

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
в.о. завідувача кафедри
_____ Рязанцев О.І.
« ____ » _____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Моделі і методи системи моніторингу територій

Освітній рівень “Магістр”
Спеціальність 122 “Комп’ютерні науки”

Науковий керівник роботи:

(підпис)

Є.В.Щербаков

(ініціали, прізвище)

Консультант з охорони праці:

(підпис)

Я.О.Критська

(ініціали, прізвище)

Студент:

(підпис)

М.В.Карташов

(ініціали, прізвище)

Група:

КН-19дм

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки

Кафедра Комп'ютерних наук та інженерії

Освітній рівень магістр

Напрямок підготовки _____

(шифр і назва)

Спеціальність 122 "Комп'ютерні науки"

(шифр і назва)

ЗАТВЕРДЖУЮ:

Т.в.о. завідувача кафедри _____

В.С.Кардашук

« _____ » _____ 20 _____ р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Карташову Марку Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи Моделі і методи системи моніторингу територій

керівник проекту (роботи) Щербаков Євген Васильович, к.т.н., доц.

(прізвище, м.я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від «5» 10 2020 р. № 140/15.15

2. Строк подання студентом роботи 10.01.2021

3. Вихідні дані до роботи Матеріали науково-дослідної практики,

науково-методична література, дані інтернет-мережі, супутникові знімки із

SAS.Planet, програмне середовище Quantum GIS, бібліотека мова

програмування Python

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно

розробити) Аналіз сучасного стану розпізнавання образів геоінформаційних

ситуацій в системах моніторингу територій, метод розпізнавання образів

геоінформаційних ситуацій на основі нечіткої штучної нейронної мережі в

системах моніторингу територій, дослідження методу розпізнавання

геоінформаційних ситуацій в системах моніторингу територій, охорона праці та

безпека в надзвичайних ситуаціях, висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	Критська Я.О. ст. викл. кафедри КНІ		

7. Дата видачі завдання 14.10.2020

Керівник

_____ (підпис)

Завдання прийняв до виконання

_____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Розробка технічного завдання	02.09.2020-15.09.2020	
2	Аналіз літературних джерел	16.09.2020-22.09.2020	
3	Аналіз технічних засобів	23.09.2020-25.09.2020	
4	Розробка методу	26.09.2020-10.10.2020	
4	Програмна реалізація	11.10.2020-06.11.2020	
5	Аналіз результатів дослідження	07.11.2020-25.11.2020	
6	Розробка частини проекту "Охорона праці та безпеки в надзвичайних ситуаціях"	26.11.2020-1.12.2020	
7	Оформлення пояснювальної записки, автореферату та презентації	2.12.2020-09.01.2021	

Студент

_____ (підпис)

Карташов М.В.

_____ (прізвище та ініціали)

Науковий керівник

_____ (підпис)

Щербаков Є.В.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Карташов М.В. Моделі і методи системи моніторингу територій.

Об'єктом дослідження магістерської роботи є системи моніторингу території. Предметом дослідження – моделі і методи розпізнавання геоінформаційних ситуацій в системах моніторингу.

Метою дослідження є покращення науково-методичного апарату обробки картографічної інформації про об'єкти моніторингу для підвищення ефективності розпізнавання геоінформаційних ситуацій.

Наукова новизна роботи полягає в тому, що в ній розроблено узагальнену модель геоінформаційних ситуацій на основі їх інформаційних ознак з урахуванням контексту зовнішніх умов та на її основі запропонований метод їх розпізнавання по нечіткій і неповній вихідній інформації.

Було реалізовано розпізнавання вільних місць на парковці, використовуючи геоінформаційні дані із SAS.Planet та програмне середовище Quantum GIS.

Ключові слова: геоінформаційна система, геоінформаційна ситуація, джерело інформації, інформаційно-аналітичний центр, канал передачі інформації, система моніторингу території.

ABSTRACT

Kartashov M.V. Models and methods of the territory monitoring system.

The subject of research is the territory monitoring systems. The subjects of the research are models and methods of recognition of geoinformation situations in monitoring systems.

The purpose of research is to improve the scientific and methodological apparatus for processing cartographic information about the objects of monitoring to increase the efficiency of recognition of geoinformation situations.

The scientific novelty of work is that it developed a generalized model of geoinformation situations based on their information features, taking into account the context of external conditions, and based on it, a method of recognizing them by fuzzy and incomplete source information was proposed.

Parking space recognition was implemented using geo-information from SAS.Planet and the Quantum GIS software environment.

Keywords: geoinformation system, geoinformation situation, source of information, information analytical center, channel of information transmission, territory monitoring system.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	7
1 АНАЛІЗ СУЧАСНОГО СТАНУ РОЗПІЗНАВАННЯ ОБРАЗІВ ГЕОІНФОРМАЦІЙНИХ СИТУАЦІЙ В СИСТЕМАХ МОНІТОРИНГУ ТЕРИТОРІЙ	9
1.1 Сучасні принципи побудови систем моніторингу територій.....	9
1.2 Особливості організації обробки інформації в системах моніторингу територій	14
1.3 Аналіз впливу зовнішніх умов на джерела інформації, що використовуються для розпізнавання образів геоінформаційних ситуацій в системах моніторингу територій .	21
1.4 Постановка завдання дослідження.....	23
2 МЕТОД РОЗПІЗНАВАННЯ ОБРАЗІВ ГЕОНФОРМАЦІЙНИХ СИТУАЦІЙ НА ОСНОВІ НЕЧІТКОЇ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ В СИСТЕМАХ МОНІТОРИНГУ ТЕРИТОРІЙ	26
2.1 Узагальнена модель геоінформаційних ситуацій на основі певних інформаційних ознак у системах моніторингу територій	26
2.2 Побудова функції належності нечітких інформаційних ознак територіальних об'єктів на основі байєсівського правила	33
2.3 Метод розпізнавання образів геоінформаційних ситуацій за нечіткою та неповною вихідною інформацією в системах моніторингу територій	38
2.4 Формування опису основних лінгвістичних змінних територіальних об'єктів у системах моніторингу територій.....	40
3 ДОСЛІДЖЕННЯ МЕТОДУ РОЗПІЗНАВАННЯ ГЕОІНФОРМАЦІЙНИХ СИТУАЦІЙ В СИСТЕМАХ МОНІТОРИНГУ ТЕРИТОРІЙ.....	42
3.1 Вибір інструментів для вирішення поставлених задач.....	42
3.2 Вихідні дані для дослідження.....	43
3.2.1 Структура нечіткого розташування геоінформаційних об'єктів та ситуацій у системах моніторингу територій	43
3.2.2 Вибір архітектури нечіткої штучної нейронної мережі для задачі розпізнавання територіальних об'єктів у системах моніторингу територій\.....	44
3.2.3 Побудова функцій активації нечітких штучних нейронів.....	45
3.3 Тестування методу розпізнавання геоінформаційних ситуацій на основі нечітких штучних нейронних мереж у системах моніторингу територій	47

3.4	Результати оцінювання ефективності розпізнавання геоінформаційних ситуацій на основі нечітких штучних нейронних мереж у системах моніторингу територій.....	48
3.5	Перспективи подальших досліджень.....	52
4	ОХОРОНА ПРАЦІ	54
4.1	Аналіз потенційно небезпечних і шкідливих виробничих факторів , що впливають на персонал.....	54
5.2	Заходи щодотехнікибезпеки	55
4.3	Заходи, що забезпечують виробничу санітарію і гігієну праці.....	58
4.4	Рекомендації по пожежній профілактиці	61
4.5	Охорона навколишнього природного середовища.....	64
	ВИСНОВКИ.....	65
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	67
	ДОДАТОК А. Лістинг модулю «pe_search_results.py».....	71
	ДОДАТОК Б. Лістинг модулю «mosaic_layer_widgets.py».....	89
	ДОДАТОК В. Електронні плакати	94

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

- ЛЛС – логіко-лінгвістична система
- СМТ – система моніторингу територій
- ШНМ – штучна нейронна мережа
- ГС – геоінформаційна ситуація
- ДІ – джерело інформації
- КПІ – канал передачі інформації
- ППВІ – проміжний пристрій прийому й відображення інформації
- СП – сенсорний пристрій
- ПП – приймальний пристрій
- ППО – пристрій попередньої обробки сигнальної інформації
- ІАЦ – інформаційно-аналітичний центр
- АРМ – автоматизоване робоче місце
- ЕС – експертна система
- ГІТ – геоінформаційні технології
- ГІС – геоінформаційна система
- БД – база даних
- ТО – територіальний об’єкт
- ГО – груповий об’єкт
- БНМ – багатошарові нейронні мережі
- ІОЗ – інформаційна ознака
- ІО – інформаційний образ
- СМ – система моделювання
- ЕК – електронна карта

ВСТУП

Обробка інформації є основою багатьох сучасних автоматизованих систем управління різними технічними системами, в тому числі системами моніторингу територій, що вирішують задачу розпізнавання об'єктів і ситуацій обстановки для різних додатків, наприклад, охоронного моніторингу великих промислових підприємств або інтелектуального пошуку місця на парковці.

Сучасні автоматизовані системи моніторингу територій представляють собою складні технічні комплекси, які мають структурну (загальна зона відповідальності для кількох технічних комплексів засобів спостереження за обстановкою) і інформаційну (отримання інформації про об'єкт за сукупністю незалежних каналів різної фізичної природи) надмірність.

Характерною особливістю такого вектора інформації є його мінливість як щодо компонентного складу (кількості каналів надходження інформації), так і якості вихідної інформації для прийняття рішень (неповнота, нечіткість, а іноді, і її суперечливість) в процесі вирішення окремих або групових практичних задач.

Ефективність таких систем багато в чому визначається не тільки рівнем автоматизації управління надмірністю, але і, найголовніше, якістю обробки інформації, що визначає рішення задачі оперативного виявлення і класифікації рухомих і нерухомих об'єктів за неповною, нечіткою і суперечливою інформацією, одержуваною одночасно або в різний час від зосереджених або просторово рознесених джерел різної фізичної природи (радіолокаційні, оптичні, акустичні, сейсмічні та інші) в зонах їх відповідальності.

Обробка інформації різної якості від різнорідних джерел для вирішення задач автоматичного виявлення і класифікації одиночних і групових географічних об'єктів в інтегрованих системах моніторингу територій є актуальною науково-технічною проблемою.

Основна складність її рішення обумовлена відсутністю прийнятних для практичного використання науково обґрунтованих підходів і методів комплексування інформації від різнорідних джерел з метою компенсації нестачі інформації, одержуваної від кожного джерела окремо, при вирішенні задач класифікації рухомих або тимчасово нерухомих об'єктів.

Загальновідомі і добре опрацьовані два підходи до побудови «інтелектуальних» систем: штучні нейронні мережі (ШНМ) і логіко-лінгвістичні системи (ЛЛС). Перевага використання ШНМ полягає в більшій (ніж у випадку використання ЛЛС)

універсальності, тобто в можливості досить простого розширення переліку джерел інформації, що класифікують об'єкти і ситуації. При цьому з'являється можливість класифікації по багатовимірним даним різної фізичної природи, що практично важко здійсненне (через складність опису) за допомогою ЛЛС. Важливою перевагою систем класифікації, побудованих на основі ШНМ, є незмінність архітектури мережі при розширенні переліку об'єктів класифікації з відносно простими алгоритмами навчання мережі, яке може бути виконане кінцевим користувачем. Застосування ж ЛЛС при розширенні переліку об'єктів класифікації вимагає модифікації бази знань системи із залученням її розробника.

Інтелектуальні системи класифікації об'єктів і ситуацій обстановки на основі ШНМ можуть використовуватися в широкому діапазоні нечіткості, неповноти і суперечності вихідних даних. Вони забезпечують високі значення ймовірностей виявлення-класифікації одиночних і групових географічних об'єктів і ситуацій обстановки. Можлива їх реалізація як систем реального (близького до реального) часу.

Однак, максимальна ефективність такого класу систем, в сильному ступені залежить від багатьох факторів – архітектури ШНМ, алгоритмів навчання, способів комплексування інформації від різномірних джерел, фізики процесів її нечіткості, неповноти, суперечливості, інформаційних ознак одиночних і групових об'єктів і ситуацій обстановки та інших.

Наукова актуальність даного дослідження обумовлена необхідністю розвитку теоретичних основ вирішення прикладної проблеми: розпізнавання територіальних одиночних, групових об'єктів і ситуацій обстановки за інформацією від різномірних джерел, які враховують всі перераховані вище фактори, для інформаційного забезпечення автоматизованих систем класифікації та подання на електронних картах локальних ситуацій географічної обстановки.

1 АНАЛІЗ СУЧАСНОГО СТАНУ РОЗПІЗНАВАННЯ ОБРАЗІВ ГЕОІНФОРМАЦІЙНИХ СИТУАЦІЙ В СИСТЕМАХ МОНІТОРИНГУ ТЕРИТОРІЙ

1.1 Сучасні принципи побудови систем моніторингу територій

В даний час системи моніторингу територій (СМТ) представляють собою напрямок, який активно розвивається. Це зумовлене створенням ситуаційних центрів, які можуть вирішувати різні задачі (наприклад, охорона промислово важливих об'єктів і територій).

Під СМТ розуміється сукупність підсистем, що забезпечують моніторинг території з метою автоматичного або автоматизованого виявлення і розпізнавання геоінформаційних ситуацій (ГС). Як правило, вона складається з мережі джерел інформації (ДІ), каналів передачі інформації (КПІ), проміжних пристроїв прийому й відображення інформації (ППВІ) та інформаційно-аналітичного центру (ІАЦ), що реалізує обробку повідомлень від ДІ. По своїй організації СМТ може бути представлена у вигляді багаторівневої системи обробки інформації та прийняття рішень [1 – 3].

Залежно від рівня ієрархії СМТ і розмірів контрольованої території в ній можна виділити 2, 3 і більше рівнів прийняття рішення, що реалізуються автоматично або «ручними» методами.

Структурна схема типової СМТ у вигляді тривірневої системи прийняття рішень представлена на рисунку. 1.1.

На першому рівні рішення по виявленню і класифікації об'єктів приймається технічними засобами – ДІ – в автоматичному режимі, в той час як другий і третій рівні орієнтовані на прийняття рішень в автоматизованому режимі з використанням людино-машинних процедур.

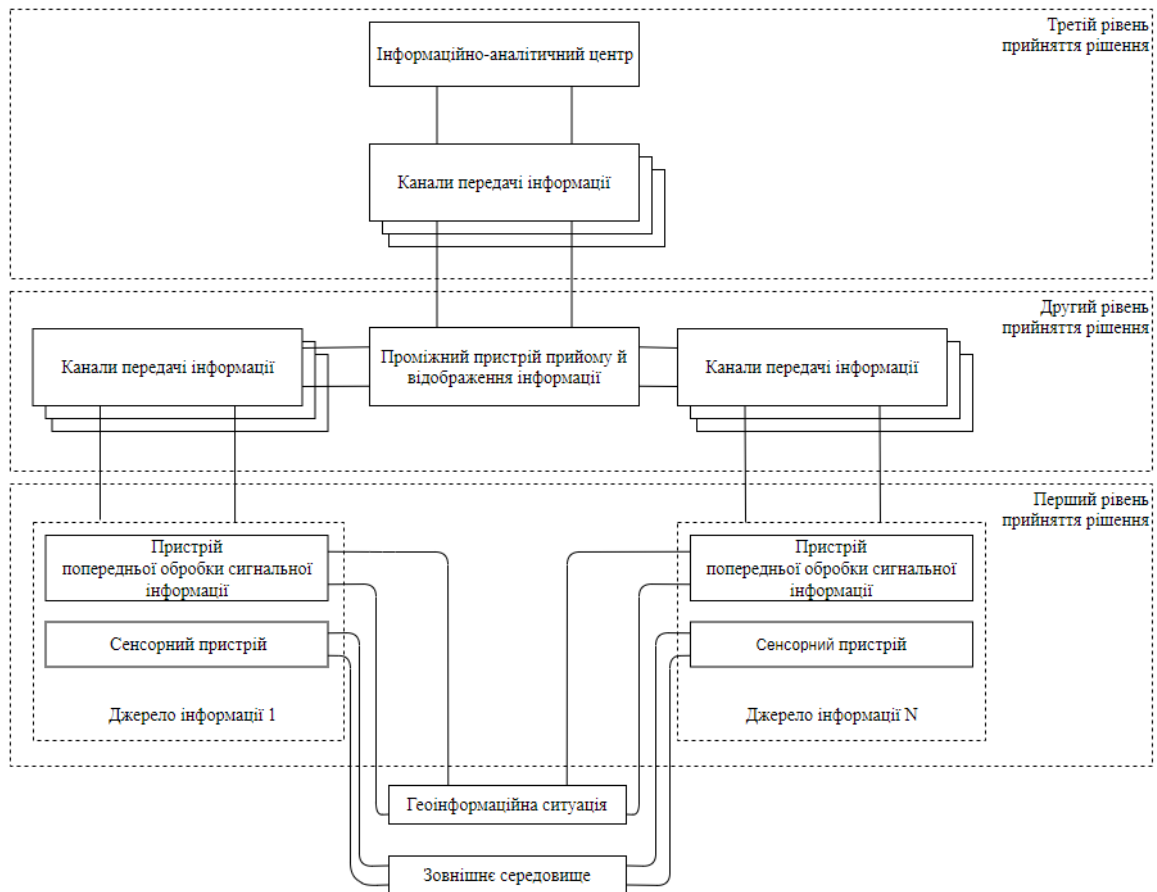


Рисунок 1.1 – Структурна схема типової СМТ у вигляді багаторівневої системи прийняття рішення в процесі виявлення та розпізнавання геоінформаційних ситуацій

Рівні обробки інформації та прийняття рішень визначаються:

- складністю і глибиною узагальнення задач, що вирішуються;
- кількістю і достовірністю наявної на кожному рівні інформації про зовнішні умови (зовнішнє середовище);
- максимально допустимим часом вироблення і прийняття рішення; швидкістю, завадостійкістю і безвідмовністю каналів передачі інформації;
- методами обробки інформації від мережі ДІ [1].

Перший рівень прийняття рішень, зазвичай, реалізується в ДІ. Наприклад, в системах охоронного моніторингу це є різні сенсорні пристрої (СП), які забезпечують фіксацію змін фізичних полів і перетворення їх в сигнальну інформацію.

У загальному випадку СП може складатися з приймальних пристроїв (ПП) і пристроїв попередньої обробки сигнальної інформації (ППО). Результатом прийнятого рішення є формалізоване повідомлення про виявлення та класифікацію елементів геоінформаційної ситуації за різними фізичними полями (оптичним, акустичним, сейсмічним, електромагнітним, інфрачервоним і т.д.).

При цьому кожен СП може працювати як в повністю автономному режимі, так і як комплексна система моніторингу (підвищення достовірності, якості та кількості інформації, а також ідентифікація елементів ГС, визначення її координат і різних параметрів).

Повідомлення від СП передаються по каналах передачі інформації на другий рівень. Кожен СП реалізує один з найбільш оптимальних даних зовнішнім умовам і стану середовища алгоритмів обробки інформації першого рівня по одному або декільком параметрам фізичних полів об'єктів спостереження (елементів ГС). Розташування групи СП (конфігурація) визначається вимогами, що пред'являються до СМТ, і задачами, які вона повинна вирішувати в інтересах моніторингу середовища і об'єктів спостереження, а також особливостями поширення сигналів від цих об'єктів; їх параметрами і особливостями середовища, в якому відбувається процес формування сигнальної інформації про елементи ГС.

У системах природного моніторингу перший рівень прийняття рішення реалізується за аналогічною схемою, тільки як джерело первинної інформації виступають не СУ, а системи збору даних дистанційного зондування поверхні Землі в оптичному, радіолокаційному, інфрачервоному та інших діапазонах радіохвиль. У таких системах інформація (знімки поверхні Землі) первинно обробляється. Результатами цієї обробки є первинно класифіковані об'єкти, наприклад, окремі осередки пожеж в лісовому масиві.

КПП призначені для передачі прийнятого рішення в строки, визначені вимогами до системи по завадостійкості і оперативності.

При цьому сумарний час видачі інформації про об'єкт спостереження в СМТ залежить від дистанції його первинного виявлення; параметрів об'єкта; кількості сенсорних пристроїв; часу прийняття рішення кожним СП; часу обробки інформації та прийняття рішення ППО; часу обробки інформації та прийняття рішення ІАЦ.

У даній роботі спеціальне дослідження впливу КПП на ефективність функціонування СМТ не проводилося, при цьому передбачалося, що їх реальна завадостійкість і ненадійність призводить до принципової нечіткості і неповноти вихідної для розпізнавання інформації геоінформаційних ситуацій на другому і третьому рівнях прийняття рішень.

ППВІ призначений для обробки інформації на другому рівні прийняття рішення. Інформація надходить у вигляді формалізованих повідомлень від декількох ДІ. Задачами, що вирішуються є:

- ідентифікація об'єктів спостереження (елементів ГС);
- класифікація, визначення координат і параметрів об'єктів;

- вироблення рекомендацій по режимах використання ДІ першого рівня; спільна обробка та управління ДІ;

- видача даних про об'єкти; відображення геоінформаційної ситуації на електронній карті.

ІАЦ вирішує задачу прийняття рішень третього рівня за інформацією, одержуваною від сукупності ДІ. Він призначений для: остаточного розпізнавання і класифікації геоінформаційних ситуацій, що спостерігаються; видачі вказівок вищим органам управління; управління і взаємодії різних елементів системи моніторингу (формування варіантів функціонування системи в залежності від розв'язуваних завдань і контексту зовнішніх умов). На ІАЦ лежить відповідальність за своєчасне прийняття рішення на видачу інформації у надзвичайних ситуаціях.

Структура і функціональна складність побудови перспективної автоматизованої системи розпізнавання геоінформаційних ситуацій в СМТ визначається такими загальними вимогами [4 – 6]:

- необхідністю вирішення задач автоматичного виявлення, розпізнавання і ототожнення елементів ГС;

- необхідністю доставки, отримання інформації в реальному часі;

- необхідністю врахування випадкових факторів, властивих як об'єктів розпізнавання, так і самої СМТ (перешкоди, збої і інше);

- високим ступенем автоматизації процесів збору, обробки, передачі інформації і представлення результатів оператору автоматизованого робочого місця (АРМ) для остаточного рішення;

- пред'явленням до основних характеристик СМТ жорстких вимог по достовірності і швидкодії в реальних умовах.

