью і їх клонування;

- клонування антитіл пропорційно афінності даного антитіла з антигеном - тобто, чим вище аффинность, тим більша кількість клонів проводиться;

- генерація випадкових генетичних змін, шляхом внесення мутацій, для підтримки різноманітності в популяції антитіл;

- мутація, якій піддаються антитіла назад-пропорційна аффинности антитіл до антигену;

- видалення новостворених антитіл, якщо вони мають низьку афінність;

- наявність клітин пам'яті для збереження знайдених рішень.

### 1.3.4 Модель мережевої взаємодії

Імунна система являє собою регульовану мережу молекул і клітин, які розпізнають один одного навіть за відсутності антигену (рисунок 1.5). Такі структури часто називають ідіотіпіческіх мережами, вони служать математіче¬ской основою для вивчення поведінки імунної системи. Така тео¬рія інтерпретується у вигляді системи диференціальних рівнянь, яка описує динаміку концентрації клонів лім¬фоцітов і відповідних молекул імуноглобулінів.

Теорія ідіотіпіческіх регуляції заснована на припущенні, що різні клони лімфоцитів один від одного не ізольовані, а підтримують зв'язок шляхом взаємодій між своїми рецепторамі і антитілами. Отже, розпізнавання антигену здійснюється не поодиноким клоном клітин, а скоріше на системному рівні, за участю різних клонів, взаємодіючих за типом реакцій антиген-антитіло як єдина мережа.

В ході імунної відповіді сам антиген викликає лише вироблення першого набору антитіл Ab1. Потім ці антитіла, діючи в якості антигену, викликають вироблення другого набору «анти-ідіотіпіческіх» антитіл Ab2, розпізнаючих ідіотопи на антитіла Ab1. Аналогічно здійснюється вироблення третього набору антитіл, Ab3, які розпізнають антитіла Ab2, і так далі.

Ключовий постулат теорії полягає в тому, що еді¬нічная клітина продукує лише один тип антитіл; звідси слідує кілька висновків:

- існує аллельному виключення;

- всі антитіло-подібні рецептори на поверхні лімфоцита повинні бути ідентичними або, принаймні, мати ідентичні легкі ланцюги і варіабельні області важких ланцюгів;

- всі антитіла, які продукують даної клітиною, і її нащадки повинні мати однаковий ідіотипів.

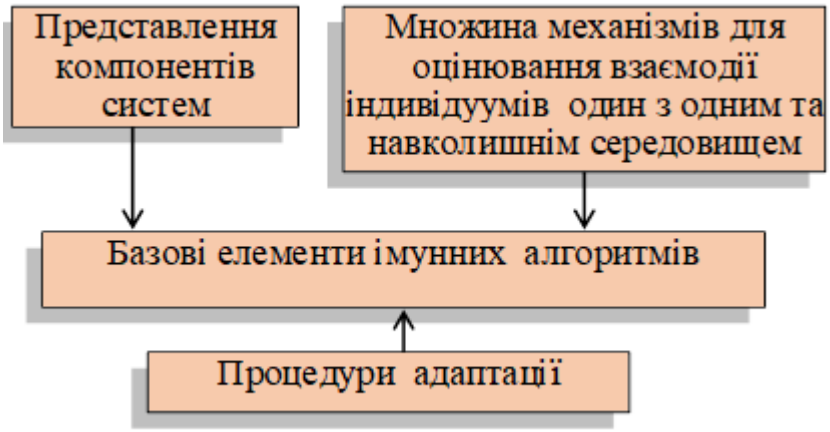


Рисунок 1.5 – Компоненти ШНМ

Формулюючи основи своєї теорії, були введені поняття формальних та функціональних мереж. Формальні мережі служать для вивчення питань репертуару, дуалізму і супрессии. При розгляді функціо¬нальних мереж представлена кількісна картина теорії. Даний підхід гранично математизированной і, в основному, пов'язаний з описом фазових переходів. Надалі площину фазових змінних даної системи рівнянь була розділена на докритическим область, область переходу і посткритическую область.

1.3.5 Інші моделі

Існують і інші обчислювальні моделі, що імітують різні імунологічні

явища, наприклад здатність імунної системи знаходити спільні структури в зашумленной середовищі, можливість виявляти і підтримувати охоплення структур, що належать різним класам, а також здатність до ефективного навчання навіть у тому випадку, якщо експресувати не всі молекули імуноглобулінів і не всі антигени присутні.

У декількох роботах для моделювання процесу соматичних мутацій (при якому утворюються антитіла для розпізнавання конкретного антигену) застосовувалися генетичні алгоритми. Хоффман [22] зіставив імунну систему з нервовою системою, показав ряд аналогій, наявних на рівні загальної поведінки цих систем (що відрізняються на рівні структурних елементів), і теореті¬ческі допустив існування симетричною моделі нейронної мережі з поведінкою, аналогічним імунної реакції у відповідь на дію стимулу. Фармер і ін. [13], а також Берсін і Варела [12] зіставили імунну систему з самонавчанням класифікаторів.

## 1.4 Програмне забезпечення для роботи з ШНМ

В даний час ШНМ представлені багатьма різнорідними інструментами програмного забезпечення, які реалізують евристичні алгоритми, еволюційні обчислення і т.д.

Фірма IBM реалізувала проект створення імунної системи для кіберпростору, яка включає в себе наступні етапи:

- виявлення невідомого вірусу на машині користувача;
- виділення примірника вірусу і відправка його центрального комп'ютера;
- автоматичний аналіз вірусу і вироблення приписи для його розпізнавання і видалення з будь-якого пристрою, підключеного до мережі;
- доставка приписи комп'ютера користувача, вбудовування його в антивірусні бази даних і запуск програми для виявлення і видалення всіх копій вірусу;

- поширення антивірусного приписи в локальній мережі користувача і потім по всій мережі. З часів появи електронної пошти з'явилася система реклами (часто небажано) під назвою спам-розсилки того чи іншого товару або послуги, аналог паперової реклами безкоштовно розповсюджується по поштових скриньках житлових будинків. В даний час становить 70-90% від усіх поштових повідомлень.

Механізми боротьби зі спамом:

- чорні списки відправників;

- сірі списки, що вимагають повторного звернення поштового сервера для відправки;

- контекстні фільтри. Так само розроблена Байєсова фільтрація спаму - метод для фільтрації спаму, заснований на застосуванні байєсівського класифікатора, в основі якого лежить застосування теореми Байєса.

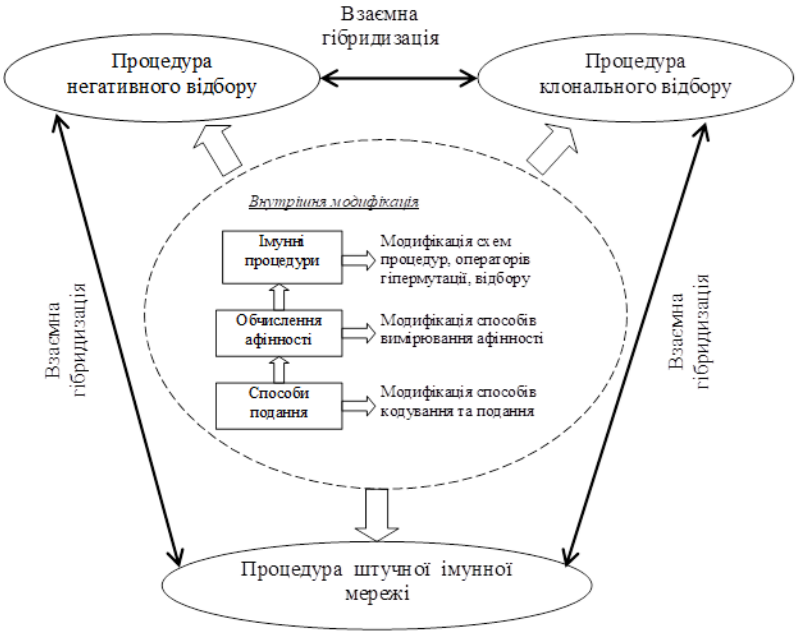


Рисунок 1.6 – Типи гібридизації ШНМ

Переваги: простота і зручність реалізації, ефективність. Принциповий недолік: метод базується на припущенні, що одні слова частіше

зустрічаються в спам, а інші - в звичайних листах, і неефективний, якщо це припущення невірно. Існує метод Байесови отруєння, що дозволяє додати багато зайвого тексту, іноді ретельно підібраного, щоб «обдурити» фільтр. Чи не принциповий недолік - метод працює тільки з текстом.

Одним з наслідків впровадження засобів боротьби зі спамом стала ймовірність «помилково позитивного» рішення щодо спаму, тобто частина листів, які не є спамом, стала позначатися як спам. У разі агресивної антиспам-політики (знищення листів, які здаються спамом, в автоматичному режимі без повідомлення відправника / одержувача) це призводить до труднообнаружіваемим проблем з проходженням пошти. Для подолання зазначених недоліків існуючих методів боротьби зі спамом була реалізована штучна імунна система для боротьби зі спамом на основі моделі негативного відбору і відповідне програмне забезпечення. Існуюче програмне забезпечення для роботи з ШНМ дозволяє вирішувати тільки вузьке коло завдань, для вирішення яких вона розроблена, що не охоплюючи всю сферу застосування моделі негативного відбору.

# 2 МЕТОДИ АРХІВАЦІЇ ДАНИХ НА ПРИКЛАДІ СТИСНЕННЯ ЗОБРАЖЕНЬ ДЛЯ ІГРОВИХ ПЕРСОНАЖІВ

Основною причиною виникнення задач стиснення даних в системах цифрової обробки даних була перемога розрахункових можливостей процесорів комп’ютерів над системами збереження і передачі даних. Стиснення дозволяє зменшити навантаження на канали передача даних або зменшити обсяг, який займають дані за рахунок їх обробки та перетворення, що як відомо створює навантаження на процесор комп’ютера, тому що стиснення даних так чи інакше пов’язано з великими обсягами математичних розрахунків [1, 2].

Здавалося, що зі збільшенням обсягів носіїв даних, зі зростом пропускної здатності мережного обладнання ця проблема відійде у минуле, проте користувачі і прогрес постійно бажають еволюції, покращення якості, ефектності у їх спілкуванні з цифровим світом, що у свою чергу призводить до збільшення обсягів потрібної для цього інформації. Враховуючи вищесказане ми розуміємо, що стиснення даних – це минуле, сьогодення і майбутнє, широко використане та з наявністю попиту. Тому для зменшення часу обробки и передачі інформації каналами зв'язку і зниження обсягу даних, що зберігаються на засобах ЕОМ, необхідно розробляти нові методи або модернізувати існуючи методи компактного представлення зображень. Таким чином, метою статті є аналіз і оцінка методів стиску зображень в системах цифрової обробки даних для вибору перспективного підходу до проблеми стиснення зображень. Основний матеріал У класифікації інформації зокрема можна виділити таку, що розподіляється на критичну до точності передачі даних, відносно критичну та некритичну. Критичність залежить від обсягу корисної інформації на одиницю (байт) фактичної. Залежно від рівня критичності даних алгоритми поділяються на втратне та безвтратне стиснення.

Стиснення без втрат (англ. Lossless) – метод стиснення даних, при використанні якого закодована інформація може бути відновлена з точністю до біта. Для кожного з типів цифрової інформації, як правило, існують свої алгоритми стиснення без втрат.

Стиснення з втратами (англ. Lossy compression) – метод стиснення даних, при якому розпакований файл відрізняється від оригіналу, проте є корисним для використання.

Стиснення із втратами найчастіше використовується для мультимедійних даних, особливо для потокової передачі даних та телефонії. Алгоритми стиснення без втрат.

Алгоритм RLE. Даний алгоритм надзвичайно простий у реалізації. Групове кодування Run Length Encoding (RLE) – один із найстаріших і найпростіших алгоритмів архівації графіки.

Зображення в ньому (як і в деяких алгоритмах, описаних нижче) витягається в ланцюжок байт по рядках растру. Стиснення в RLE відбувається за рахунок того, що у вихідному зображенні зустрічаються ланцюжки однакових байт. Заміна їх на пари <лічильник повторень, значення> зменшує надмірність даних. Алгоритм розрахований на ділову графіку – зображення з більш великими областями повторюваного кольору. Для цього простого алгоритму можлива і загалом вірогідна ситуація, коли розмір даних після стиснення збільшується. Така ситуація можлива при застосуванні групового кодування до оброблених кольорових фотографій. Для того, щоб збільшити зображення у два рази, його треба застосувати до зображення, у якому значення всіх пікселів один за одним попарно не повторюються.

Характеристики алгоритму RLE.

Коефіцієнти компресії: від 32 до 0,5. Клас зображень: орієнтований алгоритм на зображення з невеликою кількістю кольорів, тобто на ділову й наукову графіку. Симетричність: приблизно одиниця. Характерні риси: не вимагає додаткової пам'яті при пакуванні і розпаковуванні, а також швидко

працює. Цікава особливість групового кодування полягає в тому, що ступінь архівації для деяких зображень може бути істотно підвищений всього лише за рахунок зміни порядку кольорів у палітрі зображення [1].

Алгоритм LZ.

Існує досить велике сімейство LZ-подібних алгоритмів, що розрізняються, наприклад, методом пошуку повторюваних ланцюжків. Один з досить простих варіантів цього алгоритму, припускає, що у вхідному потоці ідуть або пари <лічильник, зсув щодо поточної позиції>, або просто <лічильник> "пропуску" байт і самі значення байтів. При розпакуванні для пари <лічильник, зсув> копіюються <лічильник> байт із вихідного масиву, отриманого в результаті розпакування, на <зсув> байт раніше, а <лічильник> (тобто число, що дорівнює лічильнику) значень "пропуску" байт просто копіюються у вихідний масив із вхідного потоку. Даний алгоритм є несиметричним за часом, оскільки вимагає повного перебору буфера при пошуку однакових ланок. У результаті нам складно задати великий буфер через різке зростання часу компресії. Однак потенційна побудова алгоритму, у якому на <лічильник> і на <зсув> буде виділено по 2 байти (старший біт старшого байта лічильника - ознака повтору рядка / копіювання потоку), дасть нам можливість стискати всі повторювані ланки рядка розміром до 32 Кб у буфері розміром 64 Кб. При цьому ми одержимо збільшення розміру файлу в найгіршому разі на 32770/32768 (у двох байтах записано, що потрібно переписати у вихідний потік наступний 215 байт). Максимальний коефіцієнт стиску досягає границі 8192, оскільки максимальний стиск ми одержуємо, перетворюючи 32 Кб буфера в 4 байти, а буфер такого розміру ми отримуємо не відразу. Однак, мінімальна ланка рядка, для якої нам вигідно проводити стиск, повинна складатися в загальному випадку мінімум з 5 байт, що й визначає малу цінність даного алгоритму. До достоїнств LZ можна віднести надзвичайну простоту алгоритму декомпресії [1, 2].

Алгоритм LZW.

Розглянутий нами нижче варіант алгоритму буде використовувати

дерево для подання й зберігання ланцюжків. Очевидно, що це досить сильне обмеження на вид ланцюжків, і далеко не всі однакові ланки в нашому зображенні будуть використані при стисненні. Однак у пропонованому алгоритмі вигідно стискати навіть ланцюжки, що складаються з 2 байт. Процес стиску виглядає досить просто. Зчитуються послідовно символи вхідного потоку й перевіряється, чи є в створеній таблиці рядків такий рядок. Якщо рядок є, то зчитується наступний символ, а якщо рядок відсутній, то заноситься в потік код для попереднього знайденого рядка, заноситься рядок у таблицю й починається пошук знову.

Алгоритм LZW реалізовано в форматах GIF та TIFF.

Характеристики алгоритму LZW.

Коефіцієнти компресії: від 1000 до 5/7. Стиснення в 1000 разів досягається тільки на одноколірних зображеннях розміром кратним приблизно 7 Мб. Клас зображень: 8-бітні зображення, побудовані на комп'ютері. Стискає за рахунок однакових ланок у потоці. Симетричність: майже симетрична, за умови оптимальної реалізації операції пошуку рядка в таблиці.

Характерні риси: ситуація, коли алгоритм збільшує зображення, зустрічається вкрай рідко.

Алгоритм LZW універсальний - саме його варіанти використовуються у звичайних архіваторах. Класичний алгоритм Хаффмана. Один із класичних алгоритмів, відомих з 60-х років. Використовує тільки частоту появи однакових байт у зображенні. Зіставляє символам вхідного потоку, які зустрічаються більше число раз, ланцюжок біт меншої довжини. І, навпаки, що зустрічається рідко - ланцюжок більшої довжини. Для збору статистики вимагає двох проходів по зображенню. Класичний алгоритм Хаффмана вимагає запису у файл таблиці відповідності кодуємих символів і ланцюжків, що кодують. На практиці використовуються його різновиди. Так, у деяких випадках резонно або використовувати постійну таблицю, або будувати її "адаптивною", тобто в процесі пакування/розпакування. Ці прийоми рятують

нас від двох проходів по зображенню й необхідності зберігання таблиці разом з файлом. Кодування з фіксованою таблицею застосовується як останній етап архівації в JPEG і в розглянутому нижче алгоритмі CCITT Group 3.