Якщо в своїй ранній постановці задача системи розпізнавання зводилася до інтерпретації вхідних даних в термінах невеликого числа класів (образів), заданих людиною, то створення інтегрованих систем розпізнавання, в яких задача сприйняття і інтерпретації інформації про зовнішнє середовище і завдання прийняття рішення і виконання певних дій по перетворенню цього середовища виявилися в значній мірі об'єднаними в рамках однієї системи, вимагало нової постановки задачі розпізнавання образів.

Специфічним у цій постановці є наступне:

- необхідність будувати комплексний опис зовнішнього світу на основі врахування значної апріорної інформації, а не обмеження задачі виділенням фіксованих ознак або вимірюванням окремих параметрів об'єктів;

- необхідність вирішувати задачу розпізнавання об'єктів з урахуванням певних просторових відносин між ними;
- необхідність аналізувати об'єкт розпізнавання, що включає в себе одночасно ряд довільних об'єктів (в класичній задачі системі для розпізнавання пред'являвся, як правило, один об'єкт);
- необхідність вирішувати не фіксовану задачу розпізнавання, а сукупність задач, поставлених динамікою зміни зовнішнього середовища.

Задача, яка вирішується системою, полягає в перетворенні вихідних даних про зовнішнє середовище на деякій мову, що дозволяє здійснювати такий опис об'єктів середовища, на якому можна було б здійснювати їх виділення, формування понять об'єктів з метою їх інтерпретації, виявлення відносин між об'єктами, що знаходяться в світі, і, нарешті, здійснення їх класифікації [7 – 10]. З огляду на динаміку зміни зовнішнього середовища, можна вважати задачу такої системи багатоцільовою.

Особливістю даного підходу є постановка задачі розпізнавання як задачі обробки знань [11 – 13]. Відповідно проблема розпізнавання може бути зведена до використання методів обробки знань. Одним з найбільш перспективних напрямків такого підходу до вирішення задач розпізнавання є використання технології експертних систем (ЕС) та геоінформаційних технології (ГІТ), спеціально розроблених для роботи з даними, що мають просторово-часову прив'язку, що зумовило їх швидке поширення і широке використання в багатьох галузях науки і техніки, в тому числі і в системах моніторингу.

Геоінформаційні системи (ГІС) є новим типом інтегрованих комп'ютерних систем, що з'явилися в кінці ХХ століття. Зараз це найбільш перспективна і універсальна система управління геоданими [3, 5, 14].

Найбільш загальне визначення для ГІС – це автоматизована інформаційна система, призначена для обробки даних, основою інтеграції яких є географічна інформація. Основна її перевага перед звичайними СУБД в тому, що за своєю структурою вона є СУБД, але з географічною прив'язкою даних в дво- або тривимірному просторі до певної точки на місцевості.

Крім того, ГІС має вбудовану систему просторового аналізу, якої немає в звичайній СУБД. Елементи географічної (електронної) карти забезпечуються семантикою, яка може бути проаналізована. Взаємодія між цими елементами також може бути проаналізована. Сучасні ГІС дозволяють аналізувати не тільки векторні дані, але і растрові, і текстові.

На сьогоднішній день останньою є технологія перегляду як географічної (графічної) інформації, так і прив'язаної до неї атрибутивної інформації. Для забезпечення даної можливості були розроблені спеціальні векторні формати. Система повинна

забезпечувати не тільки високу швидкість перетворення вихідної графічної інформації, а й високошвидкісну обробку атрибутивної інформації, що зберігається в базі даних (БД).

Для систем моніторингу територій розмір контрольованої зони спостереження (зони відповідальності) може варіюватися в широких межах від сотень до декількох тисяч кв. км, що, в свою чергу, передбачає використання різних за складом і кількістю джерел інформації.

Однак, треба зазначити, що ці значення можуть бути забезпечені тільки за умови, що образи сигналів отримані при відповідній адаптації джерел інформації до умов експлуатації. Як показує досвід практичного використання різних ДІ, ці показники значно залежать від кліматичних, топографічних, геологічних та акустичних особливостей території в місці установки ДІ і умов, в яких датчикам доводиться функціонувати. Тому при зміні умов експлуатації, наприклад, погодні умови, ймовірність розпізнавання, як правило, знижується до неприпустимо низьких значень, при цьому зростає ймовірність помилкових спрацьовувань.

Проте, аналіз сучасних вітчизняних і зарубіжних СМТ показує, що в них слабо автоматизований рівень обробки потоку повідомлень від ДІ: як правило, мало враховуються умови навколишнього середовища, наприклад, метеорологічні умови, взаємозв'язок між активністю природних факторів і часу доби, а також вплив існуючих антропогенних і техногенних факторів. Не враховується і людський фактор, коли оператору доводиться реєструвати велику кількість повідомлень, які при цьому можуть суперечити один одному. Процес прийняття рішень в цих умовах суб'єктивний і ґрунтується, найчастіше, на особистому досвіді і інтуїції. Все це багато в чому визначає максимальну ефективність системи, тобто ту максимальну кількість повідомлень, яку може обробити система.

Іншими словами, для підвищення ефективності СМТ необхідна організація обробки повідомлень від ДІ з урахуванням всієї наявної на момент прийняття рішення оператором апріорної інформації [15, 16].

1.2 Особливості організації обробки інформації в системах моніторингу територій

Незважаючи на велику кількість публікацій, присвячених проблемам розпізнавання образів, слід зазначити, що всі вони практично поділяються на дві основні частини: опис

методів отримання ознакового опису об'єкта розпізнавання і опис методів синтезу системи прийняття класифікаційних рішень. При цьому, зазвичай, розглядається набір конкретних алгоритмів попередньої обробки або прийняття рішень з відповідним аналізом їх властивостей.

Однак, рішення конкретних прикладних задач найчастіше не може бути виконано з використанням будь-яких стандартних рецептів. Це пов'язано з тим, що при вирішенні прикладних задач, зокрема побудови багаторівневих систем розпізнавання геоінформаційних ситуацій, мають місце особливості, пов'язані зі специфікою вирішуваних задач. Розглянемо ці особливості докладно і оцінімо їх вплив на спосіб вирішення задач синтезу систем, що розпізнають [5, 7, 14, 17, 18].

Будемо розглядати задачу розпізнавання Z у вигляді сукупності задач

$$Z_i = \{Z_1, Z_{2i}, \dots, Z_i, \dots, Z_n\},$$

де $i = \overline{1, n}$ – розмір множини Z , яка визначається числом рівнів прийняття рішень у системі моніторингу.

Задача розпізнавання на кожному рівні може бути представлена у вигляді:

$$Z_i = \{M_i, R_i, P_i, C_i, Q_i\},$$

де для $i = 1$ маємо:

M_1 – множина сигналів, що породжуються об'єктами, які розпізнаються;

R_1 – моделі середовища розповсюдження;

P_1 – ознаки об'єкта розпізнавання;

C_1 – структура множини класів;

Q_i – структура правил для прийняття рішення.

Для першого рівня розпізнавання задача формування множини M сигналів, породжуваних об'єктами, що розпізнаються, відноситься скоріше до проблем побудови інформаційно-вимірювальних систем, що вирішують задачу отримання сигнального опису деякого реального фізичного об'єкта.

Очевидно, що в основі такої системи лежить використання різного роду джерел інформації, що перетворюють фізичні поля об'єктів розпізнавання на відповідні електричні сигнали і далі в цифрову інформацію, яка надходить на вхід комп'ютерної системи.

Однак, при дослідженні реальних сигналів доводиться враховувати важливий фактор, пов'язаний з характеристикою середовища поширення реєстрованих сигналів.

Таким чином, маємо наступну модель формування сигналу, що підлягає розпізнаванню в системі моніторингу (рис. 1.2).

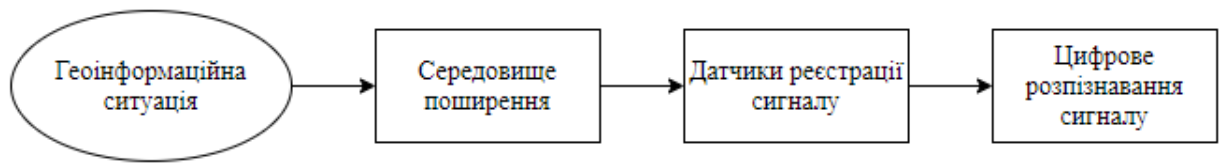


Рисунок 1.2 – Модель формування сигналу для геоінформаційної ситуації

Проблема врахування впливу середовища поширення є досить складною для вирішення більшості практичних задач.

У даній роботі цій задачі приділяється значна увага. Зокрема, розглянуто проблеми врахування впливу зовнішніх умов для задач розпізнавання геоінформаційних ситуацій по їх інформаційним ознаками.

У таких задачах вплив зовнішніх умов проявляється зміною зони нечіткості інформаційних ознак, що, в свою чергу, призводить до зміни ефективності систем, що розпізнають. Прояв впливу «пам'яті» зовнішніх умов має складний характер. Його, як правило, можна враховувати через систему математичних і логічних обмежень і припущень, що мають смислову єдність, і названі в роботі контекстом зовнішніх умов прийняття рішення.

Саме тут і виникає проблема обліку контексту зовнішніх умов при розпізнаванні геоінформаційних ситуацій за інформацією від різних її джерел (датчиків і/або систем розпізнавання ГС, що мають спільну зону відповідальності) при автоматичному створенні, наприклад, електронної карти оперативної обстановки [15, 16].

На даному рівні маємо модель формування вихідної інформації, по якій здійснюється розпізнавання геоінформаційних ситуацій (рис. 1.3).



Рисунок 1.3 – Модель формування інформації

Особливістю розглянутої проблеми є наявність апріорної невизначеності, пов'язаної з характеристиками каналу поширення сигналу. Тут «сигнал» вживається як поняття, що характеризує вихідну інформацію для розпізнавання схем будь-якого рівня.

У роботі розглянуті підходи зведення задачі до проблеми подолання параметричної апріорної невизначеності, особливості, пов'язані з формуванням ознак опису розпізнаваних сигналів. Розглядається підхід до побудови простору ознак на основі моделі складної інформаційної структури ГС. Під цим будемо розуміти деякий сигнал S за допомогою композиції різнорідних ознакових описів P_i ($i = \overline{1, m}$):

$$P = \{P_1 * P_2 * \dots * P_m\}. \quad (1.2)$$

Конкретний опис P_i є або вихідним відліком сигналу, або результатом різних перетворень [19, 20]. Вибір конкретних видів опису може визначатися на основі деякої моделі формування простору ознак або з використанням певних інтуїтивних міркувань.

На жаль, до теперішнього часу не існує закінченої теоретичної бази, що дозволяє сформулювати оптимальний опис для довільного типу сигналів. При цьому вирішальну роль при формуванні описів найчастіше мають знання людини-експерта, яка досить глибоко розуміє сутність розв'язуваної задачі.

Загальним підходом при вирішенні цих задач є використання певної феноменологічної моделі формування вихідного опису об'єкта, декомпозиції її на окремі підмоделі і формування простору ознак стосовно певної підмоделі. При цьому часто

можна впорядкувати ці підмоделі у вигляді деякої ієрархічної структури і відповідно побудувати багаторівневу систему ознакового опису.

Цей підхід був використаний при розробці технології нейромережевого розпізнавання геоінформаційних ситуацій в системах моніторингу.

Розглянемо далі поняття структури множини класів. У роботі широко використовується концепція «складних класів» [11, 21 – 23]. Під «складним» класом будемо розуміти клас, для якого множина класу X_i у просторі X може бути представлено у вигляді:

$$X_i = \cup_{j=1}^{L_i} X_j^i,$$

де X_j^i – деякі множини, що володіють певним ступенем «компактності» таким чином, що міра близькості елементів всередині X_j^i – значно менше, ніж міра близькості між окремими групами X_j^i .

Таке розбиття породжує всередині класів кілька підкласів, що мають зазвичай досить просту просторову структуру (типу гіперсфери, гіперелліпсоїдів, гіперпаралелепіпедів). Відповідно, кожному підкласу може бути зіставлений якийсь еталонний елемент (наприклад, середнє по підкласу). Таким чином, клас може бути в найпростішому випадку заданий набором еталонів M_j^i . Відповідно, саме розпізнавання може бути здійснено шляхом обчислення, наприклад, відстані від вектором x , який розпізнається до еталонів:

$$d_j^i = \|x - M_j^i\|, i = \overline{1, M}, j = \overline{1, L_i}, \quad (1.3)$$

де M – кількість класів;

L_i – кількість підкласів у i -му класі.

Прийняття рішення щодо включення до класу може бути виконано шляхом знаходження мінімальної відстані.

Розглянута концепція багаторівневого ознакового опису та концепція «складних класів» призводить до необхідності розглядати конструкції прийняття рішення у вигляді багаторівневої конструкції, в основу якої використовуються системи «окремих класифікаторів», з відповідними під моделями геоінформаційних ситуацій.

Система окремих класифікаторів визначає простір рішення окремих класифікаторів. Відповідно загальна система прийняття рішення будується шляхом об'єднання рішень окремих класифікаторів.

На підставі розглянутих вище концепцій, що лежать в основі побудови складних систем розпізнавання геоінформаційних ситуацій, пропонується загальний підхід до побудови систем розпізнавання геоінформаційних ситуацій, основними компонентами якого є:

- аналіз моделей формування повідомлень на рівні первинного джерела;
- аналіз зовнішніх умов прийняття рішень про геоінформаційну ситуацію (її елементи);
- формування багаторівневої системи ознакового опису геоінформаційної ситуації;
- аналіз структури класів геоінформаційної ситуації;
- формування багаторівневої системи прийняття рішення.

Зупинимося детальніше на проблемі апіорної невизначеності. Основною особливістю процесу розробки складних систем розпізнавання геоінформаційних ситуацій на всіх рівнях є наявність суттєвої апіорної невизначеності. Пропонується розглядати зазначену апіорну невизначеність в широкому сенсі, розуміючи при цьому такі види невизначеності, як:

- параметрична;
- непараметрична;
- структурна.

Під параметричною невизначеністю розуміється завжди апіорна невизначеність у випадку, коли математичні моделі, які описують формування сигналу, умови його поширення, функціонально визначені з точністю до кінцевого набору параметрів. Існують добре розроблені підходи для подолання (усунення) такого типу невизначеності [24, 25].

Непараметрична апіорна невизначеність фактично зводиться до відсутності апіорії математичної моделі сигналу.

Облік такого роду апіорної невизначеності вимагає вирішення завдань побудови моделей сигналів, їх ідентифікації та розробки методів усунення заважають (апіорі невідомих) факторів, що входять в ці моделі. Задачі, пов'язані з невизначеністю такого типу, також розглядаються в дипломній роботі. Як основні підходи до усунення такого типу невизначеності досліджені адаптивні методи налаштування функцій приналежності нечітких інформаційних ознак геоінформаційних ситуацій.

Під структурною невизначеністю в даній роботі буде розумітися невизначеність структури класів сигналів, які розпізнаються. Сюди входять розглянуті вище проблеми з невизначеністю структури множини класів. Різні методи побудови вирішальних функцій фактично є засобами подолання структурної апріорної невизначеності.

Запропоновані в даній роботі концепції «складних класів», концепції багаторівневих систем прийняття рішень дозволяють отримати ефективні підходи до подолання невизначеності розглянутого типу.

Важливою компонентою процесу подолання апріорної невизначеності є ідентифікація її типу. Ідентифікація типу апріорної невизначеності є важко формалізованою задачею, вона вимагає істотного розуміння всієї структури розв'язуваної задачі розпізнавання, включаючи фізичні явища, що лежать в основі формування повідомлення про об'єкт моніторингу, методи отримання опису та методи побудови системи прийняття рішень.

Запропоновані в цій роботі рішення засновані, фактично, на аналізі знань, пов'язаних з особливостями завдань розпізнавання геоінформаційних ситуацій.

Під геоінформаційною ситуацією слід розуміти частину географічного простору, в якому існують просторово рознесені лінійні (елементи ГС лінійного типу) і площинні (елементи ГС площинного типу) об'єкти, і інформацію про стан цих об'єктів і характеристики даного простору в конкретний момент часу t .



Рисунок 1.4 – Приклад геоінформаційної ситуації

Така інформація накопичується і обробляється в інформаційно-аналітичних центрах аналізу геоінформаційних ситуацій, залежить від системних факторів і комплексу поточних зовнішніх умов і служить основою для прийняття різних управлінських рішень [26].

1.3 Аналіз впливу зовнішніх умов на джерела інформації, що використовуються для розпізнавання образів геоінформаційних ситуацій в системах моніторингу територій

Під зоною моніторингу розуміється частина географічного простору, в якому засіб, комплекс засобів, система може виконати свою цільову функцію з заданою якістю. Розміри цієї зони визначаються, виходячи з обмежень фізичних носіїв інформації про події, або задаються евристично на етапі планування, виходячи з глобальної мети і задач системи.

Обробку інформації в системах, що використовують зональний принцип організації геоінформаційного простору, можна інтерпретувати у вигляді багаторівневої моделі.

Розробка узагальненої моделі обробки інформації в СМТ виконується в два етапи.

На першому етапі виконується формальний опис процесів отримання, передачі та обробки інформації. На другому – рівень інформаційно-аналітичного центру – з урахуванням виявлених обмежень описуються структурні особливості взаємодії видів інформації, які слід враховувати в процесі розпізнавання та подання на електронній карті елементів ГС (територіальних об'єктів).

У загальному випадку, як вже зазначалося раніше, процес отримання, передачі та обробки інформації в СМТ реалізується на двох рівнях:

– отримання інформації від ДІ у вигляді повідомлень, які передаються за допомогою каналів (радіоканалів) зв'язку у результаті обробки інформації;

– обробка повідомлень від ДІ за допомогою ІАЦ на алгоритмічній мові за допомогою комплексу обчислювальних пристроїв та людини-оператора, що приймає логічні рішення. Включає у себе:

1) обробку прийнятих формалізованих повідомлень від ДІ відповідно до визначених пріоритетів їх важливості;

2) формування вихідних повідомлень за результатами алгоритмічної обробки і прийнятих оператором рішень;

3) обробку службових команд і запитів операторів;

4) відображення розпізнаних геоінформаційних ситуацій на електронній карті або в формулярах нормативно-технічної документації.

Узагальнена модель процесу обробки інформації у системі моніторингу територій в припущенні ідеального зв'язку представлена на рисунку 1.5.

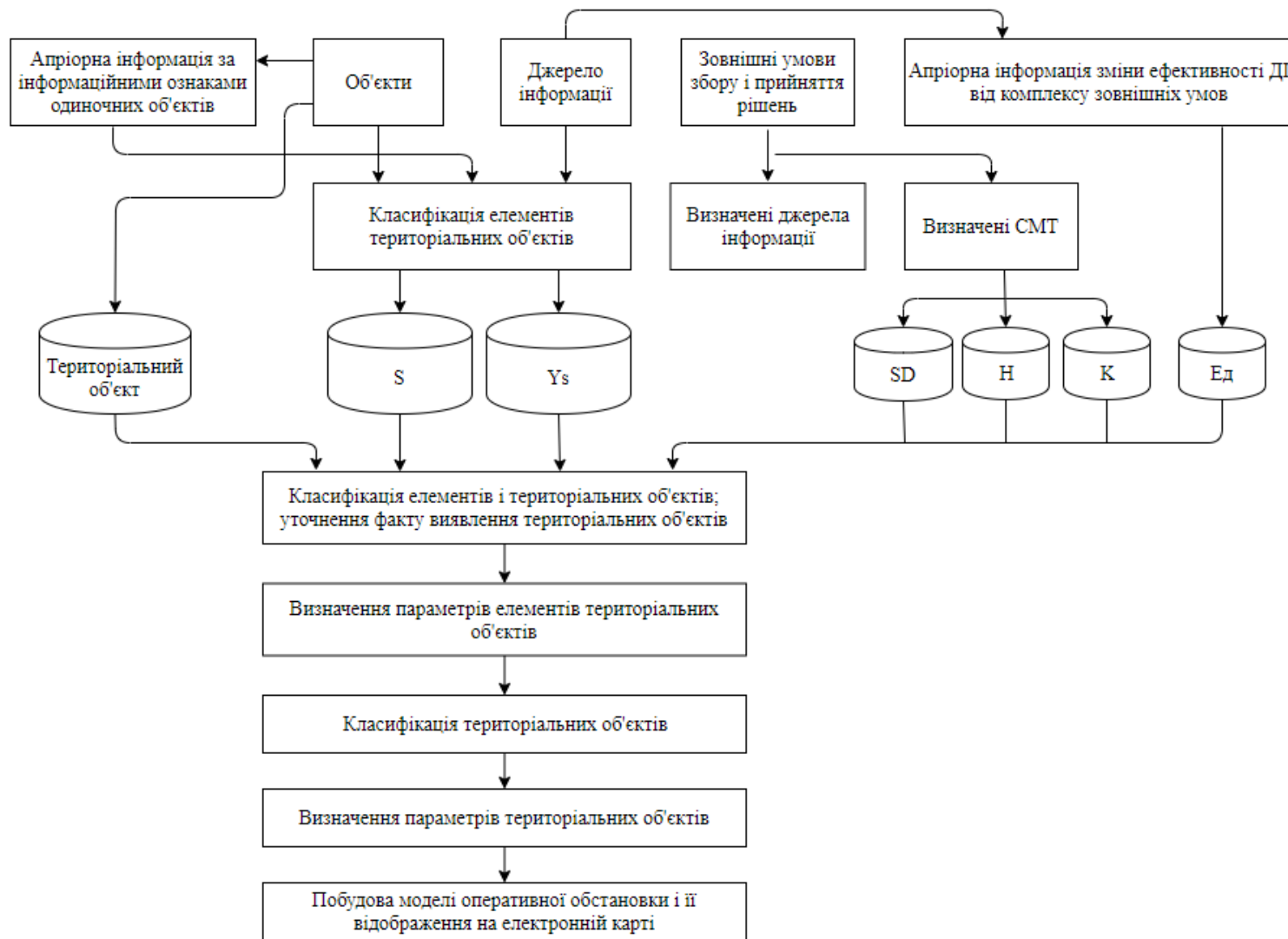


Рисунок 1.5 – Узагальнена модель процесу обробки інформації в СМТ в припущенні ідеального каналу зв'язку

На рисунку 1.5 представлені наступні бази даних:

- TO – БД територіальних об'єктів;
- S – БД повідомлень від мережі джерел інформації;
- Y_s – БД зовнішніх умов, які визначені вирішальною схемою для кожного повідомлення;
- SD – БД з інформацією про системи джерел інформації (структура, склад, координати місць установки);
- H – матриця висот та інші географічні особливості місцевості у зоні моніторингу СМТ;
- K – БД кліматичних умов, прив'язаних до часу обробки інформації через СМТ;
- E_d – оцінка ефективності джерела інформації від комплексу зовнішніх умов.