Характеристики класичного алгоритму Хаффмана. Коефіцієнти компресії: від 8 до 1. Клас зображень: практично не застосовується до зображень у чистому вигляді. Зазвичай використовується як один з етапів компресії в більш складних схемах. Симетричність: 2 (за рахунок того, що вимагає двох проходів по масиву стиснених даних). Характерні риси: єдиний алгоритм, що не збільшує розміру вихідних даних у найгіршому разі (якщо не вважати необхідності зберігати таблицю перекодування разом з файлом).

Алгоритм Хаффмана з фіксованою таблицею CCITT Group 3.

Близька модифікація алгоритму використовується при стисненні чорно-білих зображень (один біт на піксел). Послідовності один за одним ідуть чорних і білих крапок у ньому заміняються числом, що дорівнює їхній кількості, які в свою чергу, стискуються по Хаффману з фіксованою таблицею. Алгоритм реалізований у форматі TIFF. Характеристики алгоритму CCITT Group 3. Коефіцієнти компресії: від 213 до збільшення файлу в 5 разів. Клас зображень: двокольорові чорно-білі зображення, у яких переважають більші простори, заповнені білим кольором. Симетричність: близька до 1. Характерні риси: даний алгоритм надзвичайно простий у реалізації, швидкий і може бути легко реалізований апаратно [1]. JBIG. Алгоритм розроблений групою експертів ISO спеціально для стиснення однобітних чорнобілих зображень. Може застосовуватися до 2-х, 4-х бітових картинок. При цьому алгоритм розбиває їх на окремі бітові площини. JBIG дозволяє управляти такими параметрами, як порядок розбивки зображення на бітові площини, ширина смуг у зображенні, рівні масштабування. Остання можливість дозволяє легко орієнтуватися в базі більших за розмірами зображень, переглядаючи спочатку їхні зменшені копії. Розпаковуватися зображення на екрані буде поступово, як би повільно "проявляючись".

Lossless JPEG. Алгоритм розроблений групою експертів в області фотографії. На відміну від JBIG, Lossless JPEG орієнтований на повнокольорові 24бітні або 8-бітні в градаціях сірого зображення без палітри. Він являє собою спеціальну реалізацію JPEG без втрат. Коефіцієнти стиску: від 20 до 1.

Lossless JPEG рекомендується застосовувати в тих додатках, де необхідна побітова відповідність вихідного й декомпресованого зображень.

DEFLATE – це алгоритм, що використовує комбінацію алгоритму LZ і алгоритму Хаффмана. Цей метод кодування використовує принцип ковзного вікна і враховує то що раніше зустрічалася інформація, тобто інформація, що вже відома для кодувальника й декодувальника (друге й наступне входження деякого рядка символів у повідомленні заміняються посиланнями на її перше входження). Ковзне вікно можна представити у вигляді буфера або більш складної динамічної структури даних. Таким чином, сам процес стискаючого кодування згідно LZ нагадує написання програми, команди якої дозволяють звертатися до елементів "ковзного вікна", і замість значень стисненої послідовності вставляти посилання на ці значення в "ковзному вікні". Розмір ковзного вікна може динамічно змінюватися й становити 2 КБ, 4 КБ або 32 КБ. Слід також зазначити, що розмір вікна кодувальника може бути меншим або дорівнювати розміру вікна декодувальника, але не навпаки.

Характеристики алгоритму DEFLATE.

Коефіцієнти компресії: від 1000 до 1/3. Клас зображень: будь-які зображення, бажано з великими одноколірними ланцюжками. Симетричність: близька до 1. Характерні риси: достатньо простий у реалізації, швидкий, широко використовується у програмному забезпеченні. Недоліком алгоритму є мала ефективність при кодуванні незначного обсягу даних. Наведені вище алгоритми досить універсальні й покривають всі типи зображень, з іншого боку – у них, на сьогодні, занадто малий коефіцієнт архівації. Використовуючи один з алгоритмів стиску без втрат, можна забезпечити архівацію зображення приблизно у два рази, хоча це і багато в чому залежить від самого зображення його насиченості, кількості кольорів

складності зображення і т.п. У той же час алгоритми стиску із втратами оперують із коефіцієнтами в середньому 10 – 200. Крім можливості модифікації зображення, одна з основних причин подібної різниці полягає в тому, що традиційні алгоритми орієнтовані на роботу з ланцюжком. Вони не враховують "когерентність областей" у зображеннях, що полягає в малій зміні кольору й структури на невеликій ділянці зображення. Алгоритми стиснення з втратами.

Алгоритм JPEG – один з найновіших і достатньо потужних алгоритмів. Практично він є стандартом для повнокольорових зображень. Оперує алгоритм областями 8x8, на яких яскравість і колір міняються порівняно плавно. Внаслідок цього, при розкладанні матриці такої області в подвійний ряд по косинусах значущими виявляються тільки перші коефіцієнти. Стиснення в JPEG здійснюється за рахунок плавності зміни кольорів в зображенні. Алгоритм заснований на дискретно-косинусному перетворенні (ДКП), вживаному до матриці зображення для отримання нової матриці коефіцієнтів. Для отримання попереднього зображення використовується зворотне перетворення.

ДКП розкладає зображення по амплітудах деяких частот. Таким чином, при перетворенні ми отримуємо матрицю, в якій багато коефіцієнтів або близьких, або рівних нулю. Крім того, завдяки недосконалості людського зору, можна апроксимувати коефіцієнти грубіше без помітної втрати якості зображення. Для цього використовується квантування коефіцієнтів. У найпростішому випадку – це арифметичне побітове зсув вправо. При цьому перетворенні втрачається частина інформації, але можуть досягатися великі коефіцієнти стиснення. Процес відновлення зображення в цьому алгоритмі повністю симетричний. Метод дозволяє стискати деякі зображення в 10 – 15 разів без серйозних втрат. Переваги: задається ступінь стиснення; вихідне кольорове зображення може мати 24 біта на точку. Недоліки: при підвищенні ступеня стиску зображення розпадається на окремі квадрати (8x8). Це пов'язане з тим, що відбуваються більші втрати в низьких частотах

при квантуванні, і відновити вихідні дані стає неможливо. Проявляється ефект Гиббса – ореоли по межах різких переходів кольорів.

Характеристики алгоритму JPEG.

Коефіцієнти компресії: 2 – 200 (задається користувачем). Клас зображень: повнокольорові 24 бітні зображення або зображення в градаціях сірого без різких переходів кольорів (фотографії). Симетричність: 1. Фрактальний алгоритм - це алгоритм стиску зображень з втратами, заснований на застосуванні систем ітерованих функцій до зображень. Даний алгоритм відомий тим, що в деяких випадках дозволяє одержати дуже високі коефіцієнти стиснення (кращі приклади - до 1000 разів при прийнятній візуальній якості) для реальних фотографій природних об'єктів, що недоступно для інших алгоритмів стиску зображень у принципі. Через складну ситуацію з патентуванням широкого поширення алгоритм не одержав.

Основа методу фрактального кодування - це виявлення самоподібних ділянок у зображенні. Метод використовує системи доменних і рангових блоків зображення, блоків квадратної форми, що покривають все зображення. Цей підхід став основою для більшості методів фрактального кодування, застосовуваних сьогодні. Відповідно до даного методу зображення розбивається на безліч неперекритих рангових підзображень і визначається безліч неперекритих доменних підзображень. Для кожного рангового блоку алгоритм кодування знаходить найбільш підходящий доменний блок і афінне перетворення, що переводить цей доменний блок у даний ранговий блок. Структура зображення відображається в систему рангових блоків, доменних блоків і перетворень. Основна складність фрактального стиснення полягає в тому, що для знаходження відповідних доменних блоків потрібен повний перебір. Оскільки при цьому переборі щораз повинні порівнятися два масиви, дана операція виходить досить тривалою. Порівняно простим перетворенням її можна звести до операції скалярного добутку двох масивів, однак ця операція також тривала.

Характеристики фрактального алгоритму. Коефіцієнти компресії: 2 – 2000. Клас зображень: повнокольоровий 24-бітні зображення або зображення в градаціях сірого без різких переходів кольорів. Бажано, щоб області більшої значимості були більш контрастними й різкими, а області меншої значимості – неконтрастними й розмитими. Симетричність: 100 – 100000. Характерні риси: може вільно масштабувати зображення при розархівуванні, збільшуючи його в 2 – 4 рази без появи "сходового ефекту". При збільшенні ступеня компресії з'являється "блоковий" ефект на границях блоків у зображенні [1]. Рекурсивний (хвильовий) алгоритм (wavelet). Орієнтовано алгоритм на кольорові й чорно-білі зображення із плавними переходами. Ідеальний для картинок типу рентгенівських знімків. Коефіцієнт стиску задається й варіюється в межах 5 - 100. При спробі задати більший коефіцієнт на різких границях, що особливо проходять по діагоналі, проявляється "сходовий ефект".

Ідея алгоритму полягає в тому, що зберігається у файл різниця - число між середніми значеннями сусідніх блоків у зображенні, що звичайно приймає значення, близькі до 0. Вейвлетна компресія в сучасних алгоритмах компресії зображень дозволяє значно (до двох разів) підвищити ступінь стиснення чорно-білих і кольорових зображень при порівнянній візуальній якості стосовно алгоритмів попереднього покоління, заснованих на дискретному косинуперетворенні, таких, наприклад, як JPEG.

Для роботи з дискретними зображеннями використовується варіант вейвлет-перетворення, відомий як алгоритм Малла. Вихідне зображення розкладається на дві складові - високочастотні деталі (що складаються в основному з різких перепадів яскравості), і згладжену зменшену версію оригіналу. Це досягається застосуванням пари фільтрів, причому кожна з отриманих складових удвічі менша за вихідне зображення. Як правило, використовуються фільтри з кінцевим імпульсним відгуком, у яких пікселі, що потрапили в невелике "вікно", помножуються на заданий набір коефіцієнтів, отримані значення підсумуються, і вікно зсувається для

розрахунку наступного значення на виході. Між вейвлетами й фільтрами є тісний зв'язок. Вейвлети безпосередньо не фігурують в алгоритмах, але якщо ітерувати відповідні фільтри на зображеннях, що складаються з єдиної яскравої точки, то на виході будуть все чіткіше проступати вейвлети. Оскільки зображення двовимірні, фільтрація здійснюється й по вертикалі, і по горизонталі. Цей процес повторюється багаторазово, причому щораз як вхід використовується згладжена версія з попереднього кроку. Тому що зображення "деталей" складаються зазвичай з набору різких границь, і містять великі ділянки де інтенсивність близька до нуля. Якщо припустимо зневажити деякою кількістю дрібних деталей, то всі ці значення можна просто занулити. У результаті виходить версія вихідного зображення, що добре піддається стиснення. Для відновлення оригіналу знову застосовується алгоритм Малла, але з парою фільтрів, зворотної до вихідного. Характеристики хвильового алгоритму. Коефіцієнти компресії: 2 – 200.

Клас зображень: як у фрактальному і JPEG. Симетричність: 1,5. Характерні риси: при високому ступені стиску зображення розпадається на окремі блоки. Аналіз методів стиснення зображень зведено у табл. 1. Проведемо оцінку методів стиску без втрат (без CCITT, як частково-комерційне рішення) на якість і швидкість стиснення. Для цього використовувався тестовий набір повнокольорових і монохромних зображень у форматі BMP 24bit.

Алгоритм За рахунок чого відбувається стиснення Ілюстрація Тип RLE Послідовне повторювання однакових значень кольорів. Безвтратний LZW Повторювані однотипні послідовності кольорів. Безвтратний Хаффмана Різна частота з'явлення кольорів у зображенні.

За рахунок різної частоти з'явлення пікселів у зображені та їх однотипних рядків.

Монохромні зображення з повторюваними елементами.

Безвтратний Wavelet Поступові переходи кольорів З втратами JPEG Відсутність різких границь З втратами Фрактальний Самонагадування

елементів зображень З втратами

Оптимальним з розглянутих алгоритмів за співвідношенням якістьвтрати є алгоритм DEFLATE. Тому доцільно розробити новий підхід до стиснення зображень на основі методу DEFLATE з використанням етапу попередньої обробки. На цьому етапі повинні підготуватися дані, які по-перше, мають повторювані фрагменти, по-друге, мають велику кількість однакових пікселів на зображенні. Етап попередньої обробки може включати наступні основні алгоритми: визначення кольору фону; визначення наявності зображення; аналіз можливості відділення фрагментів зображення; доведення фрагментів до прямокутної форми; безпосередня обробка фрагментів зображення. Додатково збільшити коефіцієнт стиснення можливо за рахунок вибору кольорової моделі. Так як алгоритм DEFLATE стискає за рахунок різної частоти з'явлення пікселів у зображені, то доцільно використовувати кольорорізницеву модель, в якій у двох кольорових компонентів знижений динамічний діапазон даних для фрагментів зображення з однотипними рядками [3, 4]. Висновки Проведений аналіз основних методів стиснення зображень показав відсутніх універсальних алгоритмів, незалежних від класу зображень. Найбільш перспективним методом стиснення без втрат є DEFLATE, але в чистому вигляді його застосування не дасть вагомих результатів. На рисунку 1.1 представлені методи стиснення зображень.

Таблиця 2.1 – Оцінка методів безвтратного стиснення

| Алгоритм | Середній коефіцієнт | Швидкість |
|---|---|---|
| Хаффман | 2,94 | 89% |
| LZW | 1,49 | 73% |
| RLE | 2,12 | 100% |
| DEFLATE | 3,45 | 81 |

| Алгоритм | За рахунок чого відбувається стиснення | Ілюстрація | Тип |
|---|---|---|---|
| RLE | Послідовне повторювання однакових значень кольорів. | | Безвтратний |
| LZW | Повторювані однотипні послідовності кольорів. | | Безвтратний |
| Хаффмана | Різна частота з'явлення кольорів у зображенні. | | Безвтратний |
| DEFLATE | За рахунок різної частоти з'явлення пікселів у зображені та їх однотипних рядків. | | Безвтратний |
| CCITT-3 | Монохромні зображення з повторюваними елементами | АБ АБ | Безвтратний |
| Wavelet | Поступові переходи кольорів | | З втратами |
| JPEG | Відсутність різких границь | | З втратами |
| Фрактальний | Самонагадування елементів зображень | | З втратами |

Рисунок 2.1 – Методи стиснення зображень

## 2.1 Класифікація методів кодування зображень

Цифрові методи стиснення і відновлення зображень перевершують аналогові за гнучкістю та ефективністю [9]. Їх основною перевагою є скорочення об’єму переданої інформації завдяки її кодуванню [10]. Процес кодування зображення ґрунтується на його представленні у вигляді послідовності символів або цифр, який здійснюється на формалізованій мові. На рисунку 2.1 наведено класифікацію основних методів кодування зображень. Необхідно відзначити, що останнім часом розробки в царині цифрової обробки зображень привели до створення цілого ряду графічних форматів і стандартів на основі методів кодування зображень (найбільш поширеними серед них є BMP, GIF, TIFF, JPEG і MPEG). Під графічним форматом розуміють схему представлення зображення. Крім самого зображення файл графічного формату, зазвичай, містить інформацію про розмір, фізичний розмір, ім’я, дату створення тощо. Часто унікальність

формату полягає тільки в можливості зберігання додаткової інформації (наприклад, спеціальний формат для зберігання фотографій, отриманих із супутника або сучасного фотоапарата, містить, крім самого зображення, таку специфічну інформацію, як час і дату фотографування, вид обладнання, яким був проведений знімок, місце тощо) [13]. Методи стиснення з частковою втратою якості (англ. Lossy compression) ефективно функціонують на всьому діапазоні зображення [9]. При їх застосуванні відновлений файл зображення відрізнятиметься від оригіналу, проте є корисним для використання [3]. Основна особливість методів компресії з частковою втратою якості визначається забезпеченням прийнятої якості відновлення зображення при максимальному ступені стиснення.

Рисунок 2.2 – Класифікація методів кодування зображень

Вони також забезпечують значно ефективніше стиснення, ніж безвтратні методи, оскільки високий ступінь стиснення досягається за рахунок зниження якості зображення під час його відновлення в порівнянні з якістю оригіналу. Кінцева мета методів стиснення з втратами – оптимізація якості зображення під вимоги системи обробки й передачі зображень при задоволенні як “об’єктивних”, так і “суб’єктивних” критеріїв якості відновлюваних зображень.

На сьогодні існують методи, які дозволяють регулювати ступінь стиснення в залежності від необхідної якості відновлення зображення. Тому відновлене зображення може практично не змінитися по відношенню до вихідного, а може і спотворитися до невпізнаваності.

Характерним недоліком методів стиснення з частковою втратою якості є їх висока обчислювальна складність. Проте цей недолік фактично усувається завдяки потужним обчислювальним можливостям сучасної комп’ютерної техніки. Методи стиснення з частковою втратою якості реалізовані в таких форматах, як JPEG, JPEG2000, ART, FIF, PCD. Характеристики найпоширеніших в даний час форматів графічних даних подано на рисунку 2.3.