Дана узагальнена модель процесу обробки інформації в СМТ показує потенційно можливий склад вектора контексту зовнішніх умов, місце обліку його впливу і дозволяє перейти до розробки технології нейромережевого розпізнавання територіальних об'єктів з урахуванням контексту зовнішніх умов.

1.4 Постановка завдання дослідження

Повідомлення від одиночного ДІ можна розглядати як вихід дискретного каналу передачі, на вхід якого надходить достовірна інформація від гіпотетичного джерела дискретних даних, який формує повідомлення з алфавітом $A = \{a_0, a_1, a_2, \dots, a_n\}$. Кожна буква алфавіту відповідає виду елемента територіального об'єкта. Вплив контексту зовнішніх умов може інтерпретуватися як вплив випадкових перешкод, характеристики яких залежать від особливостей умов збору первинних даних і алгоритмів первинної схеми виявлення/класифікації [6, 15, 26].

Спотворення в каналі призводять до трансформацій інформаційних символів і появи помилкових символів за рахунок ехо ефекту. Крім того, алфавіт повідомлень на виході каналу розширюється діагностичними символами $\{b_1, b_2, \dots, b_m\}$, які несуть інформацію про повну групі подій, пов'язаної з класифікацією. Дані події генеруються вирішальною схемою виявлення/класифікації територіального об'єкта.

До них відносяться: відмова від класифікації, помилка у виявленні букви алфавіту, зниження ймовірного порогу і ряд інших. Наявність діагностичного символу в тексті

повідомлення свідчить про факт виявлення інформаційного символу на вході каналу, але відсутності достовірних даних про його вид.

Кожне ДІ передає безперервну послідовність символів, що належать вихідному алфавіту (символ a_0 використовується як індикатор відсутності елементів ГС). Характеристики ДІ паспортизовані для різних умов його функціонування.

У набір характеристик включається гарантована дальність виявлення, умовні ймовірності трансформацій символів в робочій зоні дистанцій, а також ймовірності дублювання символів, що виникають за рахунок різних ехо ефектів.

Дискретний канал, відповідний окремому ДІ, описується двома характеристиками: матрицею трансформації символів і ймовірністю їх дублювання. Як показано в [20, 25, 27], дублювання суміжних символів незалежно і підпорядковується біноміальному закону. У результаті дублювання довжина вихідного слова каналу може бути більше довжини вхідного слова.

Територіальні об'єкти або їх групи можуть об'єднуватися в потоки. Послідовність об'єктів в потоці утворює інформаційний образ потоку. Припустимо, що при передачі по каналу, ознака кінця територіального об'єкта не може бути спотворена та ймовірність його випадкового породження близька до нуля. Отриманий від ДІ потік повідомлень аналізується в ІАЦ по мірі надходження інформації від цих ДІ в характерних точках вимірювання, які відповідають впорядкованій по зростанню послідовності довжин еталонів.

Потік при обробці, зазвичай, ділиться на територіальні об'єкти синхронізуючими ознаками. Відлік часових позицій починається з моменту виявлення чергової ознаки синхронізації. Обробка виконується по всіх тимчасовим позиціям аж до надходження наступної ознаки синхронізації. Аналізований образ укладений між двома суміжними ознаками синхронізації і завжди починається з символу відмінного від a_0 .

На кожній тимчасовій позиції виконується порівняння отриманого образу з усіма еталонами. Оскільки в результаті трансформацій першого або останнього символу до символу a_0 , довжина образу може бути менше еталонної, то для точок вимірювання допускається тимчасова толерантність до деякого зменшення довжини образу. При толерантних викривлення обробка виконується на тимчасовій довжині образу, який надійшов.

Таким чином, контекст зовнішніх умов в загальному випадку може призводити до виникнення всіх видів помилок у повідомленнях ДІ аж до наявності вставок, заміщень і винятків елементів алфавіту, що призводить до нечітких інформаційних ознак ТО.

Все це призводить до необхідності застосування нечітких штучних нейронних мереж для вирішення задачі розпізнавання ТО за інформаційними ознаками з урахуванням впливу контексту зовнішніх умов.

Тому науковою задачею є дослідження методів розпізнавання геоінформаційних ситуацій (лінійних і площинних об'єктів електронної карти) по нечіткій і неповній вихідній інформації на основі штучних нейронних мереж, рішення якої визначило послідовність та зміст дипломного дослідження.

2 МЕТОД РОЗПІЗНАВАННЯ ОБРАЗІВ ГЕОНФОРМАЦІЙНИХ СИТУАЦІЙ НА ОСНОВІ НЕЧІТКОЇ ШТУЧНОЇ НЕЙРОННОЇ МЕРЕЖІ В СИСТЕМАХ МОНІТОРИНГУ ТЕРИТОРІЙ

2.1 Узагальнена модель геоінформаційних ситуацій на основі певних інформаційних ознак у системах моніторингу територій

Складний ієрархічно організований територіальний об'єкт спостереження (геоінформаційна ситуація) утворюється за допомогою об'єднання елементів різних рівнів ієрархії [18, 28, 29]. Окремий елемент складається з набору елементів попереднього рівня i , в свою чергу, входить як складова частина в елемент більш високого рівня, тобто об'єкт має рекурсивну структуру.

Кожен елемент i -го рівня може знаходитися в одному з станів кінцевої множини, при цьому число станів елементів різних рівнів в загальному випадку різне. Стан об'єкта спостереження визначається станом елементів всіх рівнів, об'єднанням яких він є.

Саме цей стан i є предметом розпізнавання. Очевидна аналогія такого ієрархічно організованого об'єкта спостереження з текстом, що складається з букв, слів, речень, де під літерою розуміється одиночний об'єкт (елемент ГС), під словом сукупність елементів ГС певної структури і складу і т.д.

Число можливих станів ГС дуже велике, особливо з урахуванням впливу комплексу зовнішніх умов на варіанти можливих результатів на виході каналу передачі даних, та її розпізнавання шляхом аналізу лише верхнього рівня ієрархії не є можливим. Тому розпізнавання ГС ведеться шляхом розпізнавання окремих її елементів. Таким чином, ієрархія системи розпізнавання практично визначається структурою ГС, яка розпізнається та інформацією про неї, доступною на кожному рівні. Щоб описати роботу системи, розглянемо розпізнавання елемента деякого рівня, а потім рекурсивно поширимо результати на всі рівні системи. Перш за все, визначимо дані, доступні при розпізнаванні елемента E .

По-перше, це результати розпізнавання складових частин E – його елементів, e_1, \dots, e_k , по відношенню до яких E виступає як об'єкт високого рівня. Цей потік даних назвемо висхідним, оскільки він йде від елементів нижчих рівнів до верхніх рівнів. Прикладом висхідного потоку при розпізнаванні сукупності елементів ГС певної

структури і складу (слів), є результати розпізнавання одиночних об'єктів (одиночних букв).

По-друге, це дані спостережень над елементом E , що розглядаються як певна цілісність, а також як апіорна інформація про нього. Ці дані не залежать від елементів інших рівнів, тому цей потік даних можна назвати горизонтальним. Наприклад, структура, склад певної сукупності елементів ГС (зображення слова), які використовуються в глобальному алгоритмі розпізнавання, є типовим спостереженням на рівні слів.

Прикладом апіорної інформації на рівні слів є обмежений словник банку формалізованих описів інформаційних ознак геоінформаційних ситуацій і різні частоти слів у словнику цього банку [31, 32].

По-третє, це контекстна інформація, яка надходить з верхніх рівнів в низхідному потоці даних. Прикладами такої інформації, що надходить на рівень слів з рівня речень (повного опису геоінформаційної ситуації), є різні частоти одного і того ж слова в залежності від його місця в реченні (в структурі ГС), інформація про заборонені в поточному контексті поєднання слів і т. д. Очевидна складність використання низхідної інформації полягає в тому, що вона виникає лише тоді, коли з'являються результати розпізнавання вищих рівнів. У той самий час, чим вище рівень ієрархії елемента, тим важче розпізнати його стан: саме через це задача розпізнавання стану ГС (елемента найвищого рівня) вирішується шляхом порівневого розпізнавання його елементів. Перераховані джерела і потоки інформації представлені на рисунку 2.1.

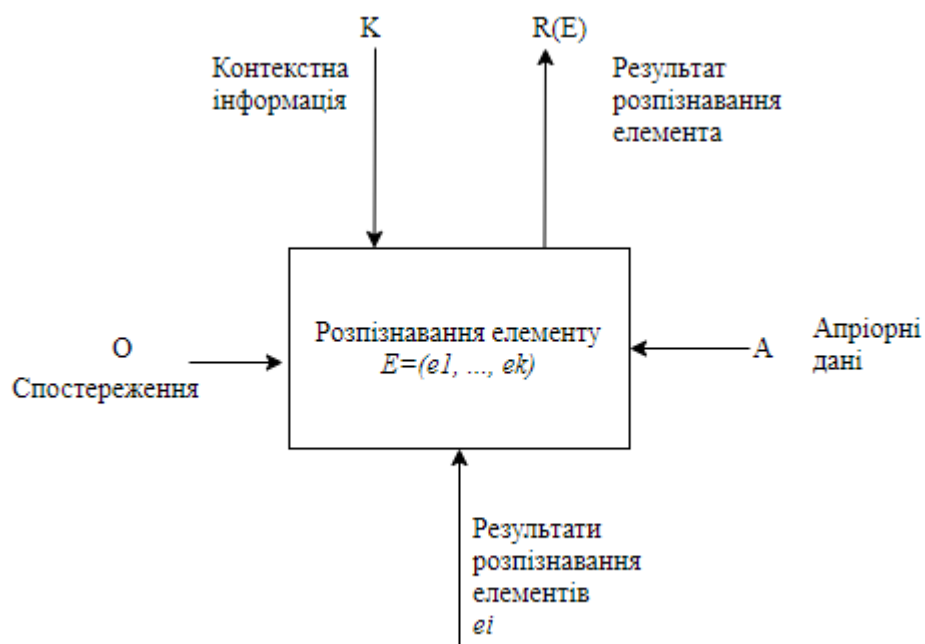


Рисунок 2.1 – Інформаційні потоки при розпізнаванні елемента об'єкта дослідження

На основі використання доступної інформації потрібно визначити стан E , тобто віднести його до певного класу. Як вже зазначалося, жорстке рішення тут неприйнятне (якщо тільки E не є елементом вищого рівня), оскільки це призводить до втрати стійкості системи розпізнавання. Тому всі рішення ґрунтуються на м'яких рішеннях та ймовірнісній формі. При цьому очевидно, що якість і складність алгоритмів розпізнавання ГС багато в чому визначаються якістю інформації, що надходить від джерел інформації.

Може здатися, що спотворення інформації на будь-якому рівні повинно призводити до фатальних спотворень на більш високих рівнях ієрархії.

Однак, якщо порівняти розпізнавання ГС зі сприйняттям тексту людиною, оскільки і текст, і образ ГС містять інформацію різних рівнів ієрархії, то це твердження не є коректним. Експерименти показують, що людина може розпізнавати текст і розуміти його сенс при спотворенні або відсутності значної частини інформації на одному або декількох рівнях.

Розглянемо результати експериментів по сприйняттю людиною слів за її елементами.

Описувані експерименти проводилися при дослідженні усного мовлення, проте, вони показові і для задачі розпізнавання ГС. Результати показують, що розбірливість слів (відношення числа правильно сприйнятих слів до загального їх числа) однозначно пов'язана з ймовірністю правильного впізнання фонем (елемента слова). При ймовірності розпізнавання окремих фонем $P_{\phi} = 0.6$ ймовірність правильного розпізнавання на рівні слів виявляється рівною 0.8, тобто істотно більшою, ніж на фонемному рівні [33].

У той самий час, якщо виходити із гіпотези незалежності елементів слова, то ймовірність правильного розпізнавання п'ятибуквеного слова склала б усього $(0.6)^5 = 0.078$.

Отже, гарна якість розпізнавання слів досяжна і при поганій якості розпізнавання символів.

Ідея, що лежить в основі формалізації причинно-наслідкових зв'язків між змінними, полягає в описі цих зв'язків на природній мові із застосуванням теорії нечітких множин і лінгвістичних змінних. Мета цього розділу полягає у введенні основних формалізмів, необхідних для створення нечіткої бази знань, що є носієм експертної інформації і дозволяє здійснити автоматичне розпізнавання одиночних і групових об'єктів за інформацією низької якості. Формалізований опис ГС на рівні їх інформаційних ознак виконаємо в два етапи по мірі зростання його складності для рухомих одиночних і групових об'єктів, які можуть інтерпретуватися як лінійні, або площинні територіальні об'єкти електронної карти.

Нехай рухомий одиночний об'єкт y може бути охарактеризований сукупністю інформаційних ознак x_1, x_2, \dots, x_n , тоді його формальне відображення буде:

$$y = f_y(x_1, x_2, \dots, x_n), \quad (2.1)$$

де y – вихідна змінна, яка характеризує тип рухомого об'єкту, присвоєний вирішальною схемою ДІ;

x_1, x_2, \dots, x_n – інформаційні ознаки, які є апіорними оцінками параметрів об'єкту, що рухається і комплексу умов його виявлення розпізнавання, визначені за допомогою ДІ та СМТ, відповідно (рис. 1.5).

Змінні x_1, x_2, \dots, x_n можуть бути кількісними та якісними.

Параметрами кількісних змінних можуть бути: швидкість рухомого об'єкта [0.40] км/год, температура повітря [20]°C, швидкість вітру [15] м/с та інші змінні, легко вимірювані в прийнятих для них кількісних шкалах.

Прикладом змінної, для якої не існує природної кількісної шкали, є ступінь відповідності поточних умов експлуатації сигнального пристрою системи моніторингу до якості дозволених умов в його паспорті, яка може оцінюватися якісними термами (низька, середня, висока) або вимірюватися в штучних шкалах, наприклад 5-бальною, 10-бальною, ..., 100-бальною.

Для кількісних змінних передбачаються відомими області зміни:

$$U_i = [\underline{x}_i, \overline{x}_i], i = \overline{1, n}, \quad (2.2)$$

де \underline{x}_i (\overline{x}_i) – нижнє (верхнє) значення вхідних змінних x_i , $i = \overline{1, n}$.

Для якісних змінних $x_1 \div x_n$ передбачається, що відомі множини всіх можливих значень:

$$U_i = \{v_i^1, v_i^2, \dots, v_i^{q_i}\}, i = \overline{1, n}, \quad (2.3)$$

де v_i^1 ($v_i^{q_i}$) – бальна оцінка, відповідна до найменшого (найбільшого) значення вхідної змінної x_i ;

q_i ($i = \overline{1, n}$) – потужність множини (2.3).

У випадку визначення частини змінних $x_1 \div x_n$ у вигляді лінгвістичних змінних x_i , $i = \overline{1, n}$ будемо використовувати якісні терми із терм множини.

$A_i = \{a_i^1, a_i^2, \dots, a_i^{l_i}\}$ – терм-множина змінної $x_i, i = \overline{1, n}$, де a_i^p – p -й лінгвістичний терм змінної $x_i, p = \overline{1, l_i}, i = \overline{1, n}$.

Потужності терм-множин $A_i, i = \overline{1, n}$ у загальному випадку можуть бути різними. Назви окремих термів $a_i^1, a_i^2, \dots, a_i^{l_i}$ також можуть відрізнятися для різних лінгвістичних змінних $x_i, i = \overline{1, n}$. Лінгвістичний терм $a_i^p \in A_i$ будемо розглядати як нечітку множину, задану на універсальній множині U_i . У випадку кількісних змінних $x_i, i = \overline{1, n}$, нечітка множина a_i^p задається відношенням:

$$a_i^p = \int_{x_i}^{\overline{x_i}} \mu^{a_i^p}(x_i)/x_i, \quad (2.4)$$

де $\mu^{a_i^p}(x_i)$ – функція приналежності значення вхідної змінної $x_i \in [x_i, \overline{x_i}]$ терму $a_i^p \in A_i, p = \overline{1, l_i}, i = \overline{1, n}$.

У випадку якісних змінних $x_i, i = \overline{1, n}$ нечітка множина a_i^p задається як:

$$a_i^p = \sum_{k=1}^{q_i} \mu^{a_i^p}(v_i^k)/\mu^{a_i^p}, \quad (2.5)$$

де $\mu^{a_i^p}(v_i^k)$ – ступінь приналежності елементу $v_i^k \in U_i$, терму $a_i^p \in A_i, p = \overline{1, l_i}, i = \overline{1, n}, k = \overline{1, q_i}; U_i$ задається відношенням (2.3).

Зауважимо, що в співвідношеннях (2.4) – (2.5) знаки інтеграла і суми позначають об'єднання пар $\mu(u)/u$. Даний етап побудови моделі, на якому визначаються лінгвістичні оцінки змінних та необхідні для їх формалізації функції приналежності, є етапом фаззифікації змінних [34 – 36].

Під груповим об'єктом (ГО) розуміється сукупність однотипних і/або різнотипних одиночних об'єктів, що володіють певним порядком, яку будемо представляти кортежем виду:

$$ГО = \langle B, K, T \rangle, \quad (2.6)$$

де $K = (X, Y)$ – координати місцезнаходження ГО;

T – поточне значення часу, зв'язане з конкретним ГО;

$B = \langle S, C, P \rangle$ – вид групового об'єкту, заданий структурою (S), складом (C) та параметрами руху (P) ГО.

$$S = \langle b, R \rangle,$$

де $b = b_{11} + b_{12} + \dots + b_{1i} + \dots + b_{1n} + b_{21} + b_{22} + \dots + b_{2j} + \dots + b_{2m} + \dots$ – набір b_{ij} видів одиночних об'єктів руху, які рухаються у певному порядку (R);

$R = \psi(Z, T)$ – послідовність (залежить від розв'язуваної ГО задачі (3) та часу її виконання).

Склад (C) характеризується сумарною кількістю наборів видів одиночних об'єктів у групі.

Під параметром ГО будемо розуміти кортеж виду:

$$\Pi = \langle L, V, H \rangle, \quad (2.7)$$

$$\begin{aligned} L &= l_{11} + l_{(11)-(12)} + l_{12} + l_{(12)-(13)} + \dots = \\ &= \sum_{i=1}^n l_{1i} + \sum_{i=1}^{n-1} l_{1i-1(i+1)} + \sum_{j=1}^m l_{2j} + \sum_{j=1}^{m-1} l_{2j-1(j+1)} \end{aligned}$$

де l_{ij} – лінійний розмір одиночного об'єкта;

$l_{(ij)-(ij+1)}$ – відстань між сусідніми об'єктами у групі;

L – лінійний розмір ГО.

Виходячи з аналізу формалізованого опису ГО видно, що найбільш важливими інформативними ознаками при вирішенні задачі розпізнавання (ототожнення) ГО, в порядку убавання їх важливості, є S , C і параметри руху.

З огляду на наявність зон невизначеності значень інформаційних ознак, різноманіття умов, в яких знаходяться об'єкти розпізнавання, значення інформаційних ознак рухомих об'єктів мають нечітку числову шкалу.

Тобто при певних умовах різним типам групових об'єктів можуть належати одні і ті ж значення інформаційних ознак. Це не дозволяє робити однозначний висновок про тип об'єкта, що спостерігається.

З огляду на принципову нечіткість, закладену в фізичну природу процесів, властивих зовнішніх умов, і специфіку особливостей обробки інформації при вирішенні задачі виявлення в приладах, розробку алгоритмів розпізнавання ГО необхідно виконати в нечіткій постановці [1, 15, 16].

Ієрархічно організований ГО (територіальний об'єкт) може бути описаний також за допомогою апарату фракталів. В роботі [30] показано, що багаточарові нейронні мережі

(БНМ) можуть служити генераторами фрактальних структур і використовуються в цій якості як інструмент моделювання та аналізу нелінійних явищ.

Як відомо, ієрархічний територіальний об'єкт (ТО), як елемент ГС, представляє собою набір вкладених територіальних компонент, кожна з яких містить власні вкладені компоненти. При описі об'єкта необхідно зберегти взаємне розташування його компонент і оціночні розміри. Сам об'єкт і будь-яка його вкладена компонента можуть бути незалежно повернені в просторі.

На територіальному об'єкті для кожного рівня ієрархії фіксується ортогональний репер (u_i, v_i) і вибирається число градацій координат p_i , мінімально достатня для розмежування вкладених компонент даного рівня ієрархії (в припущенні, що репер може бути повернений на довільний кут).

Символи (u_i, v_i) будемо вважати цілочисельними координатами реперної площини з можливими значеннями $\{0, 1, \dots, p_i - 1\}$. За умови мінімальної достатності кожна вкладена компонента рівня займає не більше однієї клітини реперною площині.

Для кожного рівня ієрархії фіксується лінійний масштаб m_i , який будемо ставити лінійним розміром однієї координатної градації. Передбачається, що реperi всіх ієрархічних рівнів мають однакову орієнтацію в площині. Реперні координати будь-якої точки об'єкта визначаються функціями:

$$u = \langle u_0 u_1 \dots u_{k-1} \rangle = u_0 p_1 p_2 \dots p_{k-1} + u_1 p_2 p_3 \dots p_{k-1} + \dots + u_{k-2} p_{k-1} + u_{k-1}$$

$$v = \langle v_0 v_1 \dots v_{k-1} \rangle = v_0 p_1 p_2 \dots p_{k-1} + v_1 p_2 p_3 \dots p_{k-1} + \dots + v_{k-2} p_{k-1} + v_{k-1}$$

Лінійні координати будь-якої точки об'єкта можна обчислити, якщо в наведених функціях зробити заміни змінних $p_i \rightarrow (p_i m_i)$.