Використання методів з частковою втратою якості призвело до широкого поширення методів кодування з перетворенням як основної складової методів стиснення і відновлення зображень. Під час кодування з перетворенням використовують перетворення груп відліків для декореляції зображення (ослаблення кореляційних зв’язків між елементами зображення).

Результатом ортогонального перетворення блоку зображення є матриця коефіцієнтів перетворення – трансформанта. Стиснення зображення здійснюється шляхом відбору коефіцієнтів перетворення, які суттєво не впливають на відновлення вихідного зображення. Тобто здійснюється фільтрація елементів трансформанти кожного блоку зображення, тим самим зменшується обсяг трансформанти, який зберігається. З метою вибору ортогонального перетворення для розробки ефективного методу стиснення і

відновлення зображень об’єктів проведемо класифікацію ортогональних перетворень.

В залежності від базисних функцій їх можна розділити на наступні класи: ортогональні перетворення, базисні функції яких не залежать від статистики вихідного сигналу; ортогональні перетворення, базисні функції яких залежать від статистики вихідного сигналу. Дослідження ортогональних перетворень, які віднесено до першого класу, вказує на їх значно менші часові характеристики в порівнянні з перетвореннями, що належать до другого класу. Це пов’язано з тим, що під час використання ортогональних перетворень другого класу для кожного сигналу обчислюється власний ортогональний базис.

Це характеризується високою обчислювальною складністю й відсутністю швидких алгоритмів перетворення. Незважаючи на те, що ортогональні перетворення, базисні функції яких залежать від статистики вихідного сигналу, є оптимальними, тривалий час обробки не дозволяє їх використовувати під час стиснення і відновлення зображень.

Ортогональні перетворення, базиси яких не залежать від вихідного сигналу, можна розділити на такі дві групи : ортогональні перетворення з базисами на основі гармонійних базисних функцій; ортогональні перетворення з базисами на основі частково-постійних базисних функцій. До першої групи належать перетворення Фур’є, дискретно-косинусне перетворення, перетворення Хартлі та ін. Їх базисні функції представлено тригонометричними функціями косинуса і синуса. Прикладом другої групи є ортогональні перетворення Уолша-Адамара, Хаара тощо. У свою чергу ортогональні перетворення з гармонійними базисними функціями можна розділити на:

– ортогональні перетворення в комплекснодійсній області;

– ортогональні перетворення в дійсній області.

| Найменування формату | Розширення | Режими, які підтримуються (біт/піксель) | Алгоритми стиснення, які підтримуються |
|---|---|---|---|
| Portable Network Graphics | PNG | 1, 2, 4, 8, 16, 24, 32, 48 | Deflate |
| Graphic Interchange Format | GIF | 1, 2, 4, 8 | LZW |
| Tagged Image File Format | TIF, TIFF | 1, 8, 12, 15, 16, 24, 32 | LZW, JPEG та ін. |
| Microsoft Windows Bitmap | BMP, RLE | 1, 2, 4, 8, 24 | RLE |
| Intel Leaved Bitmap | ILBM, LBM | 1, 2, 4, 8, 24 | RLE |
| True Vision (Targa) | TGA | 8, 15, 24, 32 | RLE |
| PC Painbrach Picture | PCX | 1, 2, 4, 8, 24 | RLE |
| HSI JPEG | HSI, JPG | 8 | LZW + JPEG |
| Joint Potographic Expert Group | JPG, JPEG | 24 | JPEG |
| Fractal Image Format | FIF | Адаптивно залежать від фрактальної функції | FIF |
| Kodak Photo CD | PCD | 24 | Chroma/2 |
| Encapsulated Post Script | EPS, EPSI, EPI | 1, 2, 4, 8, 24, 32 | JPEG, RLE |
| Dr.Halo CUT | CUT | 1, 2, 4, 8 | Hi |
| IBM Picture Maker PIC | PIC | 1, 2, 4, 8 | Hi |
| Macintosh PICT | PICT | 1, 2, 4, 8 | Hi |

Рисунок 2.3 – Зведені характеристики графічних форматів

Прикладом ортогонального перетворення в комплексно-дійсній області є перетворення Фур’є і його модифікації. Як приклад перетворення, заснованого на другій групі базисів, можна навести перетворення Хартлі і дискретно-косинусне перетворення. Під час кодування з перетворенням зображення представляється в іншому вигляді: спочатку в деякому блоці даних беруться лінійні комбінації відліків (коефіцієнти), а потім відібрані для передачі коефіцієнти фільтруються.

Адаптація методу стиснення і відновлення зображень на основі ортогонального перетворення досягається за рахунок вибору типу перетворення і критерію для відбору й фільтрації коефіцієнтів. Існує ще безліч методів кодування, які або не можливо з достатньою визначеністю

віднести до одного з основних класів, або вони є поєднанням методів, що входять в ці класи. Вибір ортогонального перетворення для розробки ефективного методу стиснення і відновлення зображень ґрунтується на трьох складових:

– вибір оптимального перетворення, що здійснює декореляцію зображень;

– вибір методів кодування коефіцієнтів перетворення, що дозволяють істотно скоротити обсяг інформації з найменшим погіршенням якості зображення під час його відновлення;

– використання принципів психофізичного кодування.

Переваги застосування методу кодування на основі ортогонального перетворення випливають з властивостей самого перетворення, тобто особливостей розподілу енергії серед його коефіцієнтів. Завдяки цим особливостям двовимірний спектр зображення більш зручний для кодування, ніж зображення у вихідному просторовому поданні.

Внаслідок кореляційних зв’язків між елементами вихідного зображення енергія його спектру може концентруватися у відносно невеликій кількості відліків.

Під час реалізації метода кодування з перетворенням зображення ділиться на блоки, після чого здійснюється перетворення кожного з них у набір “більш незалежних” коефіцієнтів. Після цього коефіцієнти фільтруються і кодуються для подальшої передачі.

Прийняті приймачем коди декодуються у коефіцієнти перетворення. Відповідно для відновлення елементів зображення над коефіцієнтами виконується зворотне перетворення.

На рисунку 2.4 наведено процес кодування з перетворенням.

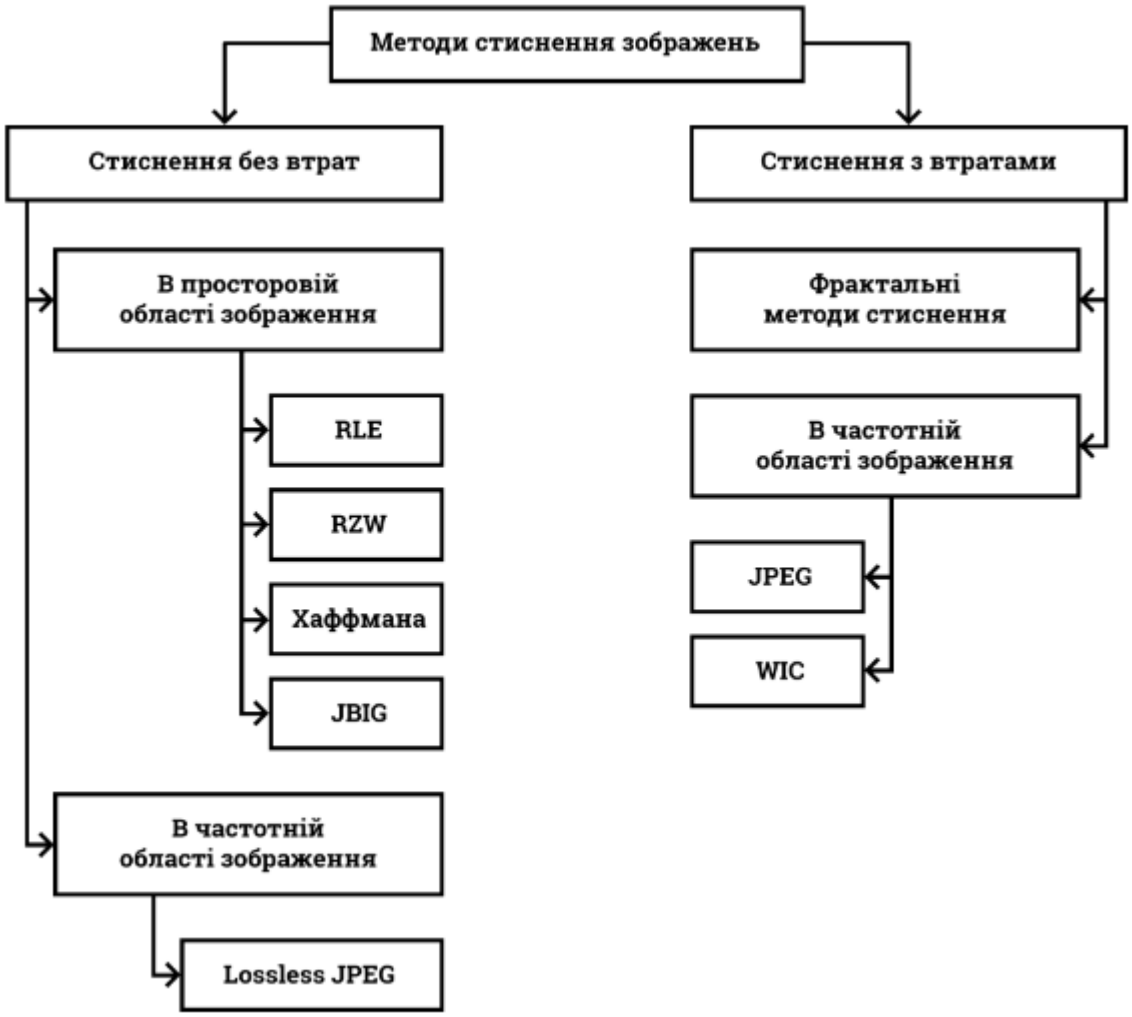


Рисунок 2.4 – Методи архівації даних для тектсур ігрових персонажів

Під час розробки метода стиснення/відновлення зображень на основі ортогонального перетворення слід так само врахувати ряд особливостей:

– під час розбиття вихідного зображення на блоки й роздільного їх кодування, що зберігається за рахунок зв’язку між ними, надмірністю зазвичай нехтують, тому зі статистичного погляду їх розмір доцільно збільшувати; однак, для того, щоб спростити реалізацію та використовувати локальні зміни статистики зображень, необхідно зменшувати їх розмір;

– з практичних міркувань бажано, щоб переважна частина енергії блоку зображення зосереджувалася в меншій кількості коефіцієнтів;

– під час вибору ортогонального перетворення слід враховувати

складність його технічної реалізації. Впровадження цифрових пристроїв обробки інформації допускає використання виключно дискретного варіанту ортогональних перетворень.

Дискретне ортогональне перетворення і відповідне зворотне перетворення визначається наступними виразами:

$$Y(p) = \frac{1}{D} \cdot \sum_{m=0}^{D-1} X(m) \cdot W(p,m), \quad p = \overline{0, \quad D-1}; \tag{2.1}$$

де X(m) – m-й відлік вихідного дискретного сигналу;

Y(p) – p-й коефіцієнт перетворення;

W(p,m) – mй відлік p-ї базисної функції ортогонального перетворення;

D – кількість відліків у вихідному сигналі або в блоці вихідного сигнала під час поблочної обробки;

D 1 – коефіцієнт нормування, який у деяких ортогональних перетвореннях може бути відсутнім.

Базисні функції W(p,m) ортогонального перетворення повинні відповідати вимогам попарно ортогональності та ортонормованості.

При цьому всю множину попарно ортогональних і ортонормованих функцій утворює ортогональний базис. Відліки базисних функцій d зручно представляти у вигляді матриці $W(d) = \lfloor w_{i,j} \rfloor$. У цьому випадку вираз прямого і зворотного перетворень варто записувати в матричному вигляді:

$$Y(d) = < \frac{1}{D} > \cdot X(d) \cdot W(d); \tag{2.2}$$

$$X(d) = Y(d \cdot W^T(d)), \tag{2.3}$$

де W(d) – матриця відліків базисних функцій ортогонального перетворення розмірності N×N;

X(d) –вектор з D відліків вихідного блоку зображення під час поблочної обробки;

Y(d) – вектор з L коефіцієнтів перетворення. Вирази (1.2) і (1.3) описують відповідно пряме та зворотне перетворення. Слід зазначити, що на сьогоднішній день існує безліч ортогональних перетворень, властивості яких і визначають їх спрямованість для різних методів обробки зображень. Середньоквадратичне відхилення (СКВ) є кількісною оцінкою якості, характеризує похибку відновленого зображення та визначається виразом:

$$\sigma = \frac{1}{3} \cdot \sum^{RGB} \sqrt{\frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (x_{i,j}^{(в)} - x_{i,j}^{(н)})^2}{M \cdot N}}, \tag{2.4}$$

де x– елемент вихідного зображення; xв – елемент відновленого зображення; N, M – розмірність зображення по вертикалі і горизонталі відповідно.

Середньоквадратичне відхилення визначається як властивостями ортогонального перетворення, так і характеристиками алгоритму стиснення [15]. Під час розробки методів стиснення і відновлення зображень на основі ортогональних перетворень необхідно забезпечити мінімальне значення цього показника.

Час виконання прямого і зворотного перетворення залежить від кількості арифметичних операцій, які виконувануються під час двовимірного перетворення блоку зображення, складності алгоритму виконання та продуктивності обчислювальної системи. Коефіцієнт стиснення визначається відношенням обсягу вихідного зображення до обсягу стисненого і визначається виразом:

$$K_{ст} = \frac{V_{вих}}{V_{ст}}, \tag{2.5}$$

де $V_{вих}$ – об’єм вихідного зображення;

$V_{ст}$ – об’єм зображення в стисненому вигляді.

Слід зазначити, що для обробки зображень найбільш широке застосування отримали перетворення Фур’є, дискретно-косинусне перетворення, перетворення Хаара, перетворення Уолша-Адамара, перетворення Карунена-Лоева.

Здійснений у розділі аналіз процесів стиснення і відновлення зображень на основі цифрових методів показав, що з метою розробки ефективного методу стиснення і відновлення зображень для конкретних характеристик зображень, що відновлюються, необхідно провести відповідне дослідження ортогональних перетворень.

# 3 РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ

В даний час різноманітність завдань, які стоять перед розробниками програмного забезпечення, є досить широким. Для забезпечення можливості реалізації всіх поставлених проблем розроблено велику кількість різноманітних програмних продуктів для створення самого широкого кола додатків.

Вибір тієї чи іншої інструментального середовища визначається поставленим завданням в кожному індивідуальному випадку. Найбільш якісним і швидким способом можуть реалізувати поставлені завдання наступні інструментальні засоби:

- Microsoft Visual Studio (v6, v7, v8) (C ++ / C #); - Borland C ++ Buider; - Borland Delphi;

- реалізація на мові Java;

- Microsoft Visual Java ++;

- Borland Java Builder; - Microsoft Visual Basic.

Вибір графічних редакторних оболонок викликаний залежністю від сучасних операційних систем, а також можливістю майбутнього програми графічно відображати результати роботи. Природно, робота в текстовому режимі також можлива, і, крім того, вона може істотно збільшити швидкодію програми. Аналіз основних мов програмування дозволив визначити тільки дві, які підходять для написання програми - C ++ і C #.

Вони обидві дозволяють просто створювати графічний інтерфейс і мають невеликі відмінності в синтаксисі. Додаток, що має математичний характер, рідко користується більшістю можливостей графічних компонентів, тому що основна реалізація функцій прихована всередині і грунтується в основному на різних математичних перетвореннях і операціях з пам'яттю. Крім того, реалізація імунних алгоритмів практично неможлива без використання об'єктно-орієнтованого підходу і створення класової моделі

незалежно від мови програмування. Мова C ++ має ряд переваг в порівнянні з іншими мовами: підтримка об'єктно-орієнтованого підходу, велика різноманітність типів даних, гнучкість при роботі з пам'яттю. Мова C # має повністю класову структуру для всіх типів даних (така загальна структурованість може бути віднесена до переваг), але при цьому він втрачає в швидкодії при своїй збільшеній громіздкість опису коду. Мова Pascal має недостатню гнучкість, незадовільну роботу з пам'яттю, Basic занадто простий у виконанні і також недостатньо гнучкий. Java, безсумнівно, виграє у перерахованих вище мов через свою гнучкості і безпосереднього об'єктно-орієнтованого програмування, але його роль проявляється в створенні кроссплатформенних додатків та Інтернет-технологіях. Оцінка засобів розробки програмування по відношенню до заданої задачі дало підставу вибрати в якості програмного середовища розробки Microsoft Visual Studio 2010 року (мова C #).