В координатному репері об'єкт кожного рівня ієрархії описується двовимірною $(0,1)$ функцією $\varphi(u_i v_i)$, яка приймає значення 1, якщо координатна точка $(u_i v_i)$ покривається вкладеною компонентою, і 0 – в іншому випадку.

Ієрархічний територіальний об'єкт у реперних координатах можна представити у вигляді фрактального добутку [26]:

$$f(u, v) = \varphi_{i^0}(u_0, v_0) \varphi_{i^1}(u_1, v_1) \dots \varphi_{i^{k-1}}(u_{k-1}, v_{k-1})$$

де:

$$i^0 = 1, i^1 = \langle u_0 v_0 \rangle, i^2 = \langle u_0 v_0 u_1 v_1 \rangle, \dots, i^{k-1} = \langle u_0 v_0 u_1 v_1 \dots u_{k-2} v_{k-2} \rangle.$$

Даний підхід до формалізованого опису геоінформаційних ситуацій забезпечує рішення задачі їх класифікації за допомогою штучних нейронних мереж.

Таким чином модель ТО є багаторівневою і відрізняється нечіткістю інформаційних ознак у вигляді набору векторів та їх характеристик, які інваріантні до масштабу, зсуву і куту повороту, і набором правил і ієрархічної збірки, що дозволяє описувати геоінформаційні ситуації довільної складності з урахуванням контексту зовнішніх умов.

Використання нечітких ІІ змушує досліджувати вид їх функцій приналежності з метою знаходження оптимальних, тобто найбільш відповідних певним зовнішнім умовам функціонування ДІ.

2.2 Побудова функції належності нечітких інформаційних ознак територіальних об'єктів на основі байєсівського правила

Нехай в ознаковому просторі з носієм X визначена лінгвістична змінна:

$$\Theta = (\theta_0, \theta_1, \theta_2, \dots, \theta_n),$$

де терми $\theta_0, \theta_1, \theta_2, \dots, \theta_n$ відповідають класам об'єктів, які розпізнаються, а терм θ_0 розглядається як відмова від розпізнавання.

Під відмовою розуміється ситуація, коли факт наявності повідомлення достовірно встановлений, але воно не може бути класифіковано. Передбачається, що для кожного ненульового терму в просторі ознак відомий еталонний образ об'єкта – \bar{x} . У імовірнісному просторі набору термів відповідає набір несумісних гіпотез.

Апріорні ймовірності гіпотез вважаються відомими, і оскільки вони становлять повну групу подій, то:

$$\sum_{i=0}^n p(\theta_i) = 1.$$

Для нульової гіпотези ймовірність $p(\theta_i)$ розглядається як завжди апріорна ймовірність появи територіального об'єкта, що не належить відомому набору об'єктів. Передбачається, що кожен ненульовий терм характеризується унімодальною, умовною функцією розподілу $f_i(x/\theta_i)$, яка задана на носії X .

При цьому максимум функції досягається, коли значення інформативної ознаки відповідає еталонному представнику об'єкта. Для нульового терму θ_0 передбачається рівномірна щільність $f_i(x/\theta_i)$ імовірнісного розподілу на носії X .

Нехай в результаті експерименту визначено значення ознаки x для об'єкта, який розпізнається.

За формулою Байєса [31] апостеріорна ймовірність гіпотези θ_i визначається правилом:

$$p(\theta_i, lx) = \frac{p(\theta_i)f_i(x/\theta_i)}{f(x)}, \quad (2.8)$$

де:

$$f(x) = \sum_{i=0}^n p(\theta_i) f_i\left(\frac{x}{\theta_i}\right), \int_{x \in X} f(x) dx = 1, \quad (2.9)$$

$$\sum_{i=0}^n p(\theta_i/x) = 1. \quad (2.10)$$

Спираючись на апостеріорні щільності ймовірності $p\left(\frac{\theta_i}{x}\right)$ ($i = \overline{1, n}$), потрібно вирішити, до якого класу слід віднести об'єкт, який розпізнається.

Будемо вважати, що відома платіжна матриця $C = \|c_{ij}\|$, де $c_{ij} = c(\theta_i, \theta_j)$ – втрати, які пов'язані з правильним рішенням (зазвичай $c_{ii} \leq 0$, а $c_{ij} > 0$). Використовуючи платіжну матрицю, можна побудувати функцію умовного ризику, пов'язаного з вибором класу θ_j :

$$r\left(\frac{\theta_j}{x}\right) = \sum_{i=0}^n c(\theta_i, \theta_j) p(\theta_i/x).$$

Для кожного ненульового терму θ_i визначимо вирішальне правило, задав його у вигляді функції приналежності $\mu_i(x) \leq 1$, з єдиною умовою $\mu_i(\bar{x}) = 1$. Тоді значення ризику при відношенні до класу θ_j при всіх можливих значеннях x можна задати правилом:

$$r(\theta_j) = \int \mu_j(x) r\left(\frac{\theta_i}{x}\right) dx = \sum_{i=0}^n c(\theta_i, \theta_j) \int \mu_j(x) p(\theta_i/x) dx.$$

Середній ризик відмови від розпізнавання при цьому визначається виразом:

$$r(\theta_j) = \int \mu_0(x) r\left(\frac{\theta_i}{x}\right) dx = \sum_{i=0}^n c(\theta_i, \theta_0) \int \mu_0(x) p(\theta_i/x) dx,$$

де $\mu_0(x) = (1 - \mu_1(x))(1 - \mu_2(x)) \dots (1 - \mu_n(x))$.

Середній ризик прийняття рішення по всім термам буде:

$$r = \sum_{j=0}^n \sum_{i=0}^n c(\theta_i, \theta_j) p(\theta_j) \int \mu_j(x) p(\theta_i/x) dx.$$

Оптимальними вважаються функції приналежності, які мінімізують значення середнього ризику.

Коли носій X має нечислову природу, то отриманими формулами скористатися неможливо. Стандартний прийом [23] подолання зазначеного обмеження полягає в побудові для кожного класу числових оцінок близькості елементів носія до еталону класу. Як правило, ці функції оцінок відомі. Вони визначаються вибраними методами вимірювань нечислових ознак в системі розпізнавання.

Позначимо функцію оцінки близькості для класу θ_i , через $\Delta_i = \Delta_i(x, \bar{x}_i)$. Мінімальною вимогою до функції оцінки є однозначність відображення простору X у просторі Δ . Умова $\Delta_i(x_1, \bar{x}_i) = \Delta_i(x_2, \bar{x}_i) = a$ задає відношення еквівалентності на носії X , яке розбиває його на області, які не перетинаються – ядра відображення, які параметризовані значенням a .

Кожне ядро (позначається далі як $\Delta_i^{-1}(a)$) визначає множини елементів x , для яких $\Delta_i(x, \bar{x}_i) = a$. Чим менше гранулярність ядер, тим вище якість вимірювача.

Використовуючи значення оцінок близькості, побудуємо для кожного терму умовні функції розподілу $f_i^\Delta(\Delta/\theta_i)$. Правило побудови можна описати наступною символічною формулою:

$$f_i^\Delta\left(\frac{\Delta}{\theta_i}\right) = \frac{1}{\Delta} \int f_i(x/\theta_i) dx.$$

Інтеграл тут виражає сумарну імовірність приналежності елементів x ядру відображення, параметризованого значенням Δ . Функцію приналежності у просторі Δ визначимо правилом:

$$\mu_i^\Delta(\Delta) = \max_{x \in \Delta_i^{-1}(\Delta)} (\mu_i(x)).$$

Далі з метою спрощення запису верхній індекс Δ в позначеннях f_i^Δ і μ_i^Δ будемо опускати, використовуючи ті ж позначення, що і в просторі ознак X .

Використовуючи оцінки близькості, знайдемо вирази для знаходження оптимальних функцій приналежності при нечислових носіях. За формулою Байєса маємо:

$$p\left(\frac{\theta_i}{\Delta_i}\right) = \frac{p(\theta_i)f_i(\Delta_i/\theta_i)}{f_i(\Delta_i)}. \quad (2.11)$$

де:

$$f_i(\Delta_i) = \sum_{i=0}^n p(\theta_i)f_i(\Delta_i/\theta_i), \quad \int f(\Delta_i)d\Delta_i = 1. \quad (2.12)$$

Основна відмінність даних формул від (2.8) – (2.9) полягає в тому, що для кожного терму визначена своя «метрика» і свій простір, в якому обчислюються функції $p(\theta_i/\Delta_i)$ і $f_i(\Delta_i)$. Тому аналога для рівності (2.10) не існує.

Формула обчислення середнього ризику через оцінки близькості має вигляд:

$$r = \sum_{j=0}^n \sum_{i=0}^n c(\theta_i, \theta_j)p(\theta_j) \int \hat{\mu}_j(\Delta_j)p(\theta_i/\Delta_i)d\Delta_i.$$

При виконанні інтегрування співвідношення між метриками Δ_i та Δ_j визначається з умови:

$$\Delta_i^{-1} \cap \Delta_j^{-1} \neq \emptyset, \quad \text{при чому } \hat{\mu}_j(\Delta_j) = \max_{\Delta_i^{-1} \cap \Delta_j^{-1}} \mu_j(\Delta_j). \quad (2.13)$$

Середній ризик відмови від розпізнавання визначається виразом:

$$r(\theta_0) = \int \hat{\mu}_0(\Delta_0)r\left(\frac{\theta_i}{\Delta_0}\right)d\Delta_0 = \sum_{i=0}^n c(\theta_i\theta_0) \int \hat{\mu}_0(\Delta_0)p(\theta_i/\theta_0)d\Delta_0,$$

$$\text{де } \hat{\mu}_0(\Delta_0) = (1 - \hat{\mu}_1(\Delta_1))(1 - \hat{\mu}_2(\Delta_2)) \dots (1 - \hat{\mu}_n(\Delta_n)).$$

У даному виразі відповідність метрик визначається умовами (2.13).

Побудовані таким чином функції приналежності є оптимальними при фіксованих (стандартних) імовірнісних характеристиках дискретного каналу. У реальній системі статистичні характеристики каналу істотно змінюються в залежності від погодних умов і часу доби [37].

Тому при фіксованих функціях приналежності будь-яка відмінність характеристик каналу від стандартних в кращу або гіршу сторону буде приводити до збільшення помилок розпізнавання. І якщо зі збільшенням помилок при погіршенні каналу в більшості випадків можна миритися, то помилкове розпізнавання свідомо достовірних слів є неприйнятним.

Дозволити дане протиріччя можна двома шляхами: або безперервно відстежувати ймовірні характеристики каналу і змінювати функції приналежності в реальному часі, або

вибирати квазіоптимальну функції приналежності, які оптимізовані до заданої варіації характеристик каналу.

Перший варіант досить дорогий в реалізації і пред'являє високі вимоги до обчислювальної потужності центрального інформаційного центру, другий варіант технічно більш простий, але має гірші характеристики в розпізнаванні, ніж перший.

Розглянемо далі спосіб побудови адаптивної схеми розпізнавання з квазіоптимальним правилом вибору функцій приналежності.

Квазіоптимальна функція приналежності передбачає, що при передачі інформаційних слів по каналах з різними імовірнісними характеристиками (від ідеального каналу до стандартного) ймовірність помилок класифікації для вирішальної схеми буде лежати в межах заданого допуску. Ця вимога може бути реалізованою, коли функції приналежності при більш високому рівні перешкод покривають функції приналежності, побудовані для більш низьких рівнях спотворень в каналі.

Іншими словами нечіткі множини однойменних термів утворюють послідовні вкладення при зростанні рівня спотворень в каналі. При варіаціях характеристик каналу відбувається зміна умовних щільностей ймовірностей $f_i(\Delta_i/\theta_i)$.

У разі ідеального каналу ці щільності прагнуть до імпульсної δ -функції, навпаки, при погіршенні характеристик збільшується дисперсія і змінюється математичне очікування випадкових величин Δ_i .

Зміна математичного очікування породжує зону, в межах якої значення функції приналежності слід прирівнювати до одиниці (відповідно до умови $\mu_i(\bar{x}) = 1$). Це призводить до необхідності використання (L, R) – апроксимації даних функцій.

Задача вибору функцій приналежності зазнає суттєвого спрощення, якщо покласти, що довжина зони, де $\mu_i(\Delta_i) = 1$, пропорційна різниці математичних очікувань апріорних розподілів $f_i(\Delta_i/\theta_i)$ у крайніх варіаціях каналу, а ширина лівої і правої граней функції приналежності пропорційна дисперсія цих розподілів.

У цьому випадку апостеріорно необхідно визначити тільки коефіцієнти пропорційності, які будуть спільними для всіх термів лінгвістичної змінної. Число параметрів, що налаштовуються при цьому істотно скорочується.

І якщо вдається зв'язати параметри розподілів з характеристиками каналу, то можна знайти аналітичні вирази для правил адаптації схеми розпізнавання до статистичних характеристик каналу. Це дозволяє значно прискорити процедуру перебудови та адаптації схеми розпізнавання.

Даний підхід використовується для знаходження квазіоптимальних функцій приналежності для лінгвістичних змінних «семантика» і «довжина об'єкта» [38].

Для лінгвістичної змінної «структура» специфіка оціночної функції, не дозволяє явно зв'язати параметри випадкових величин Δ_i , з характеристиками каналу і тому тут використовується прийом автоматичного формування $L - R$ зони за рахунок нормування вектора значень функцій приналежності лінгвістичної змінної.

Нормування, крім того, дозволяє отримати оцінки достовірності результатів розпізнавання, і з такою метою використовується і для інших лінгвістичних змінних.

2.3 Метод розпізнавання образів геоінформаційних ситуацій за нечіткою та неповною вихідною інформацією в системах моніторингу територій

Метод розпізнавання геоінформаційних ситуацій по нечіткій і неповній вихідній інформації на основі нечіткої штучної нейронної мережі заснований на отриманні та використанні наступних оцінок і моделей подання знань:

- інформації від систем спостереження (оцінки прямих вимірювань параметрів обстановки, наприклад, параметри одиночних і групових об'єктів в зоні спостереження;
- формалізованому багаторівневому описі класів еталонів ТО в нечіткій постановці (ап'юріорні оцінки інформаційних ознак еталонів ситуацій обстановки);
- формалізованому описі контексту зовнішніх умов і знань осіб, які здійснюють розпізнавання ситуацій обстановки неавтоматизованим способом у вигляді функцій приналежності відповідного виду.

Суть методу розпізнавання полягає в обчисленні оцінок узагальненого критерію зовнішніх умов отримання і обробки вектора вихідної інформації, в застосуванні порогових процедур адаптації нечітких еталонів ТО до зміни контексту зовнішніх умов за допомогою управління їх інваріантними властивостями і використанні нечіткої ШНМ (рис. 2.2).

При розпізнаванні геометричного об'єкта проглядається множина векторів карти і до кожного вектору прикладається формоутворювальний вектор a_i еталонного образу. При цьому еталонний вектор позиціонується так, що його початкова точка і напрямок збіглися з початковою точкою і напрямом вектора карти. З початкової точки вектора на карті проводиться орієнтований промінь під кутом ψ_i до вектору карти. Дана процедура породжує множину пересічних променів. Для наочного уявлення методу розпізнавання

будемо вважати, що всі точки орієнтованого променя мають одиничну яскравість, а в точках перетину променів яскравості складаються.

Тоді, очевидно, що якщо еталонний об'єкт представлений на карті, то точка перетину променів, що володіє максимальною яскравістю, буде відповідати положенню опорної точки об'єкта.

Даний метод розпізнавання має повну інваріантність до положення об'єкта, його орієнтації і масштабу. При наявності перешкод деякі вектора еталонного образу можуть виявитися не виявленими на карті, це призводить не більше ніж до зниження яскравості опорної точки, і може бути легко компенсовано збільшенням числа опорних векторів в описі еталонного об'єкта.

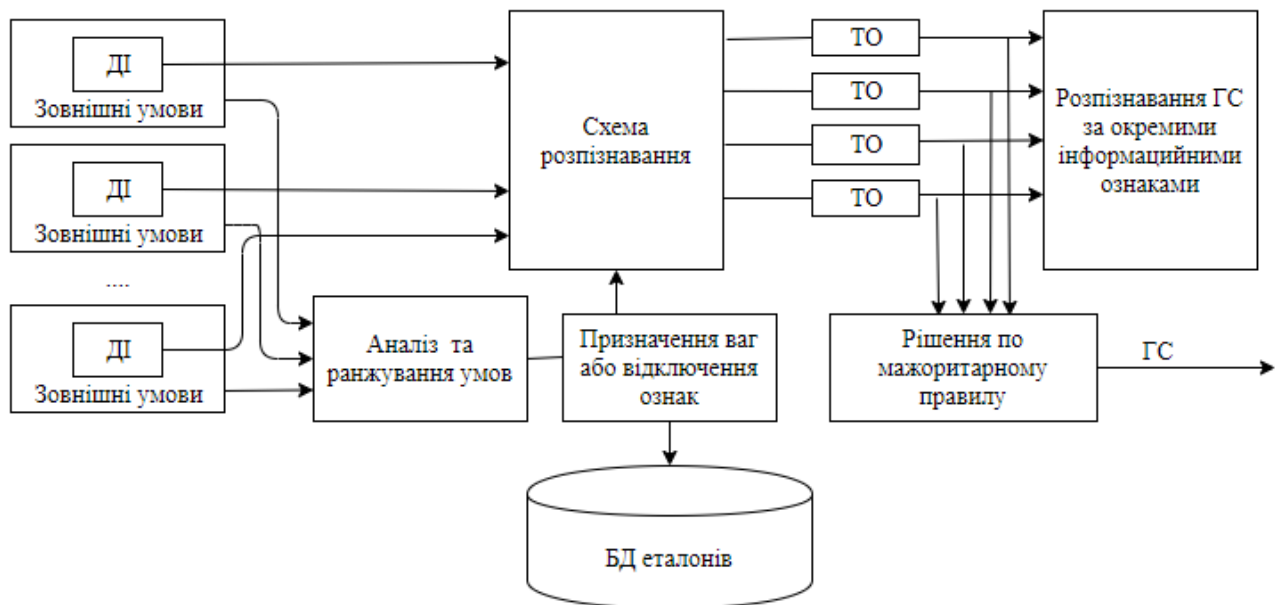


Рисунок 2.2 – Метод розпізнавання геоінформаційних ситуацій в залежності від зовнішніх умов

Важливою перевагою запропонованого методу є те, що він має здатність до зміни інваріантних властивостей. Наприклад, для того щоб розрізняти однотипні об'єкти по кутовий орієнтації, досить при скануванні множини векторів карти пропускати всі вектора, які по напрямку не збігаються ні з одним формотворчим вектором еталонного об'єкта.

Для розрізнення об'єктів за масштабом досить при скануванні безлічі векторів карти, пропускати всі вектора які по довжині не збігаються ні з одними з формотворчих векторів еталонного об'єкта [39].

У загальному випадку для кожної властивості при виконанні операції порівняння векторів можна визначити деякий інтервал толерантності. Це дозволяє кількісно регулювати ступінь інваріантності методу розпізнавання.

Подібним чином можна задати і область допустимого розміщення об'єкта на карті. Властивість керованої інваріантності дозволяє досить просто класифікувати по таксономічним класам однотипні об'єкти, що відрізняються діапазоном орієнтації, масштабу і розміщення.

2.4 Формування опису основних лінгвістичних змінних територіальних об'єктів у системах моніторингу територій

Як було показано вище, в якості основних ІОЗ територіального об'єкта лінійного типу при побудові системи розпізнавання доцільно вибрати такий набір інформаційних ознак: структуру, склад і довжину територіального об'єкта. Тоді в буквено-цифровій нотації інформаційний образ (ІО) територіального об'єкта лінійного типу матиме вигляд:

$$I_{\text{обр}} = (m_{i_1} a_{i_1} m_{i_2} a_{i_2} \dots m_{i_p} a_{i_p}), \quad (2.14)$$

де m_i – довжина поля однорідних символів виду a_i ;

p – число полів у груповому об'єкті. Поля розділені не більше ніж трьома символами a_0 .

Кортеж чисел, складений з довжин полів ІО об'єкта, будемо називати структурою територіального об'єкта, а кортеж з символів його полів – складом або семантикою ТО, підкреслюючи цим смисловий зміст порядку проходження однотипних об'єктів, що утворюють поле.

Далі в роботі використовується термін «Семантика ТО». Наприклад, якщо для розглянутого вище джерела ввести літерні позначення для символів алфавіту, поклавши $K = 1$, $W = 2$, $T = 3$, $G = 4$, то інформаційний образ територіального об'єкта (2.14) у цьому випадку може бути записаний в такій формі:

$$I_{\text{обр}} = (2G 2K 4G 3T 4G 4G),$$

де цифри вказують на розмір поля, а буквені символи визначають тип одиночного об'єкта. Для приведеного вище приклада кортежі структури та семантики матимуть вигляд:

$$m = (m_1 m_2 m_3 m_4 m_5 m_6) = (2 \ 2 \ 4 \ 3 \ 4 \ 4),$$

$$s = (s_1 s_2 s_3 s_4 s_5 s_6) = (G \ K \ G \ T \ G \ G).$$

Сума чисел кортежу структури визначає довжину територіального об'єкта лінійної структури.

Всі ці ознаки взаємопов'язані, але в той же час доповнюють один одного. Для кожної інформаційної ознаки система розпізнавання включає в себе окремі підсистеми, виходи яких об'єднуються при формуванні остаточного виведення. В даний час в такому варіанті задача розпізнавання інформаційного образу територіального об'єкта вирішується людиною-оператором.

Операторами накопичено багатий досвід аналізу інформації, який доцільно використовувати при побудові системи автоматичного розпізнавання. При наявності експертних знань найбільш підходящим варіантом побудови системи автоматичного розпізнавання, є нечіткі нейронні мережі [40, 41].

У цьому класі нейронних мереж, експертні знання можна використовувати як при виборі функцій належності вхідних лінгвістичних змінних, так і при виборі структури та топології нейронної мережі.

Використання нечіткої нейронної мережі пов'язано з фазифікацією вхідної інформації. На цьому етапі вводяться лінгвістичні змінні і визначаються відповідні функції приналежності. У розглянутій задачі лінгвістичними змінними вважаються структура, семантика і довжина територіального об'єкта. Число термів лінгвістичних змінних визначається числом типів територіальних об'єктів.