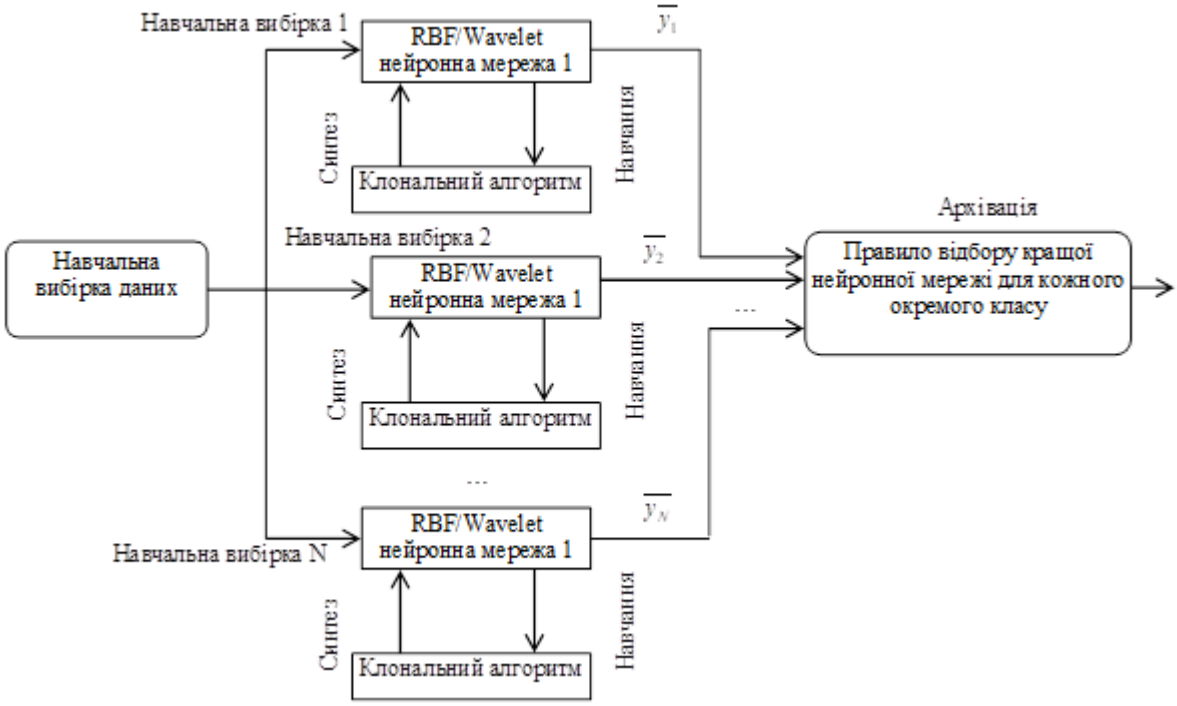


Рисунок 3.1 – Синтез колективу бінарних класифікаторів вейвлет-нейронних мереж

Була розроблена модель програмної системи (рисунок 3.2). Розроблений імунний алгоритм архівації використовує дійсне кодування антитіл, де $Ab_{ij}$ – координати центру антитіла в евклідовому просторі, $r$ – радіус антитіла, що виконує функцію крос-реактивного порога, який адаптивно підбирається в процесі роботи алгоритму.

Формальне подання запропонованої процедури імунної мережі представлено на рисунку.

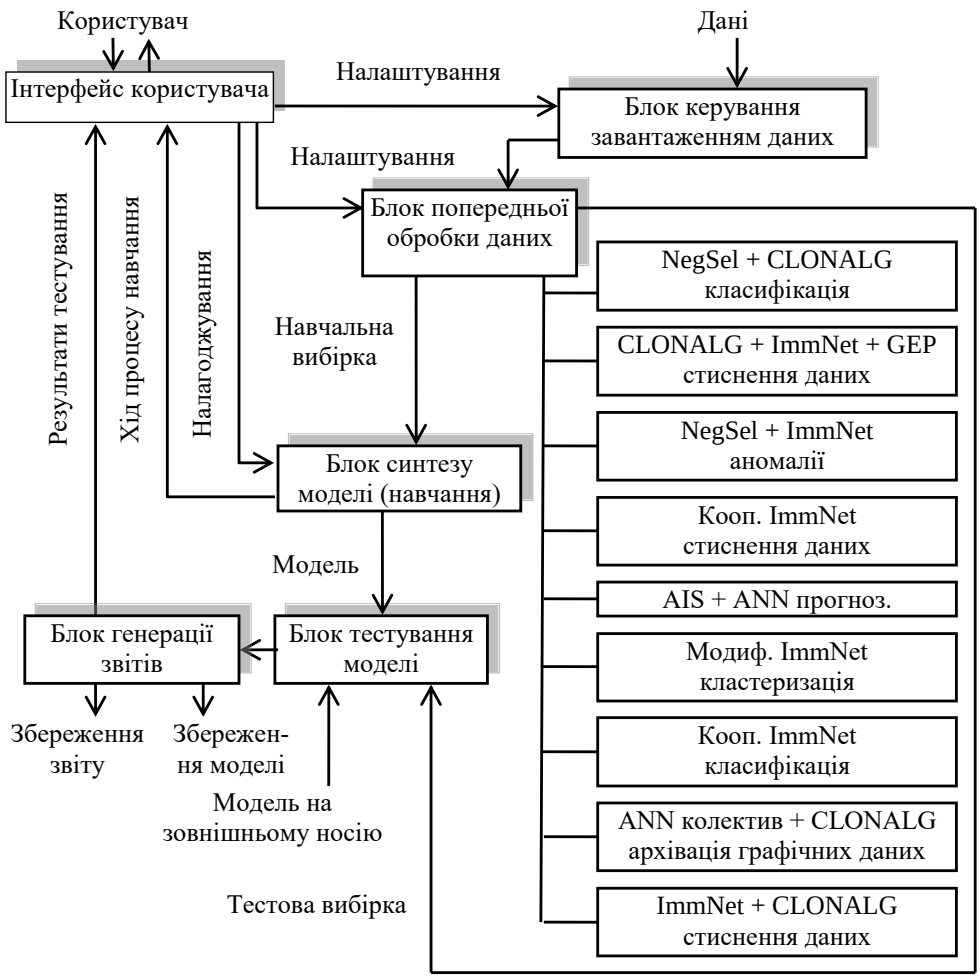


Рисунок 3.2 – Модель програмної системи

## 3.1 Об'єктно орієнтована модель штучної імунної системи

Основними поняттями об'єктно-орієнтованого підходу є об'єкт і клас. Об'єкт визначається як відчутна реальність - предмет або явище, що мають

чітко визначається поведінка. Об'єкт має статки, поведінкою і індивідуальністю; структура і поведінка схожих об'єктів визначають загальний для них клас.

Клас - це безліч об'єктів, пов'язаних спільністю структури і поведінки. Будь-який об'єкт є екземпляром класу. Розглянемо розроблені класи:

- клас DataVector (рисунок 3.3) зберігає і надає доступ до списку довільної довжини data, який зберігає елементи типу double.
- Count - кількість елементів у списку;
- Data - надання доступу до самих даних; Equals - перевантажений метод, що виконує порівняння двох об'єктів типу DataVector.
- EuclideAffinity - обчислення афінності за формулою Еквкліда;
- HemmingAffinity - обчислення афінності за формулою Хемминга;
- ManhattanAffinity - обчислення афінності по Манхеттенського віддалі.
- LoadVectors - завантаження векторів з файлу;
- ToString - перетворення даних, що зберігаються в рядок.

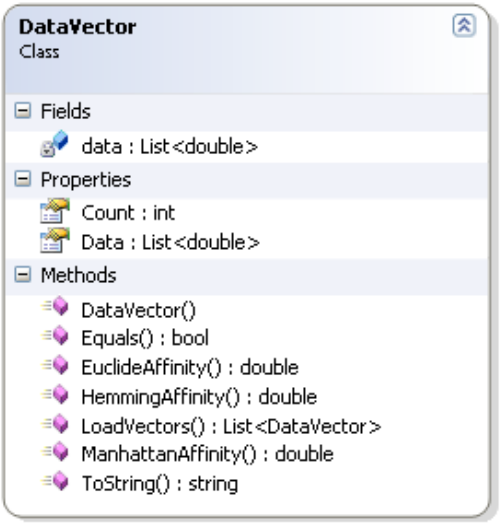


Рисунок 3.3 – Класи

Клас DoubleRange (рисунок 3.4) зберігає діапазон чисел, обмежений

мінімальним і максимальним значенням.