Відмінною особливістю процедури фазифікації в даній задачі є імовірнісна природа породження нечіткості, обумовлена випадковим характером перешкод в дискретному каналі. У такому варіанті функції приналежності, по суті, визначають вирішальні правила статистичної класифікації. Тому для побудови оптимальних функцій приналежності пропонується використовувати метод, заснований на баєсівському правилі, яке було розглянуто раніше.

3 ДОСЛІДЖЕННЯ МЕТОДУ РОЗПІЗНАВАННЯ ГЕОІНФОРМАЦІЙНИХ СИТУАЦІЙ В СИСТЕМАХ МОНІТОРИНГУ ТЕРИТОРІЙ

3.1 Вибір інструментів для вирішення поставлених задач

У зв'язку з відсутністю реально функціонуючої СМТ, оцінка ефективності запропонованих методів та алгоритмів розпізнавання геоінформаційних ситуацій по нечіткій і неповній вихідній інформації виконувалася засобами комп'ютерного моделювання. Для даних цілей було вирішено використовувати геоінформаційну систему Quantum GIS.

Основним призначенням системи є обробка і аналіз просторових даних, підготовка різної картографічної продукції. Інтерфейс QGIS побудований на базі бібліотеки Qt. Пакет має гнучку систему розширень, які можна створювати на мовах C++ і Python. Підтримуються різноманітні векторні і растрові формати з ESRI Shapefile і GeoTIFF включно.

GIS QGIS дозволяє користувачам створювати карти з безліччю шарів, використовуючи різні картографічні проекції. Карти можуть бути зібрані в різні формати і використовуватися з різною метою. У системі QGIS карти можуть складатися з растрових або векторних шарів. Типовими для такого роду програмного забезпечення, векторні дані зберігаються як точка, лінія, полігон. Підтримуються різні види растрових зображень. Програмне забезпечення може виконувати геоприв'язку зображень.

У QGIS можна створювати і редагувати векторні дані, а також експортувати їх в різні формати. Щоб мати можливість редагувати і експортувати в інші формати растрові дані, необхідно спочатку імпортувати їх в GRASS. QGIS надає, зокрема, такі можливості роботи з даними:

- інструменти оцифровки для форматів, підтримуваних бібліотекою OGR, і векторних шарів GRASS;
- створення і редагування shape-файлів і векторних шарів GRASS;
- геокодування зображень за допомогою модуля просторової прив'язки;
- інструменти GPS для імпорту і експорту даних в форматі GPX, перетворення інших форматів GPS в формат GPX або завантаження безпосередньо в прилад GPS;
- візуалізація і редагування даних OpenStreetMap;

- створення шарів PostGIS з shape-файлів за допомогою плагіна SPIT;
- обробка шарів PostGIS.

Ви можете аналізувати векторні просторові дані в PostgreSQL/PostGIS та інших форматах, підтримуваних OGR, використовуючи модуль Processing, написаний на мові програмування Python.

У даний час QGIS надає можливість використовувати інструменти аналізу, вибірки, геопроецювання, управління геометрією і базами даних. Також можна використовувати інтегровані інструменти GRASS, які включають в себе функціональність більш ніж 300 модулів GRASS.

Джерелом геоінформаційної інформації було вирішено обрати вільне програмне забезпечення для навігації SAS.Planet. Воно поєднує в собі можливість завантаження, перегляду та аналізу карт і супутникових фотографій земної поверхні великої кількості картографічних online-сервісів.

SAS.Планета надає єдиний інтерфейс завантаження і обробки картографічного матеріалу, що допомагає розв'язати такі проблеми:

- незважаючи на наявність у мережі безлічі географічних сервісів, що надають можливість перегляду картографічного матеріалу, всі вони різняться якістю і форматами матеріалів, що надаються (за такими параметрами як масштаб карт, охоплення територій, відображувані об'єкти тощо), що ускладнює пошук потрібної карти;
- обмеження доступу до більшості таких сервісів, коли користувач має можливість перегляду карти тільки в браузері, часто витрачаючи зайвий трафік на завантаження одних і тих же областей;
- неможливість або важкість для користувача окремих географічних сервісів зберегти необхідну йому ділянку карти для того щоб використовувати її у своїх потребах (наприклад, в програмах для розпізнавання геоінформаційних ситуацій).

3.2 Вихідні дані для дослідження

3.2.1 Структура нечіткого розташування геоінформаційних об'єктів та ситуацій у системах моніторингу територій

Система моделювання нечіткого розпізнавання геоінформаційних об'єктів і ситуацій складається з сукупності алгоритмів розпізнавання ТО, а також їх сервісного оточення у вигляді модулів завдання впливу контексту зовнішніх умов, завдання

структури і параметрів узагальненої моделі ТО, статистичної обробки і представлення результатів моделювання. Структура СМ представлена на рис. 3.1.

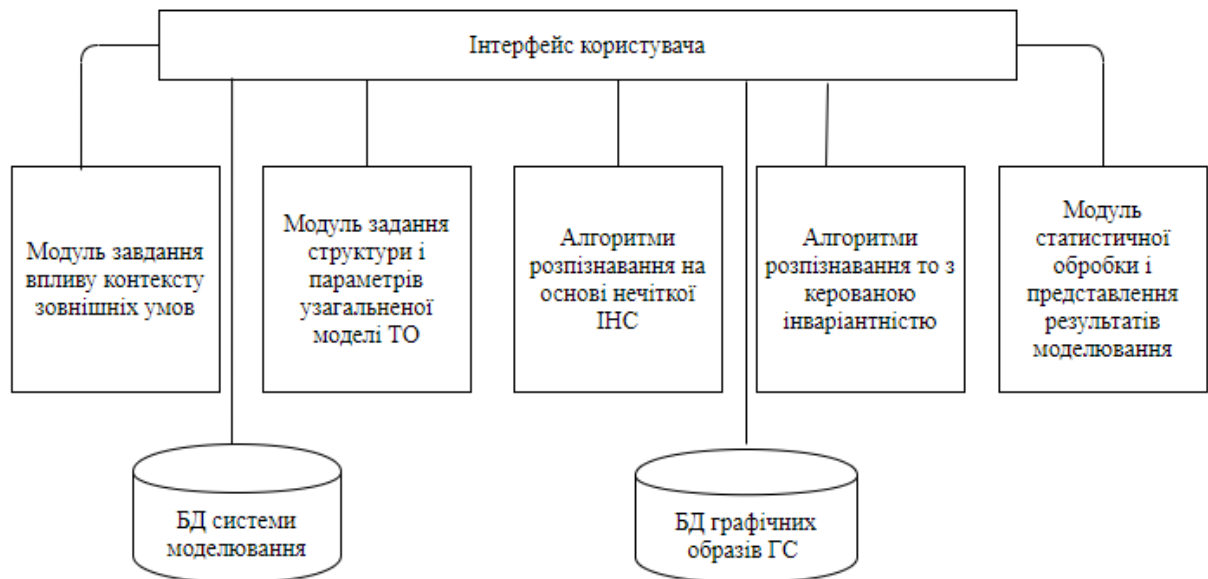


Рисунок 3.1 – Структура системи моделювання нечіткого геоінформаційного розпізнавача об'єктів та ситуацій

Модуль статистичної обробки і представлення результатів моделювання реалізує стандартні процедури, виконані в даному пакеті прикладних програм. Модуль завдання структури і параметрів моделі ТО через інтерфейс кінцевого користувача реалізує спосіб їх опису, розглянутий раніше в розділі 2. Слід зазначити, що варіант реалізації модуля завдання впливу контексту зовнішніх умов визначається конкретним типом джерела інформації і вимагає розгляду фізики процесів впливу зовнішнього середовища на результати прийняття рішень в системі в конкретних умовах його функціонування, що є самостійним дослідженням.

3.2.2 Вибір архітектури нечіткої штучної нейронної мережі для задачі розпізнавання територіальних об'єктів у системах моніторингу територій\

У розпізнаванні образів нечіткі нейронні мережі є унікальним інструментом, що дозволяє реалізувати ефективне поєднання експертних знань з навчанням до фактичного набору даних.

Експертні знання можна використовувати як при виборі функцій приналежності лінгвістичних змінних, так і при виборі структури та топології нечіткої нейронної мережі. У першому випадку використовуються апіорні знання про інтервали невизначеності вхідних змінних, а в другому – знання про логічне алгоритмі класифікації.

Можливість вкласти в структуру мережі логічні знання заснована на подобі операцій нечіткої логіки і логічних правил, якими керується людина при прийнятті рішень.

Значення функції приналежності в цьому випадку інтерпретується як ступінь довіри до результату, а нечітка нейронна мережа виконує агрегування даних за допомогою послідовного виконання операцій fuzzyлогіки «І», «АБО».

3.2.3 Побудова функцій активації нечітких штучних нейронів

Типові операції fuzzy логіки (*max – min* композиції, *min – max* композиції, *t*-норми, *k*-норми та ін.) мають той недолік, що результат їх виконання не цілком відповідає інтерпретації функцій приналежності як ступеня достовірності наших знань.

Наприклад, якщо вхідні аргументи «І»-композиції мають високі значення (високий ступінь довіри), то в разі спільних подій від результату «І»-композиції суб'єктивно очікується більш високе значення, ніж значення кожного аргументу окремо.

Оскільки якість класифікації за двома показниками з високими ступенями довіри, що діють спільно, має бути краще, ніж з будь-якого показника окремо (при несумісних подіях результат типової «І»-композиції нульовий, тобто відповідає суб'єктивно очікуваному).

Якщо значення аргументів на вході «І»-композиції малі (низький ступінь довіри), то як при спільних, так і несумісних подіях логіка операції «І» відповідає суб'єктивно очікуваній – ступінь довіри до результату «І»-композиції буде не вище, ніж до кожної події окремо.

Аналогічно для «АБО»-композиції малі значення вхідних аргументів, що відповідають спільним подіям, повинні підсилювати один одного, оскільки зростає ступінь довіри до факту, що якась подія сталася. Навпаки, якщо вхідні значення великі, а відповідні події є несумісними, то тоді ступінь довіри до їх «АБО»-композиції суб'єктивно очікується нижче, ніж кожного окремо (а при спільних подіях – не вище ніж кожного окремо, що відповідає операції *max*).

Дані міркування приводять до необхідності використовувати в fuzzy композиціях контрастуючі функції з граничними властивостями, які враховують сумісність і несумісність подій і встановлюють межу між високою і низькою ступенем довіри. Контрастуючі функції можна розглядати як функції активації нечітких нейронів.

Простір нечітких змінних визначено на інтервалі $[0 \div 1]$, тому від контрастуючої функції потрібно, щоб при зміні її аргументів в діапазоні $[0 \div 1]$ значення функції належало діапазону $[0 \div 1]$.

Глибина контрастування зростає при збільшенні порогового рівня та падає до мінімуму, якщо граничний рівень прагне до нуля, В тому випадку, коли необхідний зворотний характер зміни глибини контрастування, можна використовувати активаційну функцію виду

$$\varphi_{\alpha}(x) = 1 - f_t(1 - x) \text{ при граничній умові } \alpha = 1 - t.$$

Використання контрастуючих функцій в нечіткій логіці дозволяє отримати коректну суб'єктивну інтерпретацію базових логічних операцій. Нечіткі нейрони з контрастуючими функціями активації істотно розширюють сферу застосування нечітких нейронних мереж.

Перший шар. У нейронному шарі формуються нечіткі лінгвістичні змінні: «Структура» – S , «Семантика 1» – S_{m_1} , «Семантика 2» – S_{m_2} , «Довжина об'єкта» – L .

Другий шар. В шарі виконується поразрядне контрастне логічне множення векторів функцій приналежності:

$$y_n(k) = S(k) \& L(k) = Cntr_{0.2}(\min(S(k), L(k))).$$

Операція логічного множення контрастується функцією $Cntr_{\alpha}(x) = x^{b(1-x)^{\beta}}$ з граничним значенням $\alpha = 0.2$ і параметром $\beta = 1$.

Третій шар. В шарі виконується порозрядне контрастне логічне складання:

$$z(k) = Cntr_{0.5}^{-}(\max(y_1(k), y_2(k), y_3(k))).$$

Функція зворотного контрастування з параметром $\beta = -0.6$ має граничне значення $\alpha = 0.5$. Виходом третього шару є композиційна функція приналежності образу, що розпізнається, до набору еталонів.

Четвертий шар. У нейронному шарі реалізовано посилення діагнозів. Операції в шарі виконуються, коли координати $z(k)$ і $z(j)$ мають близькі значення:

$$\gamma_z = \frac{\max(z(k),z(j)) - \min(z(k),z(j))}{\max(z(k),z(j))} < 0.2.$$

В цьому випадку перевіряються окремі діагнози по змінним y_1, y_2, y_3 в порядку переваг $y_1 > y_2 > y_3$.

3.3 Тестування методу розпізнавання геоінформаційних ситуацій на основі нечітких штучних нейронних мереж у системах моніторингу територій

Метод роботи функції поділу потоку на лінійні об'єкти ЕК представлений на рисунку 3.2. Для роботи на вхід алгоритму потрібно подати послідовність, прийняту з каналу зв'язку джерела інформації. Вихідними даними є список лінійних об'єктів, знайдених алгоритмом в потоці.

Блок 1 цього алгоритму містить простий цикл, який пропускає у вхідній послідовності початкові нульові сигнали. З появою на вході першого значущого символу цикл передає управління далі.

Блок 2 містить простий цикл, який пропускає у вхідній послідовності ненульові сигнали до появи на вході нуля. Після цього символу цикл передає управління далі.

У блоці 3 відбувається підрахунок нульових символів у вхідному потоці. Підрахунок припиняється з появою на вході першого значущого символу. Після цього цикл передає управління далі.

У блоці 4 перевіряється умова: якщо лічильник нулів з третього блоку дорівнює п'яти або більше (це означає, що знайдений ознака поділу на лінійні об'єкти або вхідна послідовність завершилася), то відбувається перехід на блок 5, інакше алгоритм переходить на блок 6.

У блоці 5 створюється новий елемент списку лінійних об'єктів, В нього записуються сам лінійний об'єкт і деякі його характеристики: фактична довжина, передбачуваний порядковий номер і т.д.

У блоці 6 відбувається перевірка: якщо вхідна послідовність закінчилася, то відбувається повернення отриманого списку і завершення методу, інакше (якщо в потоці ще є об'єкти) – відбувається перехід на блок 2, і метод продовжується.

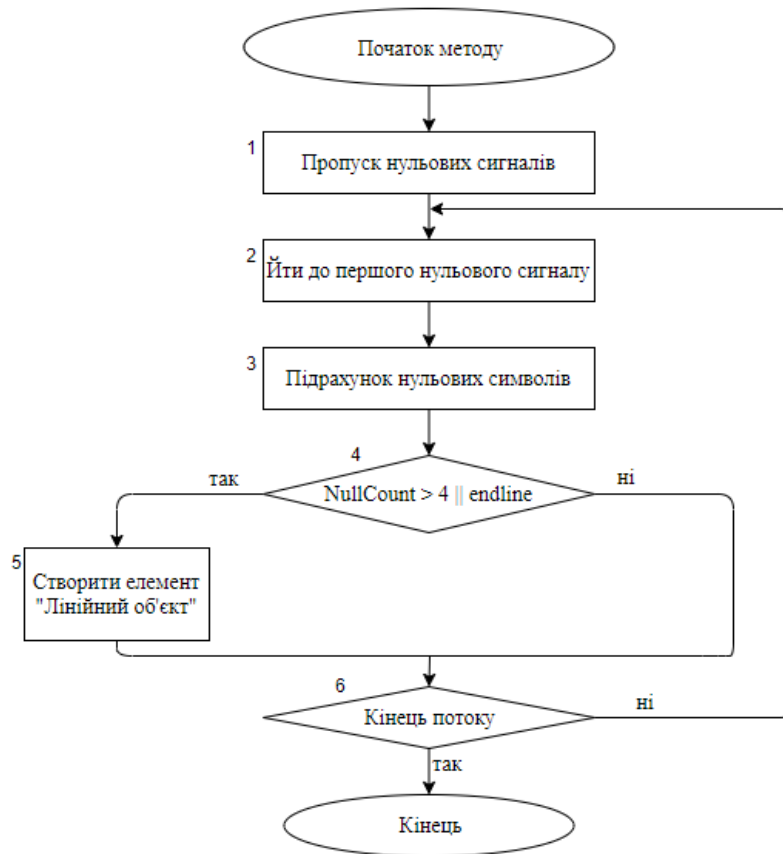


Рисунок 3.2 – Метод поділу потоку на лінійні територіальні об'єкти

3.4 Результати оцінювання ефективності розпізнавання геоінформаційних ситуацій на основі нечітких штучних нейронних мереж у системах моніторингу територій

Дослідження ефективності методів виявлення і розпізнавання ТО по інформаційним ознаками на основі нечіткої ШНМ виконувалося за допомогою розробленої системи моделювання, що складається із сукупності модулів, керованих в процесі комп'ютерних експериментів через загальний інтерфейс кінцевого користувача: модуль завдання структури, складу і властивостей одиночних і групових об'єктів, модуль вихідної інформації, що імітує повідомлення різномірних джерел про виявлені елементах

ТО різного типу, модуля контексту зовнішніх умов, бази даних нечітких інформаційних ознак еталонів, модуля розпізнавання на основі нечіткої ШНМ, модуля обробки статистичних даних (ДОДАТОК А), модуля візуалізації результатів моделювання (ДОДАТОК Б).

Реалізація СМ була виконана на базі прикладного пакету програм QGIS 3.10.0 на прикладі пошуку вільних місць для машини на парковці (рис. 3.3).

Моделювалася обробка інформації в системі моніторингу територій, коли на вхід засобу розв'язання надходить потік повідомлень від джерел інформації про одиночних і групових територіальних об'єктах, що мають різну ймовірність правильного виявлення/класифікації в двох принципово різних режимах. Об'єкти знаходяться на дорозі або розподілені на деякій території, що відповідає зоні відповідальності системи моніторингу. У першому варіанті в процесі експерименту обчислювалася ймовірність помилки розпізнавання і оцінювалася її залежність від змін довжини територіального об'єкта відносно встановленої.

Також оцінювався вплив якості роботи джерела первинної інформації (змін ймовірності помилки у виборчому розпізнаванні одиночних об'єктів), що в імітаційному плані еквівалентно визначенню чутливості досліджуваних методів до змін значень ймовірностей на головній діагоналі матриці змішування [15].

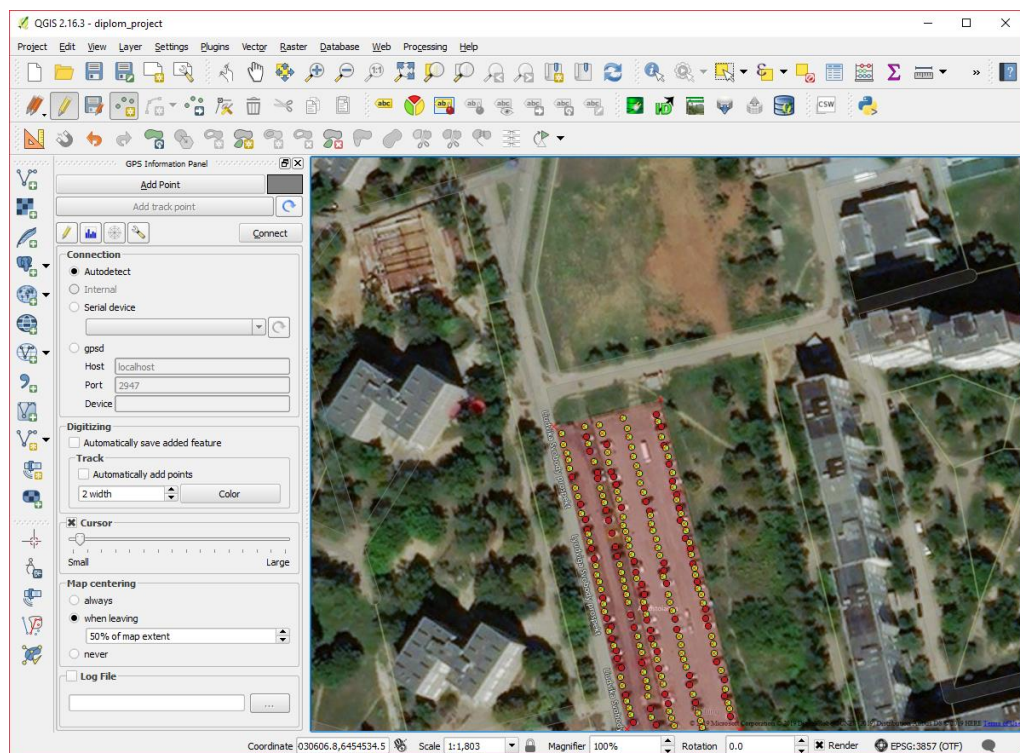


Рисунок 3.3 – Реалізація пошуку вільних місць на парковці з використанням пакету програм QGIS

Перший критерій складається з помилок, пов'язаних з ймовірністю дублювання через ехоєфекту і з взаємодією двох тимчасових потоків подій: руху об'єктів з певними структурними властивостями, параметрами руху і зайнятості каналу обслуговування з кінцевим часом його відновлення. Графік стійкості методу до помилок такого роду наведено на рисунку 3.4.

Експерименти показали високу стійкість даного методу (розроблених методів) при зміні зовнішніх умов. Так, наприклад, при зміні довжини об'єкта, що розпізнається, в діапазоні -27% і $+48\%$ від довжини зразка зберігалася ймовірність правильного його розпізнавання рівна 0.8. Дослідження проводилися на чотирнадцяти стандартах при одиничній матриці зміщення з кроком зміни довжини на 1% від еталонної.

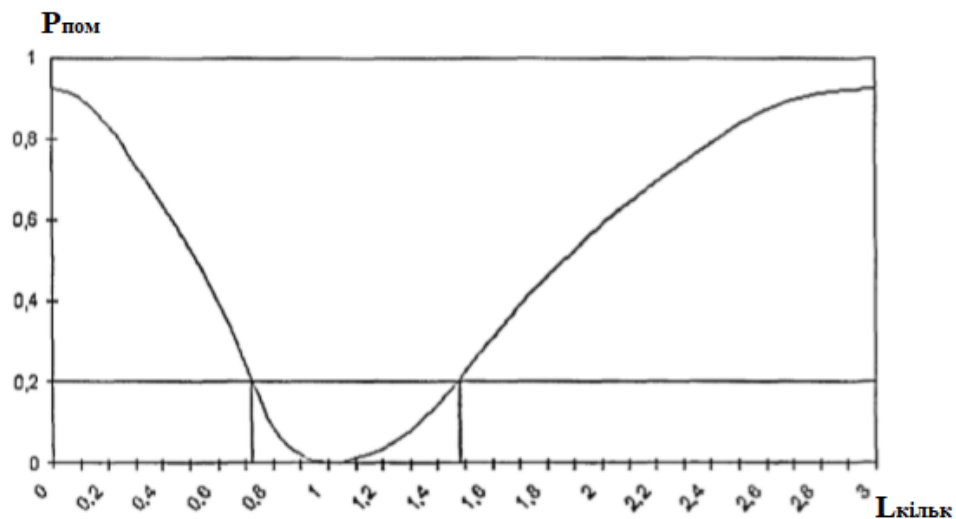


Рисунок 3.4 – Графік стійкості методу до зміни довжини лінійного об'єкта відносно встановленої

Другий критерій стійкості складається з помилок, пов'язаних з великим шумовим фоном природного і штучного походження і з множинним перевідбиттям сигналу від багат шарових порід з різною акустико-сейсмічною провідністю. Графік стійкості методу до помилок такого роду наведено на рисунку 3.5.

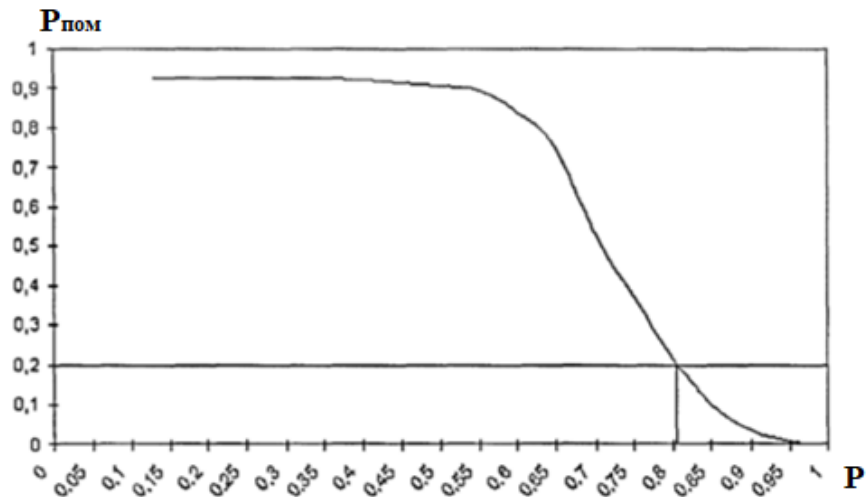


Рисунок 3.5 – Графік стійкості методу до зміни значень на головній діагоналі матриці змішування

Експерименти показали, що даний метод зберігає свої характеристики в широкому діапазоні зміни параметрів матриці змішування.

Так, наприклад, ймовірність правильного розпізнавання, рівна 0,8, зберігалася при зміні значень елементів матриці змішування на головній діагоналі в діапазоні $0,6 \div 0,81$. Дослідження проводилися на чотирнадцяти стандартах при нульовій ймовірності дублювання символів і повної синхронізації руху об'єктів з часом відновлення каналу обслуговування. Декремент значення головної діагоналі дорівнює 0,01.

Класифікації лінійних територіальних об'єктів з інформаційних ознаками з використанням нечіткої нейронної мережі досліджувався в наступній послідовності. Перед початком роботи методу потрібно створити текстовий файл певного формату з вихідними даними для розрахунків. Для цього можна використовувати будь-який текстовий редактор. У файлі опису допустимо використання коментарів.

Запустивши метод класифікації лінійних об'єктів електронної карти з інформаційних ознаками на виконання, потрібно ввести ім'я файлу з вихідними даними для розрахунків, ім'я файлу для запису вихідних даних від сенсорів і ім'я файлу для виведення результату процесу класифікації.

3.5 Перспективи подальших досліджень

Якщо в своїй ранній постановці завдання розпізнавання зводилося до інтерпретації вхідних даних в термінах невеликого числа класів (образів), заданих людиною, то створення інтегрованих систем розпізнавання, в яких завдання сприйняття і інтерпретації інформації про зовнішній світ і завдання прийняття рішення і виконання певних дій по перетворенню цього світу виявилися в значній мірі об'єднаними в рамках однієї системи, що зажадало нової постановки задачі розпізнавання образів.

Нова постановка задачі, яка вирішується системою, полягає в перетворенні вихідних даних про зовнішній світ на деякий мову, що дозволяє здійснювати такий опис сприймаються об'єктів світу, на якому можна було б здійснювати їх виділення, формування понять об'єктів, що входять в зображення, з метою інтерпретації, виявлення просторово-часових відносин між об'єктами, що знаходяться в світі, і, нарешті, здійснення їх класифікації.

Особливістю даного підходу є представлення задачі розпізнавання як задачі обробки знань [11, 12, 14]. Відповідно проблема розпізнавання може бути зведена до задач подання знань та використання методів обробки знань. Одним з найбільш перспективних напрямків такого підходу до вирішення задач розпізнавання є використання технології експертних систем.

У даний час цей напрям отримав розвиток при створенні нового покоління систем інтелектуальних обробки і розпізнавання сигнальної інформації. Тут можна виділити ряд робіт по створенню систем розуміння мови HEARSAY-2, систем аналізу та інтерпретації електрокардіограм, сейсмограм, системи обробки зображень. Необхідно відзначити і поява відкритих публікацій з розробок СІ в галузі підводної акустики. Тут можна відзначити такі розробки як система розпізнавання сигналів HASP SIAP, системи для вирішення задач обробки і розпізнавання гідроакустичних сигналів, система розуміння сигналів в умовах великої невизначеності.

Перевагою підходу, заснованого на штучному інтелекті, в задачах розпізнавання сигнальної інформації є можливість використання багаторівневих систем представлення даних з різним ступенем абстракції, використання різнорідних джерел інформації і об'єднання їх в єдину структуру прийняття рішень на основі використання систем логічного висновку, можливість ефективного моделювання діяльності людини-оператора і використання його знань. Практичне застосування методів штучного інтелекту в системах

класифікації сигналів вже в даний час дозволяє досить ефективно обійти сформульовані вище проблеми класичного підходу до завдань розпізнавання і прийняття рішень.

4 ОХОРОНА ПРАЦІ

4.1 Аналіз потенційно небезпечних і шкідливих виробничих факторів, що впливають на персонал

Персональні ЕОМ типу IBM PC AT має наступні характеристики:

- споживана потужність 350 Вт;
- робоча напруга 220 В;
- напруга джерел живлення +12 В, -12 В, 5 В;
- робоча частота 50 Гц.

Виходячи з приведених характеристик, очевидно, що для користувача існує небезпека поразки електричним струмом у разі недбалого поводження з комп'ютером і порушення правил експлуатації (невиконання огляду відкритих частин ПЕВМ, що знаходяться під напругою або знятих для ремонту вузлів і т. д.).

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;

У відповідності з ДСанПіН 3.3.2-007-98 [41] до легкої фізичної роботи відносяться всі види діяльності, вироблювані сидячи і не вимагаючи фізичної напруги. Робота користувача розробленого пакету програм відноситься до категорії Іа.

Згідно з ДСТУ Б А.3.2-13:2011 [42] приміщення для ПЕОМ по ступеню небезпеки поразки людини електричним струмом відноситься до приміщень без підвищеної небезпеки (немає струмопровідної половини, вогкості, підвищеної температури, можливості одночасного дотику до корпусів устаткування з “землею” і до струмонесучих частин).

У відповідності з НПАОП 0.00-7.15-18 [43] при обслуговуванні ПЕВМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

- підвищене значення напруги в електричному ланцюзі, замикання якого може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювань;
- підвищена або знижена температура повітря робочої зони;
- підвищена або знижена рухливість повітря;
- підвищена або знижена вогкість повітря;
- відсутність або недолік природного світла;

- підвищена пульсація світлового потоку;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційні навантаження;
- монотонність праці.

Щодо впливу на довкілля, то програмний засіб, який було розроблено під час дипломного проекту на довкілля ніяк не впливає.

5.2 Заходи щодотехнікибезпеки

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка усугубляє тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм надає на нього складну дію, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної (роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Ступінь ураження людини електричним струмом залежить від наступних факторів:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- роду і частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.

Даним проектом передбачаються наступні технічні способи і засоби, застережливі поразки людини електричним струмом:

- заземлення електроустановок;
- занулення;
- захисне відключення;
- електричне розділення сітей;
- використання малої напруги;
- ізоляція струмоведучих частин;
- огорожа електроустановок.

Проведемо розрахунок заземлюючого пристрою.

Початкові дані для розрахунку заземлюючого пристрою:

- напруга установки, що заземляється, - 220В;
- режим нейтралу мережі - з ізольованою нейтралюю;
- питомий опір ґрунту – 100 Ом·м(суглинок);
- гранично допустимий опір заземлюючого пристрою - 4 Ом;
- характеристика кліматичної зони (III):
 - а) середня багаторічна низька температура, °С- від –14 до -10;
 - б) тривалість замерзання вод, дні - 150;
 - в) коефіцієнт сезонності для вертикального електроду завдовжки 3м -1,5.

Визначимо розрахунковий опір ґрунту (Ом·м) по формулі (4.1).

$$\rho_{расч} = \psi \cdot \rho = 1,5 \cdot 100 = 150 \text{ Ом} \cdot \text{м} \quad (4.1)$$

де ρ - питомий опір ґрунту;

ψ_i – кліматичний коефіцієнт, що враховує стан ґрунту під час вимірювань (таблиця 4 [47]).

Розрахуємо опір розтіканню одиночного трубчастого заземлювача по формулі (5.2).

$$R_{з.1} = \left(\frac{\rho_{расч}}{2 \cdot \pi \cdot l} \right) \cdot \ln\left(4 \cdot \frac{l}{d}\right) \quad (4.2)$$

де l – довжина заземлювача ($l=5\text{м}$);

d – діаметр труби і стрижня ($d=0,05\text{м}$);

$$R_{з.1} = \left(\frac{\rho_{расч}}{2 \cdot \pi \cdot l} \right) \cdot \ln\left(4 \cdot \frac{l}{d}\right) = \left(\frac{150}{2 \cdot 3,14 \cdot 5} \right) \cdot \ln\left(4 \cdot \frac{5}{0,05}\right) = 28,6 \text{ Ом}$$

Розрахуємо кількість паралельно сполучених одиночних заземлювачей по формулі (4.3).

$$n = \frac{R_{з.1}}{R_{дон} \cdot \eta} = \frac{28,6}{4 \cdot 0,47} = 15,2 \quad (4.3)$$

де $R_{доп}=4$. – самий допустимий опір заземлюючого пристрою;

η - коефіцієнт використання ґрунтового заземлення (для шістки заземлювачей $\eta=0,47$).

Округлятимемо отримане значення у більшу сторону $n=[15,2]=16$.

Розрахуємо довжину горизонтальної сполучної смуги по формулі (4.4).

$$L = a \cdot (n - 1) = 3 \cdot (16 - 1) = 45 \text{ м} \quad (4.4)$$

де a – відстань між вертикальними заземлювачами ($a=3\text{м}$);

n – кількість вертикальних заземлювачей ($n=16$).

Розрахуємо опір сполучної смуги по формулі (5.5).

$$R_n = \frac{\rho_{расч}}{2 \cdot \pi \cdot l} \cdot \ln\left(\frac{L^2}{d \cdot h}\right) \quad (4.5)$$

де d – еквівалентний діаметр смуги шириною $l=5$ ($d=0,05\text{м}$);

h – глибина заставляння смуги ($h=0,8\text{м}$).

$$R_n = \frac{\rho_{расч}}{2 \cdot \pi \cdot l} \cdot \ln\left(\frac{L^2}{d \cdot h}\right) = \frac{150}{2 \cdot 3,14 \cdot 5} \cdot \ln\left(\frac{45^2}{0,05 \cdot 0,8}\right) = 51,7 \text{ Ом}$$

Розрахуємо результуючий опір заземлюючого електроду з урахуванням сполучної смуги по формулі (5.6).

$$R_{cp} = \frac{R_{з.1} \cdot R_n}{R_{з.1} \cdot \eta_n + R_n \cdot n \cdot \eta_з} \leq R_{доп} \quad (4.6)$$

де η_n – коефіцієнт використання сполучної смуги (для 6-ї заземлювачей $\eta_n=0,27$).

$$R_{cp} = \frac{R_{з.1} \cdot R_n}{R_{з.1} \cdot \eta_n + R_n \cdot n \cdot \eta_з} = \frac{26,6 \cdot 51,7}{26,6 \cdot 0,27 + 51,7 \cdot 16 \cdot 0,47} = 3,47 \text{ Ом}$$

$3,47 < 4 \Rightarrow$ умова забезпечення електробезпеки персоналу виконується.

Таким чином, остаточна кількість заземлювачей 15 шт.

4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Підвищення працездатності людини і збереження його здоров'я забезпечується стабільними метеорологічними умовами.

Мікроклімат виробничих приміщень – це поєднання температури, вогкості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і пітovidільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

В приміщенні для виконання робіт операторського типу, пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (див. таблицю 4.1).

Таблиця 4.1 - Оптимальні параметри мікроклімату в робочій зоні виробничого приміщення для категорії робіт 1

Період року	Температура, Ос	Відносна вогкість %	Швидкість руху повітря, м/с
Холодний	22.24	40.60	0,1
Теплий	23.25	40.60	0,1

Оскільки в приміщенні немає джерел виділення шкідливих речовин, можна використовувати природну вентиляцію. Площа приміщення складає 18 м². Для забезпечення прийнятних параметрів мікроклімату в приміщенні з такою площею можна використовувати 1 кондиціонер типу БК-2000.

Спектр випромінювання монітора комп'ютера включає рентгенівську, ультрафіолетову, інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння нехтує мала, оскільки цей вид випромінювання поглинається речовиною екрану.

Для зниження дії електромагнітного випромінювання пропонується захист часом і відстанню. Захист часом передбачає обмеження часу перебування людини в зоні дії полів. Тривалість роботи на ПЕОМ повинна складати не більше 3.5–4.5 години.

Також необхідно забезпечити раціональне освітлення в робочому приміщенні. В проекті, що розробляється, передбачається використовувати суміщене освітлення. В

світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Штучне освітлення в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла в світильниках загального освітлення, оскільки люмінесцентні лампи володіють високою світловою віддачею до 75 Лам/Вт і більш, тривалим терміном служби до 10000 годин, спектральним складом випромінюваного світла, близьким до сонячного.

Зорова робота оператора ПЕВМ відповідно до ДБН В.2.5-28-2018 [45] відноситься до розряду Va з світловим потоком $\Phi_{л}=3120$ кожна. Нормована освітленість на робочому місці (E_n) при загальному освітленні складає 200 лк.

Проведемо розрахунок кількості світильників в робочому приміщенні завдовжки $a=6$ м, шириною $b=3$ м, заввишки $c=4$ м. Формула розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку (54.7):

$$\Phi_{л} = \frac{E_n \cdot S \cdot Z \cdot K}{N \cdot U \cdot M} \quad (4.7)$$

де $\Phi_{л}$ – світловий потік, Лм;

E_n – нормована освітленість;

S – площа підлоги, кв.м;

$Z=1.1-1.3$ - поправочний коефіцієнт світильника (для стандартних світильників);

K – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників;

N – число світильників;

$U=0.55-0.6$ – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і др.;

M – число ламп в світильнику.

З формули (4.7) виразимо N і визначимо кількість світильників для даного приміщення:

$$N = \frac{E_n \cdot S \cdot Z \cdot K}{\Phi_{л} \cdot U \cdot M}$$

$$N = \frac{200 \cdot 18 \cdot 1,2 \cdot 1,5}{3120 \cdot 0,6 \cdot 2} = 1,7$$

Виходячи з цього, рекомендується використовувати 2 світильники. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. (5.1).

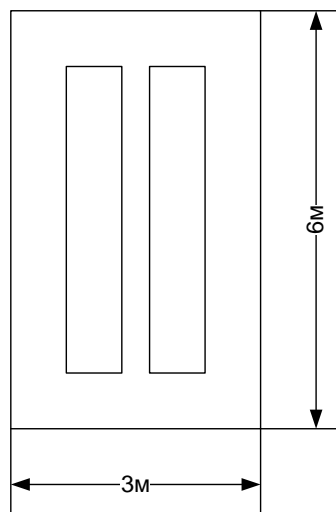


Рисунок 4.1 – Схема розташування світильників

Зниження шуму можна добитися раціонально розпланувавши приміщення, установкою устаткування на спеціальні амортизуючі прокладки. Згідно вимогам ДСН 3.3.6.037-99 [46] рівні звуку не повинні перевищувати 50 дБ.

Для зниження стомлюваності обслуговуючого персоналу в приміщеннях, де розташовані обчислювальні засоби передбачаються використовувати спокійні колірні поєднання і покриття, що не дають відблисків. Від електромагнітного випромінювання, витікаючого від ПЕОМ, використовуються захисні екрани.

Для забезпечення чистоти повітря і відповідних мікрокліматичних умов пропонується застосувати приточування-витяжну вентиляцію. Для зменшення дії шкідливих речовин і загазованості для роботи з розплавленими матеріалами робоче місце забезпечується примусовою витяжною вентиляцією. Цей метод забезпечує приток потрібної кількості свіжого повітря (30 м³ /ч на одного працюючого).

Кількість повітря, яка необхідна подавати в приміщення для забезпечення необхідних параметрів повітряного середовища, визначається на підставі кількості тепла, вологи і шкідливих речовин, що поступають в приміщення, а також враховуючи видалення повітря місцевими відсмоктуваннями від устаткування, загальнообмінною вентиляцією.

4.4 Рекомендації по пожежній профілактиці

Пожежі представляють небезпеку для життя людини і зв'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що спричиняє за собою порушення ходу технологічного процесу.

Горючими матеріалами в приміщенні, де розташовані ПЕОМ, є:

- поліамід - матеріал корпусу мікросхеми. Горюча речовина. Температура самозапалення 420 °С, енергія запалення 2мДж;
- полівінілхлорид - ізоляційний матеріал. Горюча речовина. Температура самозаймання 480 °С, енергія запалення 50мДж;
- склостоліт ДЦ - матеріал друкарської платні. Складногорючий матеріал;
- пластикат кабельний No.489 - матеріал ізоляції кабелю. Складногорючий матеріал. Температура самозаймання 1500 °С;
- плита деревостружкова - будівельний і обробний матеріал, матеріал з якого виготовлені меблі. Складнозапалений матеріал. Показник горючості 1.8;
- папір – довідкова і робоча документація, література. Горючий матеріал. Показник горючості більше 2.1.

Відповідно до ДСТУ Б В.1.1-36:2016 [48] приміщення відноситься до категорії В (пожежовибухонебезпечної).

Джерелами запалення можуть бути:

- іскри при замиканні і розмиканні ланцюгів;
- іскри і дуги коротких замикань;
- перегриви від тривалого перевантаження і наявності перехідного опору;
- розряди статичної електрики.

Для того, щоб зупинити реакцію горіння, порушують умови її виникнення і підтримки. Звичайно для гасіння використовуються порушення двох основних умов

сталого стану – пониження температури і режим руху газів. Пониження температури може бути досягнутий шляхом введення речовин, які поглинають багато тепла в результаті випаровування і дисоціації (наприклад, вода, порошки).

При повному тому, що згоряє органічних сполук утворюються С, SO, H Про, N,a при тому, що згоряє неорганічних з'єднань – оксиди. Залежно від температури плавлення і тривалості реакції можуть знаходитися або у вигляді розплавів (Al O, Ti O), або підійматися в повітря у вигляді диму (P O, Na Про, MgO).

Склад продуктів неповного згоряє горючих речовин складений і різноманітний. Це можуть бути горючі речовини:

- H, З, СН;
- атомарний водень і кисень;
- різні радикали – ВН, СН .

Продуктами неповного згоряє можуть бути також оксиди азоту, спирти, альдегіди, кетони і високотоксичні з'єднання, наприклад, синильна кислота.

Для захисту персоналу від дій небезпечних і шкідливих чинників пожежі проектом передбачено застосування промислового фільтруючого протигазу з коробкою марки В (жовтий).

До системи запобігання пожежі відносяться: запобігання утворення горючого середовища і освіти в горючому середовищі джерел запалення, забезпечення пожежебезпеки устаткування.

Щоб запобігти пожежі в обчислювальних центрах, проектом пропонується виконання наступних вимог:

- електроживлення ЕОМ має автоматичне блокування відключення електроенергії на випадок перегріву системи, що може бути результатом зупинки системи охолодження і кондиціонування;
- система вентиляції обчислювальних центрів обладнується блокуючими пристроями, що забезпечують її відключення на випадок пожежі. Система обладнується вогнеперегороджуючими клапанами;
- застосування устаткування, що задовольняє вимогам електростатичної іскробезпеки [44];
- після закінчення роботи, перед закриттям приміщення, всі електроустановки і персональні комп'ютери відключаються від сіті електроживлення;
- в приміщеннях обчислювальних центрів забороняється:
 - 1) влаштовувати електророзетки на основах, що згоряють;
 - 2) використовувати синтетичні доріжки і килими;

- 3) користуватися побутовими електронагрівальними приладами;
- 4) захарашувати евакуаційні виходи і проходи;
- 5) влаштовувати на вікнах глухі ґрати;
- 6) залишати без нагляду включену в електромережу апаратуру, що використовується для вимірювань і нагляду.

Для протипожежного захисту проектом пропонується обладнати приміщення площею 18 м², яке відноситься до категорії В, автоматичною протипожежною сигналізацією із застосуванням датчиків сповіщення РІД-1 (оповіщувач димовий іонізаційний) в кількості 1 штуки і застосовується в первинних засобах пожежегасінні. Площа контрольована оповіщувачем 150 м².

Крім того, необхідно проводити навчання робочого персоналу правилам пожежної безпеки.