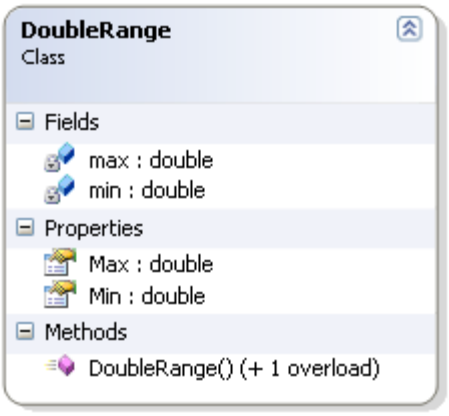


Рисунок 3.4 – Класи

Клас NegativeSelectionAlgoritm (рисунок 3.4) Цей клас виконує реалізацію алгоритму негативного відбору:

- alienElements - чужорідні елементи;
- detectors - детектори; vectors - «свої» елементи;
- rand - об'єкт класу, що відповідає за генерацію випадкових чисел; ranges - мінімальне і максимальне значення кожного поля в «своїх» даних;
- LoadData - виконує завантаження «своїх» даних;
- LoadAlienData - виконує завантаження «чужих» даних;
- GenerateDetector - виконує генерацію одного детектора; GenerateDetectors - виконує генерацію count детекторів;
- GenerateDataFile - генерує базу значень, які визначаються як «свої» і виписує їх в файл;
- GenerateAlienElements - генерує базу значень, які визначаються як «чужі» і виписує їх в файл;
- EvalfRanges - пошук мінімальних і максимальних значень полів в «своїх» даних;
- IsSelf - виконує перевірку входження елемента v в список «своїх»
- DoWork - виконує пошук «чужих» рядків з урахуванням алгоритму

обчислення афінності і точності обчислень.

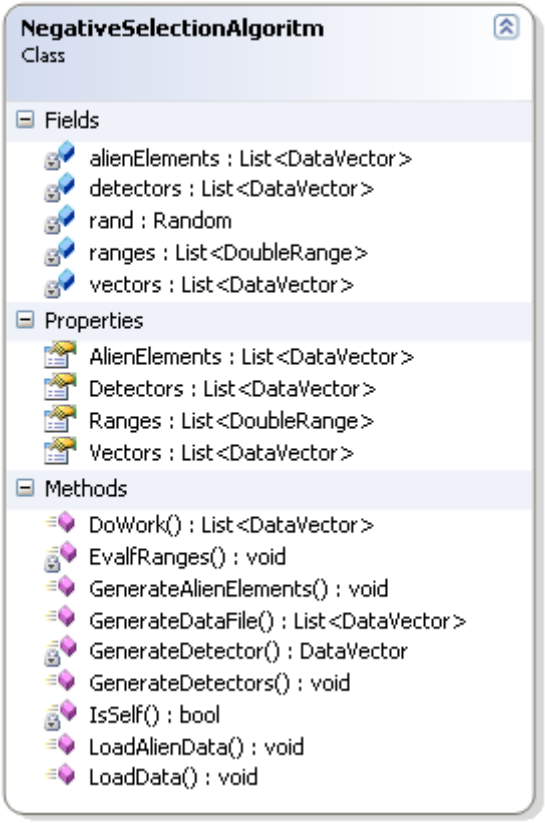


Рисунок 3.5 – Класс NegativeSelectionAlgoritm

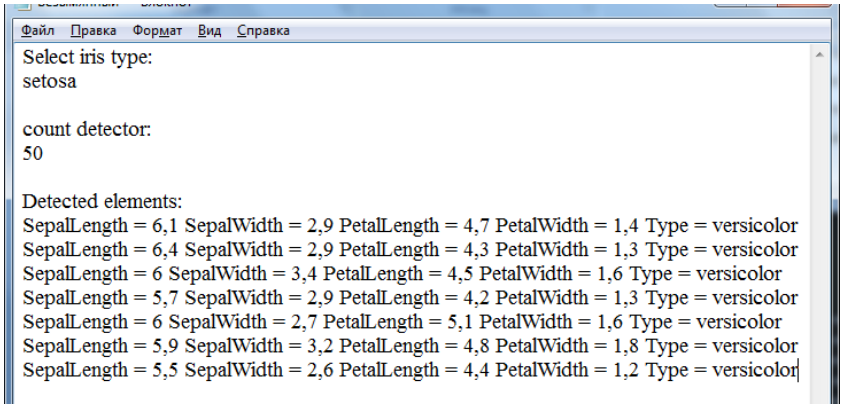


Рисунок 3.6 – Результати роботи

Для функціональної верифікації розробленого ПЗ використана база даних по ірісам Фішера. Іриси Фішера складаються з даних про 100

примірниках ірису, по 50 примірників з трьох видів - Iris setosa, Iris versicolor. Для кожного примірника виміряні чотири характеристики (в сантиметрах):

- довжина чашолистки;
- ширина чашолистка;
- довжина пелюстки;
- ширина пелюстки.

Приклад роботи програми надано на рисунку 3.6. В якості вихідних даних вибирається один із сортів ірисів (Iris setosa). Процес навчання залючается в тому, що відповідно до алгоритму негативного відбору формується набір детекторів таким чином, щоб жоден з детекторів збігався з жодною з рядків «свого» сорти ірисів. Для проведення тестування завантажуються тестові дані - в рядки «свого» сорти ірисів внесені аномальні дані («чужі» сорти ірисів - Iris versicolor). За допомогою детекторів, згенерованих на етапі навчання спроектована імунна система знаходить аномалії у вихідних даних.

Результат програми виведенню в текстовий файл і видно, що при початковому етапі програми була пропозиція користувачу вибрати який тип ірису необхідно вибрати, який буде вважатися як «свій». Був обраний тип «setosa», далі було запропоновано ввести кількість детекторів, які будуть захищати «свої» рядки: користувач ввів 50 детекторів. І на останньому етапі, де визначаються «чужі» рядки, програма надала кількість примірників ірису типу «versicolor» в обсязі 7 рядків.

# ВИСНОВКИ

Обробка та архівація даних ігрових персонажів з використанням штучних імунних мереж – це передова технологія, яка застосовується для забезпечення безпеки та збереження важливої інформації в ігрових системах. ШІМ здатна масштабуватися відповідно до потреб ігрової системи, забезпечуючи ефективний захист незалежно від обсягів даних. Система адаптується до нових загроз без необхідності ручного втручання, що робить її особливо корисною для динамічних онлайн-ігор.

В ході підготовки кваліфікаційної роботи розглянуто методи обробки та архівації даних ігрових персонажів. Розглянуто існуючі види штучних нейронних та імунних систем. Запропоновано модель штучної імунної мережі для архівації даних. Представлені підходи та процедури розв’язання завдання архівації даних з використанням штучних імунних систем та штучних нейронних мереж типу радіально-базисних функцій.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Бурда А.С., Прудіус М.А., Стефанюк Я.Г., Фомічов О.О. Методи обробки та інтелектуального аналізу даних з використанням штучних імунних систем // Системи управління, навігації та зв'язку. 2024. № 3. С.100-103.

2. Ротштейн А. П. Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети.– Винница: УНИВЕРСУМ – Винница, 1999.– 320 с.

3. Искусственные иммунные системы и их применение / Под ред. Д. Дасгупты. Пер. с англ. под ред. А.А.Романюхи. – М.: ФИЗМАТЛИТ, 2006. – 344 с.

4. Timmis J. An Overview of Artificial Immune Systems / J. Timmis, T. Knight, L.N. de Castro, E. Hart // Computation in Cells and Tissues: Perspectives and Tools for Thought, Natural Computation Series, Springer, 2004 – pp. 51-86.

5. Castro L. N. Artificial Immune Systems: A Novel Paradigm to Pattern Recognition / Castro L. N., Timmis J. I. // In Artificial Neural Networks in Pattern Recognition**, S**OCO-2002, University of Paisley, UK, pp. 67-84.

7. Castro L. N. The Clonal Selection Algorithm with Engineering Applications / Castro L. N., Von Zuben, F. J. // In Proceedings of GECCO'00, Workshop on Artificial Immune Systems and Their Applications, pp. 36-37.

8. Борисов В.В., Круглов В.В., Федулов Ф.С. Нечеткие модели и сети. – М: Горячая линия. – Телеком, 2007. 7. Е.В. Бодянский, О.Г. Руденко Искусственные нейронные сети: архитектуры, обучение, применения // Харьков: ТЕЛЕТЕХ, 2004. – 372с.

9. Л. А. Растригин Адаптация сложных систем. Методы и приложения // Рига: Зинатне, 1981.– 375 с.

10. Я. 3. Цыпкин Адаптация и обучение в автоматизированных системах // М., Наука, 1968.

11. Методы сжатия даннях / В. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М.: ДИАЛОГ – МИФИ, 2002. – 384 с.

12. Зив Дж. Алгоритм универсального сжатия данных / Дж. Зив // Проблемы передачи информации. – 1996. – № 2. – С. 47-55.

13. Стрюк А.Ю. Цветовые модели в системах сжатия видеоинформации / А.Ю. Стрюк, К.О. Бохан // Радиоэлектроника и информатика. – 2002. – № 1. – С. 23-25.

14. Гриньов Д.В. Цветоразностная модель представления цвета в задачах сжатия изображений / Д.В. Гриньов // Системи обробки інформації: зб. наук. пр. – Х.: ХУ ПС, 2009. – Вип. 6 (80). – С. 30-34.