Розрахуємо вірогідність виникнення пожежі у виробничому приміщенні у разі запалювання транзистора:

$$Q = l \cdot T \cdot R_{кз/отк} \cdot Q_{воспл} \cdot R_{защ} \quad (4.8)$$

де l – інтенсивність відмов пожежеопасних ЕРІ;

T – час роботи пожежеопасного ЕРІ за оцінюваний інтервал часу;

$R_{кз/отк}$ - умовна вірогідність виходу ЕРІ в стан короткого замикання при його відмові;

$Q_{воспл}$ - вірогідність запалювання ЕРІ, що знаходиться в стані короткого замикання;

$R_{защ}$ – вірогідність відмови захисту пожежеопасного ЕРІ. Якщо захист відсутній, $R_{защ}$ приймається рівній 1.

Вірогідність виникнення пожежі у разі запалювання транзистора:

$$Q = 1 \cdot 10^{-6} \cdot 1 \cdot 10^{-4} \cdot 0.1 \cdot 1 \cdot 10^{-4} = 1 \cdot 10^{-15}$$

Розрахована вірогідність виникнення пожежі значно менше допустимої, яка складає $1 \cdot 10^{-6}$.

В даному розділі були проаналізовані небезпечні і шкідливі виробничі чинники, що роблять вплив на персонал, розроблені заходи щодо техніки безпеки, заходу, забезпечуючи виробничу санітарію і гігієну праці, а також заходи щодо пожежної профілактики.

4.5 Охорона навколишнього природного середовища

Діяльність за темою магістерської роботи, а саме: оптимізація запитів до бази даних в процесі її виконання впливає на навколишнє природне середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища» [49], Законом України «Про забезпечення санітарного та епідемічного благополуччя населення» [50], Законом України «Про відходи» [51].

В процесі роботи виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- відпрацьовані люмінесцентні лампи - I клас небезпеки;
- змінні носії інформації - IV клас небезпеки;
- відходи друкуючих пристроїв - IV клас небезпеки;
- макулатура - IV клас небезпеки;
- побутові відходи - IV клас небезпеки.

ВИСНОВКИ

Аналіз сучасних принципів побудови систем моніторингу та організації обробки інформації в них показав:

- наявність суттєвої залежності ефективності застосування систем моніторингу від впливу географічних і кліматичних факторів;
- необхідність введення належного рівня обробки інформації, що міститься в повідомленнях від систем спостереження за територіальними об'єктами, як єдино можливого способу забезпечення заданих показників ефективності розпізнавання геоінформаційних ситуацій за інформацією низької якості;
- низький рівень автоматизації обробки інформації, включаючи етап побудови моделей оперативної обстановки на електронній карті.

Виявлена принципово непереборна нечіткість і неповнота інформації від систем спостереження за територією визначила необхідність введення обробки інформації як способу компенсації низької якості вектора вихідної інформації.

Розроблено узагальнену модель процесу обробки інформації в системі моніторингу територій в припущенні ідеального каналу зв'язку, що відрізняється урахуванням впливу контексту зовнішніх умов на результати багаторівневого прийняття рішень у СМТ. Це дозволяє розробити процедури, що компенсують його негативний вплив.

Розроблено узагальнену модель геоінформаційних ситуацій на основі їх інформаційних ознак, яка відрізняється способом організації простору ознак, сформованого у вигляді комбінації векторів, що описують структурні і нечіткі параметричні властивості територіальних об'єктів, а також правил їх багаторівневого узагальнення, що дозволяє описувати геоінформаційні ситуації довільної складності з урахуванням контексту зовнішніх умов.

Розроблено метод і алгоритми розпізнавання геоінформаційних ситуацій по нечіткій і неповній вихідній інформації на основі нечіткої штучної нейронної мережі, що відрізняються адаптацією функцій приналежності нечітких інформаційних ознак і їх числа відповідно до змін контексту зовнішніх умов, що дозволяє підвищити якість розпізнавання.

Щодо складності практичної реалізації способу побудови оптимальних функцій приналежності на основі байєсівського правила, в роботі запропоновано спосіб побудови квазіоптимальних функцій приналежності нечітких інформаційних ознак територіальних

об'єктів, що володіє відносною простотою їх отримання за рахунок скорочення числа розглянутих параметрів.

У зв'язку з відсутністю реально функціонуючої СМТ, оцінка ефективності запропонованих методів та алгоритмів розпізнавання геоінформаційних ситуацій по нечіткій і неповній вихідній інформації виконувалася засобами комп'ютерного моделювання.

Для цієї мети розроблено та реалізовано систему імітаційного моделювання нечіткого розпізнавача геоінформаційних об'єктів і ситуацій, що складається із сукупності модулів завдання впливу контексту зовнішніх умов, завдання структури і параметрів узагальненої моделі ТО, статистичної обробки і представлення результатів моделювання. Ядро якої реалізовано на основі QGIS 3.10.0.

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливи та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в кваліфікаційній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника. Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Була наведена схема, розміри приміщення та наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

Результати роботи були опубліковані на восьмій міжнародній наукової-конференції «Майбутній науковець 2020».

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

- 1) Творошенко, І. С., & Табашник, В. А. (2018). Розробка просторової моделі геоінформаційної підтримки людей з обмеженими можливостями, що пересуваються на інвалідних колясках, у місті Харків.
- 2) Глібовець М.М. Штучний інтелект: Підруч. для студ. вищ. навч. закладів / М.М. Глібовець, О.В. Олецкий. – Київ: Видавничий дім «КМ Академія», 2002.– 336 с.
- 3) Аникин Д.В. Краткий обзор перспектив развития интеллектуальных систем: [Електронний ресурс] / Д.В. Аникин. – Режим доступу: <http://infotech.com.ua/view-articles/id-kratkii-obzor-perspektivrazvitiya-intellektualnyh-sistem-26.htm>.
- 4) Береза А.М. Основи створення інформаційних систем: Навч. посібник / А.М. Береза. – К.: КНЕУ, 2001. – 205 с.
- 5) Глібовець М.М. Штучний інтелект: Підруч. для студ. вищ. навч. закладів / М.М. Глібовець, О.В. Олецкий. – Київ: Видавничий дім «КМ Академія», 2002.– 336 с.
- 6) Шаров С.В. Ш-26 Інтелектуальні інформаційні системи: навч. посіб. / С.В. Шаров, Д.В. Лубко, В.В. Осадчий. – Мелітополь: Вид-во МДПУ ім. Б. Хмельницького, 2015. – 144 с..
- 7) Глинський Я.М. Штучний інтелект. Інтелектуальні роботи / Я.М.Глинський, В.А. Рязька В.А. – Львів: Деол, 2002. - 168 с
- 8) Громов Ю.Ю. Интеллектуальные информационные системы и технологии: учеб. пособ. / Ю.Ю. Громов, О.Г. Иванова, В.В. Алексеев и др. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2013. – 244 с.
- 9) Джейн А. Введение в искусственные нейронные сети / Анил К. Джейн, Жианчанг Мао, Моиуддин К.М. Введение в искусственные нейронные сети // Открытые системы. – 1997. – № 140 4. – С. 16 – 24.
- 10) Интеллектуальные информационные системы и технологии: учебное пособие / Ю.Ю. Громов, О.Г. Иванова, В.В. Алексеев и др. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2013. – 244 с.
- 11) Кисіль Н.М. Класифікація інформаційних систем / Н.М. Кисіль, З.П. Гаталяк, Н.І. Горбаль // Лісове господарство, лісова, паперова і деревообробна промисловість : Міжвідомчий науково-технічний збірник. – Львів : УкрДЛТУ. – 2004. – Вип. 29. – С. 242 – 249.
- 12) Кононюк А.Ю. Нейронні мережі і генетичні алгоритми / А.Ю. Кононюк. – К.: Корнійчук, .2008. – 446 с.

- 13) Цифрова обробка сигналів в телекомунікаційних системах: підручник/ Г. Г. Бортник, В. М. Кичак. – Вінниця : ВНТУ, 2014. – 232с.
- 14) Цифрова обробка аудіо- та відеоінформації у мультимедійних системах: Навчальний посібник / О.В. Дробик, В.В. Кідалов, В.В. Коваль, Б.Я. Костік, В.С. Лазебний, Г.М. Розорінов, Г.О. Сукач. – К.: Наукова думка, 2008. – 144 с.
- 15) Цифрова обробка сигналів та зображень: навчальний посібник для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / Укладачі: Тотосько О.В., Стухляк П.Д. – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2016. – 140с
- 16) Ту Дж. Принципы распознавания образов / Ту Дж., Гонсалес Р. – М.: Мир, 1978.
- 17) Р. Гонсалес, Р. Вуде. Цифровая обработка изображений. – Техносфера.: М.: 2006. – 1072 с.
- 18) Таусенд К. Проектирование и программная реализация экспертных систем на персональных ЭВМ / К. Таусенд, Д. Фохт. – М.: Финансы и статистика, 1990. – 246 с
- 19) Системи оброблення інформації. Інтелектуальні інформаційні технології. Терміни та визначення. ДСТУ 2481 // 1994. , Державний стандарт України.
- 20) Творошенко, І. С., & Шевченко, А. Р. Практичні аспекти застосування геоінформаційного аналізу для забезпечення раціонального благоустрою міста Северодонецька.
- 21) Воронін А. М. Інформаційні системи прийняття рішень: навчальний посібник. / Воронін А. М., Зіатдінов Ю. К., Климова А. С. – К. : НАУ-друк, 2009. – 136с..
- 22) Годун В.М. Інформаційні системи і технології в статистиці: навч. посіб. / В.М. Годун, Н.С. Орленко, М. А. Сендзюк; за ред. В.Ф. Ситника. – К.: КНЕУ, 2003. – 267 с.
- 23) Інформаційні системи в промисловості : навчальний посібник / Л. О. Добровольська, О. О. Черевко. – Маріуполь : ПДТУ, 2014. – 238 с.
- 24) . Павлиш В. А. Основи інформаційних технологій і систем: Навчальний посібник. / Павлиш В. А., Гліненко Л. К. - Львів: Видавництво Львівської політехніки, 2013. – 500 с.
- 25) Пасічник В. В. Глобальні інформаційні системи та технології (моделі ефективного аналізу, опрацювання та захисту даних) / В.В. Пасічник, П.І. Жежнич, Р.Б. Кравець та ін. – Львів : Вид-во Національного університету «Львівська політехніка», 2006.- 350 с.
- 26) Авалиани, Г. В. (1988). Эвристические методы в распознавании образов. Мецниереба.

- 27) Васильев, В. И. (1969). Распознающие системы: справочник. Наукова думка.
- 28) Верхаген, К., Дейн, Р., Грун, Ф., Йостен, Й., & Вербек, П. (1985). Распознавание образов: состояние и перспективы. М.: Радио и связь, 104.
- 29) Геппенер, В. В. (2000). Математическое обеспечение многоуровневых систем распознавания сигнальной информации в условиях априорной неопределенности: Дис.... д-ра техн. наук: 05.13. 11. Санкт-Петербург. гос. электротехн. ун-т «ЛЭТИ». Изд. II Санкт-Петербург. гос. электротехн. ун-т "ЛЭТИ". СПб.
- 30) Дубров, А. М., Мхитарян, В. С., & Трошин, Л. И. (2003). Многомерные статистические методы: учебник.—М.: Финансы и статистика, 2011.—352 с.
- 31) Афанасьев, В. Н., & Юзбашев, М. М. (2001). Анализ временных рядов и прогнозирование. М.: Финансы и статистика, 228, 2.
- 32) Ашерев, А. Т., & Адаменко, А. Н. (1993). Информационно-управляющие человеко-машинные системы. Исследование, проектирование, испытания. Справочник/Под общ.ред. АИ Губинского, ВГ Евграфова.
- 33) Кофман, А. (1982). Введение в теорию нечетких множеств. М.: Радио и связь, 432.
- 34) Круглов, В. В., & Борисов, В. В. (2002). Искусственные нейронные сети. Теория и практика. М.: Горячая линия-Телеком.
- 35) Лурье, И. К. Основы геоинформатики и создание ГИС: Учеб.пособие/Дистанционное зондирование и географические информационные системы.
- 36) Мосалев, В. (2000). Системы дистанционного наблюдения за полем боя на базе разведывательно-сигнализационных приборов. Зарубежное военное обозрение, (2).
- 37) Орловский, С. А. (1981). Проблемы принятия решений при нечеткой исходной информации. Наука. Гл. ред. физ.-мат. лит.
- 38) Советов, Б. Я., & Яковлев, С. А. (2012). Моделирование систем. Общество с ограниченной ответственностью Издательство ЮРАЙТ.
- 39) Trier, O. D., Jain, A. K., & Taxt, T. (1996). Feature extraction methods for character recognition-a survey. Pattern recognition, 29(4), 641-662.
- 40) Дж, Т., & Гонсалес, Р. (1978). Принципы распознавания образов.
- 41) ДСанПіН 3.3.2-007-98 Державні санітарні правила і норми. Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин Режим Доступу- : <https://zakon.rada.gov.ua/rada/show/v0007282-98>
- 42) ДСТУ Б А.3.2-13:2011 Система стандартів безпеки праці. Будівництво. Електробезпе́чність. Загальні вимоги. Режим доступу- : <https://zakon.rada.gov.ua/laws/show/z0633-12>

43) НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями Міністерство доходів і зборів України Наказ від 05.09.2013 р. № 443 "Про затвердження Примірної інструкції з охорони праці під час експлуатації електронно-обчислювальних машин") Режим доступу- :

https://zakon.rada.gov.ua/laws/show/%D0%BD%D0%BF%D0%B0%D0%BE%D0%BF_0.00-7.15-18

44) ДСТУ 7237:2011. Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту. Режим доступу- :

<https://zakon.rada.gov.ua/laws/show/ru/z0052-13>

45) ДБН В.2.5-28-2018. Природне і штучнеосвітлення. Режим доступу - :

<https://zakon.rada.gov.ua/rada/show/v0168667-06>

46) ДСН 3.3.6.037-99 Державні санітарні норми виробничого шуму, ультразвуку та інфразвуку. Режим Доступу-: <https://zakon.rada.gov.ua/rada/show/va037282-99>

ДБН В.1.2-12-2008 Система забезпечення надійності та безпеки будівельних об'єктів. Будівництво в умовах ущільненої забудови. Вимоги безпеки. Режим доступу- :

<https://zakon.rada.gov.ua/rada/show/v0385661-08>

48) ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою». Наказ від 15.06.2016 №158. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/v0158858-16](http://www.url.com)

49) Закон України «Про охорону навколишнього природного середовища» - Режим доступу: <https://zakon.rada.gov.ua/laws/show/1264-12> - 26.06.1991 р.

50) Закони України «Про охорону навколишнього природного середовища» - Режим доступу - <https://zakon.rada.gov.ua/laws/show/4004-12> - 24.02.1994 р.

51) Закон України «Про відходи» - Режим доступу: <https://zakon.rada.gov.ua/laws/show/187/98-%D0%B2%D1%80> –

ДОДАТОК А.

Лістинг модулю «pe_search_results.py»

```
# -*- coding: utf-8 -*-

import os
import json
import sys
import logging

# noinspection PyPackageRequirements
from typing import (
    Optional,
    Any,
    List,
    Set,
)

from qgiscommons2.settings import (
    pluginSetting,
)

# noinspection PyPackageRequirements
from qgis.PyQt import uic

# noinspection PyPackageRequirements
from qgis.PyQt.QtCore import (
    pyqtSignal,
    pyqtSlot,
    Qt,
    # QObject,
    QModelIndex,
    QAbstractItemModel,
    QItemSelectionModel,
    # QItemSelection,
    QRect,
    QSize,
    QMargins,
    QTextCodec,
    QEvent,
    QPoint,
)

# noinspection PyPackageRequirements
from qgis.PyQt.QtGui import (
    QIcon,
    QCursor,
    QColor,
    QPen,
    QBrush,
    QMouseEvent,
    QKeyEvent,
    QTextDocument,
    QAbstractTextDocumentLayout,
    QPalette,
)

# noinspection PyPackageRequirements
from qgis.PyQt.QtWidgets import (
    QApplication,
```



```

QAction,
QLabel,
QFrame,
QMenu,
QToolButton,
QPlainTextEdit,
QAbstractItemView,
QTreeView,
QHeaderView,
QStyle,
QStyledItemDelegate,
QStyleOptionButton,
QStyleOptionViewItem,
# QTabWidget,
QGroupBox,
)

from qgis.core import (
    QgsApplication,
    QgsGeometry,
    # QgsGeometryCollection,
    # QgsFeature,
    QgsCoordinateReferenceSystem,
    QgsCoordinateTransform,
    QgsProject,
    QgsWkbTypes,
    QgsJsonUtils,
    QgsFields,
    QgsRectangle,
)

from qgis.gui import (
    QgsCollapsibleGroupBox,
    QgsRubberBand,
)

from planet_explorer.pe_utils import(
    ITEM_BACKGROUND_COLOR
)

plugin_path = os.path.split(os.path.dirname(__file__))[0]

if __name__ == "__main__":
    print(plugin_path)
    sys.path.insert(0, plugin_path)
    from pe_thumbnails import (
        PlanetQgisRenderJob,
    )
    # noinspection PyUnresolvedReferences
    from planet_explorer.pe_utils import (
        qgsgeometry_from_geojson,
        # qgsmultipolygon_from_geojsons,
        add_menu_section_action,
        remove_maplayers_by_name,
        zoom_canvas_to_aoi,
        preview_local_item_raster,
        # clear_local_item_raster_preview,
        create_preview_group,
        temp_preview_group,
        PE_PREVIEW,
        SETTINGS_NAMESPACE,
    )
    from gui.waiting_spinner.waitingspinnerwidget import QtWaitingSpinner

```

```

from planet_api.p_client import (
    ITEM_GROUPS,
)
from planet_api.p_node import (
    PlanetNode,
    PlanetNodeType as NodeT,

from planet_api.p_models import (
    PlanetSearchResultsModel,
)
from planet_api.p_thumbnails import (
    THUMB_EXT,
    THUMB_GEO,
)
from planet_api.p_utils import (
    geometry_from_request,
)
from planet_api.p_specs import (
    RESOURCE_DAILY,
)
else:
    from ..resources import resources
    from ..gui.waiting_spinner.waitingspinnerwidget import QtWaitingSpinner
    from .pe_thumbnails import (
        PlanetQgisRenderJob,
    )
    from .pe_utils import (
        qgsgeometry_from_geojson,
        add_menu_section_action,
        remove_maplayers_by_name,
        zoom_canvas_to_aoi,
        preview_local_item_raster,
        create_preview_group,
        temp_preview_group,
        PE_PREVIEW,
        SETTINGS_NAMESPACE,
        SEARCH_AOI_COLOR,
        PLANET_COLOR
    )
    from ..planet_api.p_client import (
        ITEM_GROUPS,
    )
    from ..planet_api.p_node import (
        PlanetNode,
        PlanetNodeType as NodeT,
    )
    from ..planet_api.p_models import (
        PlanetSearchResultsModel,
    )
    from ..planet_api.p_thumbnails import (
        THUMB_EXT,
        THUMB_GEO,
    )
    from ..planet_api.p_utils import (
        geometry_from_request,
    )
    from ..planet_api.p_specs import (
        RESOURCE_DAILY,
    )

CHILD_COUNT_THRESHOLD_FOR_PREVIEW = 500

LOG_LEVEL = os.environ.get('PYTHON_LOG_LEVEL', 'WARNING').upper()

```

```

logging.basicConfig(level=LOG_LEVEL)
log = logging.getLogger(__name__)
LOG_VERBOSE = os.environ.get('PYTHON_LOG_VERBOSE', None)

RESULTS_WIDGET, RESULTS_BASE = uic.loadUiType(
    os.path.join(plugin_path, 'ui', 'pe_search_results_base.ui'),
    from_imports=True, import_from=f'{os.path.basename(plugin_path)}',
    resource_suffix=""
)

COG_ICON = QIcon('/:plugins/planet_explorer/cog.svg')
LOCK_ICON = QIcon('/:plugins/planet_explorer/lock-light.svg')
RESPONSE_TIMEOUT = 60

class PlanetNodeItemDelegate(QStyledItemDelegate):

    previewFootprint = pyqtSignal('PyQt_PyObject')
    clearFootprint = pyqtSignal()

    def __init__(self, parent):
        super().__init__(parent)

    def paint(self, painter, option, index):
        if index.column() != 0:
            QStyledItemDelegate.paint(self, painter, option, index)
            return
        model: Any[PlanetSearchResultsModel | QAbstractItemModel] = \
            index.model()
        node: PlanetNode = model.get_node(index)
        option_v_i = QStyleOptionViewItem(option)
        self.initStyleOption(option_v_i, index)
        style = QApplication.style() \
            if option_v_i.widget is None else option_v_i.widget.style()
        opt_rect = option_v_i.rect

        doc = QTextDocument()
        doc.setHtml(option_v_i.text)

        option_v_i.text = ""
        style.drawControl(QStyle.CE_ItemViewItem, option_v_i, painter)

        ctx = QAbstractTextDocumentLayout.PaintContext()

        text_rect = style.subElementRect(QStyle.SE_ItemViewItemText, option_v_i)
        painter.save()

        painter.translate(text_rect.topLeft())
        painter.setClipRect(text_rect.translated(-text_rect.topLeft()))
        doc.documentLayout().draw(painter, ctx)

        painter.restore()

        if option.state & QStyle.State_MouseOver:
            if node.has_footprint() or node.has_group_footprint():
                painter.save()

                painter.setPen(
                    QPen(QBrush(QColor.fromRgb(0, 157, 165, 245)), 1.5))
                painter.setBrush(Qt.NoBrush)
                painter.drawRect(opt_rect.marginsRemoved(QMargins(1, 1, 1, 1)))

                painter.restore()

```

```

if node.has_footprint() or node.has_group_footprint():
    self.previewFootprint.emit(node)
else:
    self.clearFootprint.emit()

if not node.can_be_downloaded():
    painter.save()

    painter.setPen(Qt.NoPen)
    painter.setBrush(QBrush(QColor.fromRgb(250, 250, 250, 255)))
    painter.drawRoundedRect(ci_rect,
                            ci_rect.height() / 6,
                            ci_rect.height() / 6)

    LOCK_ICON.paint(painter, ci_rect, Qt.AlignCenter, QIcon.Normal)

    painter.restore()

```

```

def sizeHint(self, option, index):
    option_vi = QStyleOptionViewItem(option)
    self.initStyleOption(option_vi, index)

    doc = QTextDocument()
    doc.setHtml(option_vi.text)
    doc.setTextWidth(option_vi.rect.width())
    return QSize(int(doc.idealWidth()), int(doc.size().height()))

```

```
class PlanetNodeActionDelegate(QStyledItemDelegate):
```

```

def __init__(self, parent=None):
    super().__init__(parent)
def paint(self, painter, option, index):
    if index.column() != 1:
        QStyledItemDelegate.paint(self, painter, option, index)
        return
    model: Any[PlanetSearchResultsModel | QAbstractItemModel] = \
        index.model()
    node = model.get_node(index)
    if node.node_type() in [NodeT.LOADING, NodeT.LOAD_MORE]:
        QStyledItemDelegate.paint(self, painter, option, index)
        return
    rect = option.rect

    painter.save()

    btn = QStyleOptionButton()
    btn.icon = COG_ICON
    btn.iconSize = QSize(18, 18)
    btn.features = QStyleOptionButton.Flat
    if option.state & QStyle.State_Selected:
        painter.fillRect(option.rect, option.palette.highlight())

    btn.state = QStyle.State_Enabled

    btn.rect = QRect(rect.left() + rect.width() - 26,
                    rect.top(), 26, rect.height())

    QApplication.style().drawControl(
        QStyle.CE_PushButton, btn, painter)

    painter.restore()

```

```

class PlanetSearchResultsView(QTreeView):
    """
    """

    checkedCountChanged = pyqtSignal(int)

    def __init__(self, parent, iface=None, api_key=None,
                 request_type=None, request=None,
                 response_timeout=RESPONSE_TIMEOUT,
                 sort_order=None):
        super().__init__(parent=parent)

        self._parent: PlanetSearchResultsWidget = parent
        self._iface = iface
        self._api_key = api_key
        self._request_type = request_type
        self._request = request
        self._response_timeout = response_timeout
        self._sort_order = sort_order

        self._footprint = None
        self._setup_footprint()

        self._thumb_cache_dir: str = pluginSetting(
            'thumbCachePath', namespace=SETTINGS_NAMESPACE)

        self._checked_count = 0
        self._checked_queue = {}

        self._search_model = PlanetSearchResultsModel(
            parent=self,
            api_key=api_key,
            request_type=request_type,
            request=request,
            thumb_cache_dir=self._thumb_cache_dir,
            sort_order=self._sort_order
        )

        self._search_model.thumbnail_cache().set_job_subclass(
            PlanetQgisRenderJob
        )

        p = self.palette()
        p.setColor(QPalette.Highlight, ITEM_BACKGROUND_COLOR)
        self.setPalette(p)

        self.setModel(self._search_model)

        self.setIconSize(QSize(48, 48))
        self.setAlternatingRowColors(True)
        self.setHeaderHidden(False)
        self.setColumnWidth(1, 26)

        self.setIndentation(int(self.indentation() * 0.75))

        hv = self.header()
        hv.setStretchLastSection(False)
        hv.setSectionResizeMode(0, QHeaderView.Stretch)
        hv.setSectionResizeMode(1, QHeaderView.Fixed)
        if len(sort_order) > 1:
            if sort_order[1] == 'asc':
                sort_indicator = Qt.AscendingOrder
            else:

```

```

        sort_indicator = Qt.DescendingOrder
        hv.setSortIndicator(0, sort_indicator)
        hv.setSortIndicatorShown(True)

self.viewport().setAttribute(Qt.WA_Hover)
self.setMouseTracking(True)
self.setSelectionMode(QAbstractItemView.ExtendedSelection)
self.setSelectionBehavior(QAbstractItemView.SelectRows)

self.item_delegate = PlanetNodeItemDelegate(parent=self)
self.item_delegate.previewFootprint['PyQt_PyObject'].connect(
    self.preview_footprint)

self.item_delegate.clearFootprint.connect(self.clear_footprint)
self.setItemDelegateForColumn(0, self.item_delegate)
self.act_delegate = PlanetNodeActionDelegate(parent=self)
self.setItemDelegateForColumn(1, self.act_delegate)

self.setContextMenuPolicy(Qt.CustomContextMenu)
self.customContextMenuRequested['QPoint'].connect(self.open_menu)
self.clicked['QModelIndex'].connect(self.item_clicked)
self.expanded['QModelIndex'].connect(self.item_expanded)

def search_model(self):
    return self._search_model

def checked_count(self):
    return self._checked_count

def checked_queue(self):
    return self._checked_queue

def _setup_footprint(self):
    if self._iface:
        log.debug('iface is available, adding footprint support')
        self._footprint = QgsRubberBand(
            self._iface.mapCanvas(), QgsWkbTypes.PolygonGeometry)
        self._footprint.setFillColor(QColor(255, 255, 255, 10))
        self._footprint.setStrokeColor(PLANET_COLOR)
        self._footprint.setWidth(2)
    else:
        log.debug('iface is None, skipping footprint support')
        self._footprint = None

def qgsfeature_feature_from_node(self, node: PlanetNode):

    feature_collect = {
        'type': 'FeatureCollection',
        'features': [
            node.resource()
        ]
    }

    feature_collect_json = json.dumps(feature_collect)
    features = []
    try:
        utf8 = QTextCodec.codecForName("UTF-8")
        fields = QgsFields()
        features = QgsJsonUtils().stringToFeatureList(
            string=feature_collect_json, fields=fields, encoding=utf8)
    except Exception:
        log.debug('Footprint GeoJSON could not be parsed')
        return

```

```

if not len(features) > 0:
    log.debug('GeoJSON parsing created no features')
    return

return features[0]

@pyqtSlot()
def clear_footprint(self):
    if self._footprint:
        self._footprint.reset(QgsWkbTypes.PolygonGeometry)

@pyqtSlot('PyQt_PyObject')
def preview_footprint(self, node: PlanetNode):
    if not self._footprint:
        if LOG_VERBOSE:
            log.debug('Footprint is None, skipping footprint preview')
        return

    if node.has_group_footprint():
        geoms = node.geometries()
    else:
        geoms = [node.geometry()]

    self.clear_footprint()

    qgs_geoms = [qgsgeometry_from_geojson(g) for g in geoms]

    for qgs_geom in qgs_geoms:
        self._footprint.addGeometry(
            qgs_geom,
            QgsCoordinateReferenceSystem("EPSG:4326")
        )

    if LOG_VERBOSE:
        log.debug('Footprint sent to canvas')

@pyqtSlot(list)
def zoom_to_footprint(self, nodes: [PlanetNode]):
    skip = 'skipping zoom to footprint'
    if not self._footprint:
        log.debug(f'Footprint is None, {skip}')
        return

    if len(nodes) < 1:
        log.debug('No nodes available, skipping zoom to footprint')
        return

    first_node = nodes[0]
    if first_node.has_group_footprint():
        json_geoms = first_node.geometries()
    else:
        json_geoms = [node.geometry() for node in nodes]

    qgs_geoms: [QgsGeometry] = \
        [qgsgeometry_from_geojson(j) for j in json_geoms]

    if len(qgs_geoms) < 1:
        log.debug(f'Geometry collection empty, {skip}')
        return

    rect_geoms: QgsRectangle = qgs_geoms[0].boundingBox()
    for i in range(len(qgs_geoms)):

```

```

    if i == 0:
        continue
    r: QgsRectangle = QgsGeoms[i].boundingBox()
    rect_geoms.combineExtentWith(r)

if rect_geoms.isNull():
    log.debug(f'Footprint geometry is null, {skip}')
    return

# noinspection PyArgumentList
transform = QgsCoordinateTransform(
    QgsCoordinateReferenceSystem("EPSG:4326"),
    QgsProject.instance().crs(),
    QgsProject.instance()
)
rect_footprint: QgsRectangle = \
    transform.transformBoundingBox(rect_geoms)

if not rect_footprint.isEmpty():
    rect_footprint.scale(1.05)
    self._iface.mapCanvas().setExtent(rect_footprint)
    self._iface.mapCanvas().refresh()

@pyqtSlot('PyQt_PyObject')
def preview_thumbnail(
    self, node, name=PE_PREVIEW, remove_existing=True):
    item_name_geo = f'{node.item_type_id_key()} {THUMB_GEO}'
    item_geo_path = os.path.join(
        self.search_model().thumbnail_cache().cache_dir(),
        f'{item_name_geo} {THUMB_EXT}')
    if not preview_local_item_raster(
        item_geo_path, name, remove_existing=remove_existing):
        log.warning(f'Item preview {item_name_geo} failed to load')

@pyqtSlot(dict)
def update_preview_thumbnails(self, selected_nodes):
    group = temp_preview_group()

    tree_node_names: List[str] = \
        [t_node.name() for t_node in group.children()]
    for name in tree_node_names:
        if name.endswith('_thumb'):
            name_sans = name.replace('_thumb', '')
            if name_sans not in selected_nodes:
                remove_maplayers_by_name(name, only_first=True)
            if name_sans in selected_nodes:
                # Keep already loaded layer, skip reloading it
                selected_nodes[name_sans] = None

    for _, node in selected_nodes.items():
        if node is not None:
            self.preview_thumbnail(node)

@pyqtSlot(list)
def add_preview_groups(self, nodes: List[PlanetNode]):
    if len(nodes) < 1 or not isinstance(nodes[0], PlanetNode):
        log.debug('No nodes found to add to preview group')
        return

    if nodes[0].is_group_node():
        log.debug('Adding preview group for group')
        # Grouping tree items are only singularly passed
        self.add_preview_group_for_group(nodes[0])
    else:

```



```

log.debug('Adding preview group for items')
self.add_preview_groups_for_items(nodes)

@pyqtSlot(list)
def add_preview_group_for_group(self, node: PlanetNode):
    name = None
    child_node_type = None
    if node.node_type() == NodeT.DAILY_SCENE:
        child_node_type = NodeT.DAILY_SCENE_IMAGE
        item_type = node.name() or ""
        title = ['Daily', item_type, 'Scene']
        name = f{" ".join(title)} '\
            f{node.formatted_date_time(node.sort_date())}'
    elif node.node_type() == NodeT.DAILY_SAT_GROUP:
        child_node_type = NodeT.DAILY_SCENE_IMAGE
        item_type = node.parent().name() or ""
        title = ['Daily', item_type, f'Satellite {node.name()} Group']
        name = f{" ".join(title)} '\
            f{node.formatted_date_time(node.sort_date())}'

    if child_node_type is None:
        log.debug('No node type found for tree group searching')
        return

    item_nodes = node.children_of_node_type(child_node_type)
    if item_nodes:
        create_preview_group(
            name, item_nodes,
            self._search_model.p_client().api_key(),
            tile_service='xyz',
            search_query=self._request,
            sort_order=self._sort_order
        )
    else:
        log.debug(f"No items found for node type '{child_node_type.name}' "
            f"in tree group '{name}'")

@pyqtSlot(list)
def add_preview_groups_for_items(self, nodes: List[PlanetNode]) -> None:
    prev_types = [] # maintain some sort order
    prev_type_nodes = {}
    for node in nodes:
        item_type = node.item_type()
        if item_type not in prev_types:
            prev_types.append(item_type)
            prev_type_nodes[item_type] = []
        prev_type_nodes[item_type].append(node)

    for prev_type in sorted(prev_types):
        prev_nodes: List[PlanetNode] = prev_type_nodes[prev_type]
        create_preview_group(
            prev_type, prev_nodes,
            self._search_model.p_client().api_key(),
            tile_service='xyz',
            search_query=self._request,
            sort_order=self._sort_order
        )

@pyqtSlot(list)
def copy_ids_to_clipboard(self, nodes):
    node_ids = [n.item_type_id() for n in nodes if n.item_id()]
    if node_ids:
        cb = QgsApplication.clipboard()

```

```

        cb.setText(', '.join(node_ids))

@pyqtSlot('QPoint')
def open_menu(self, pos):
    """
    :type pos: QPoint
    :return:
    """
    index = self.indexAt(pos)
    node: PlanetNode = self.model().get_node(index)
    if (node.node_type() == NodeT.LOAD_MORE
        and node.parent() == self.model().root):
        return
    menu = QMenu()

    # Single, current Item's index
    add_menu_section_action('Current item', menu)

    if node.has_footprint() or node.has_group_footprint():
        zoom_fp_act = QAction('Zoom to footprint', menu)
        zoom_fp_act.triggered[bool].connect(
            lambda: self.zoom_to_footprint([node]))
        menu.addAction(zoom_fp_act)

    if node.can_load_preview_layer():
        prev_layer_act = QAction('Add preview layer to map', menu)
        prev_layer_act.triggered[bool].connect(
            lambda: self.add_preview_groups([node]))
        if node.child_images_count() > CHILD_COUNT_THRESHOLD_FOR_PREVIEW:
            prev_layer_act.setEnabled(False)
            prev_layer_act.setToolTip("The node contains too many images to preview")
            menu.setToolTipsVisible(True)

        menu.addAction(prev_layer_act)

    if node.item_id() and node.has_resource():
        copy_id_act = QAction('Copy ID to clipboard', menu)
        # noinspection PyUnresolvedReferences
        copy_id_act.triggered[bool].connect(
            lambda: self.copy_ids_to_clipboard([node]))
        menu.addAction(copy_id_act)
    sel_model = self.selectionModel()
    model = self.model()
    all_nodes = [model.get_node(i) for i in sel_model.selectedIndexes()
                 if i.column() == 0]
    log.debug(f'Selected items: {len(all_nodes)}')
    if len(all_nodes) == 1 and all_nodes[0] == node:
        menu.exec_(self.viewport().mapToGlobal(pos))
        return
    nodes_have_footprints = \
        [node for node in all_nodes if node.has_footprint()]
    nodes_w_ids = \
        [node for node in all_nodes
         if node.item_id() and node.has_resource()]
    nodes_can_prev = \
        [node for node in all_nodes
         if node.can_load_preview_layer() and node.has_resource()]

    if any([nodes_have_footprints, nodes_w_ids, nodes_can_prev]):
        add_menu_section_action(f'Selected images', menu)

    if nodes_have_footprints:
        zoom_fps_act = QAction(

```

```

        f'Zoom to total footprint '
        f'({len(nodes_have_footprints)} items)', menu)
    zoom_fps_act.triggered[bool].connect(
        lambda: self.zoom_to_footprint(nodes_have_footprints))
    menu.addAction(zoom_fps_act)

if nodes_can_prev:
    prev_layers_act = QAction(
        f'Add preview layer to map '
        f'({len(nodes_can_prev)} items)', menu)
    prev_layers_act.triggered[bool].connect(
        lambda: self.add_preview_groups(nodes_can_prev))
    menu.addAction(prev_layers_act)

if nodes_w_ids:
    copy_ids_act = QAction(
        f'Copy IDs to clipboard ({len(nodes_w_ids)} items)', menu)
    copy_ids_act.triggered[bool].connect(
        lambda: self.copy_ids_to_clipboard(nodes_w_ids))
    menu.addAction(copy_ids_act)
    menu.exec_(self.viewport().mapToGlobal(pos))

@pyqtSlot("QModelIndex")
def item_clicked(self, index):
    node: PlanetNode = self.model().get_node(index)
    log.debug(f'Index clicked: row {index.row()}, col {index.column()}, '
              f'{node.item_type_id()}'')
    if index.column() == 0:
        if (node.node_type() == NodeT.LOAD_MORE
            and node.parent() == self.model().root):
            self.model().fetch_more_top_items(index)
    elif index.column() == 1:
        self.open_menu(self.viewport().mapFromGlobal(QCursor.pos()))

@pyqtSlot("QModelIndex")
def item_expanded(self, index: QModelIndex):
    node: PlanetNode = self.model().get_node(index)
    log.debug(f'Index expanded: row {index.row()}, col {index.column()}, '
              f'{node.item_type_id()}'')
def _update_checked_queue(self,
                          checked_nodes: Set[PlanetNode],
                          unchecked_nodes: Set[PlanetNode]):
    for c_node in checked_nodes:
        it_id = c_node.item_type_id()
        self._checked_queue[it_id] = c_node

    for u_node in unchecked_nodes:
        it_id = u_node.item_type_id()
        if it_id in self._checked_queue:
            del self._checked_queue[it_id]

    self._checked_count = len(self._checked_queue)
    log.debug(f'checked_count: {self._checked_count}'')

if LOG_VERBOSE:
    sorted_item_ids = sorted(self._checked_queue.keys())
    nl = '\n'
    log.debug(f'checked_queue:\n'
              f' {"{0} ".format(nl).join(sorted_item_ids)}')

    self.checkedCountChanged.emit(self._checked_count)

def event(self, event: QEvent) -> bool:

```

```

if event.type() == QEvent.Leave:
    if self._iface:
        self.clear_footprint()
        event.accept()

return QTreeView.event(self, event)

def mousePressEvent(self, event: QMouseEvent) -> None:
    index = self.indexAt(event.pos())
    sel_model: QItemSelectionModel = self.selectionModel()
    if (index.column() == 1
        and event.button() == Qt.LeftButton
        and sel_model.isSelected(index)):
        log.debug('Ignoring mouse press')
        return

return QTreeView.mousePressEvent(self, event)

def mouseReleaseEvent(self, event: QMouseEvent) -> None:
    index = self.indexAt(event.pos())
    sel_model: QItemSelectionModel = self.selectionModel()
    if (index.column() == 1
        and event.button() == Qt.LeftButton
        and sel_model.isSelected(index)):
        log.debug('Swapping left button for right, on release')
        self.open_menu(event.pos())
        return

return QTreeView.mouseReleaseEvent(self, event)

def keyPressEvent(self, event: QKeyEvent) -> None:
    index = self.currentIndex()
    if (index.column() == 0
        and event.key() in [Qt.Key_Enter, Qt.Key_Return]):
        node: PlanetNode = self.model().get_node(index)
        if (node.node_type() == NodeT.LOAD_MORE
            and node.parent() == self.model().root():
            self.model().fetch_more_top_items(index)

return QTreeView.keyPressEvent(self, event)

def dataChanged(self, t_l: QModelIndex, b_r: QModelIndex,
                roles: Optional[List[int]] = None) -> None:

    # This may need to go at end of slot
    super().dataChanged(t_l, b_r, roles=roles)

    # Note: empty roles indicates *everything* has changed, not nothing
    if roles is not None and (roles == [] or Qt.CheckStateRole in roles):
        if LOG_VERBOSE:
            log.debug('Node (un)checked')
            checked = set()
            unchecked = set()

        def update_queues(indx):
            node: PlanetNode = self._search_model.get_node(indx)
            if node.is_base_image() and node.can_be_downloaded():
                if node.checked_state() == Qt.Checked:
                    checked.add(node)
                elif node.checked_state() == Qt.Unchecked:
                    unchecked.add(node)

    # Note: if parent of t_l and b_r differ, we ignore undefined input

```

```

if t_l == b_r:
    if LOG_VERBOSE:
        log.debug('Nodes (un)checked is single')
    update_queues(t_l)
elif t_l.parent() == b_r.parent():
    if LOG_VERBOSE:
        log.debug('Nodes (un)checked have same parent')
    parent = t_l.parent()
    col = t_l.column()
    for row in range(b_r.row(), t_l.row() + 1):
        m_idx = self._search_model.index(row, col, parent=parent)
        update_queues(m_idx)

if checked or unchecked:
    if LOG_VERBOSE:
        log.debug(f'Nodes checked: {checked}')
        log.debug(f'Nodes unchecked: {unchecked}')
    self._update_checked_queue(checked, unchecked)

@pyqtSlot()
def clean_up(self):
    self.clear_footprint()

class PlanetSearchResultsWidget(RESULTS_BASE, RESULTS_WIDGET):
    """
    """

    zoomToAOIRequested = pyqtSignal()
    setAOIRequested = pyqtSignal(dict)
    setSearchParamsRequested = pyqtSignal(dict, tuple)
    checkedCountChanged = pyqtSignal(int)

    grpBoxQuery: QGroupBox
    teQuery: QPlainTextEdit
    frameSearching: QFrame
    btnCancel: QToolButton
    btnShowQuery: QToolButton
    btnZoomToAOI: QToolButton
    btnSetAOI: QToolButton
    btnSetSearchParams: QToolButton
    lblSearching: QLabel
    frameResults: QFrame

    def __init__(self, parent=None, iface=None, api_key=None,
                 request_type=None, request=None,
                 response_timeout=RESPONSE_TIMEOUT,
                 sort_order=None):
        super().__init__(parent=parent)

        self.setupUi(self)

        self._parent = parent

        self._iface = iface
        self._response_timeout = response_timeout
        self._request_type = request_type
        self._request = request
        self.sort_order = sort_order
        self.teQuery.setPlainText(json.dumps(request, indent=2))
        self.grpBoxQuery.hide()
        self.btnZoomToAOI.clicked.connect(self._zoom_to_request_aoi)
        self.btnSetAOI.clicked.connect(self._set_aoi_from_request)
        self.btnSetSearchParams.clicked.connect(self._set_search_params_from_request)

```

```

self.btnShowQuery.clicked.connect(self._toggle_query)
self._aoi_box = None
self._setup_request_aoi_box()
self.results_tree = PlanetSearchResultsView(
    parent=self, iface=iface, api_key=api_key,
    request_type=request_type, request=request,
    response_timeout=self._response_timeout,
    sort_order=sort_order
)
self.results_tree.checkedCountChanged[int].connect(
    self.checked_count_changed)
self.frameResults.layout().addWidget(self.results_tree)
self.results_tree.setHidden(True)
search_model = self.results_tree.search_model()
if self.results_tree.search_model():
    search_model.searchFinished.connect(self._search_finished)
    search_model.searchNoResults.connect(self._search_no_results)
    search_model.searchCancelled.connect(self._search_cancelled)
    search_model.searchTimedOut[int].connect(self._search_timed_out)
    self.btnCancel.clicked.connect(search_model.cancel_search)
self.waiting_spinner = QtWaitingSpinner(
    self.frameResults,
    centerOnParent=True,
    disableParentWhenSpinning=False,
    modality=Qt.NonModal
)
self.waiting_spinner.setRoundness(80.0)
self.waiting_spinner.setMinimumTrailOpacity(15.0)
self.waiting_spinner.setTrailFadePercentage(75.0)
self.waiting_spinner.setNumberOfLines(15)
self.waiting_spinner.setLineLength(14.0)
self.waiting_spinner.setLineWidth(3.0)
self.waiting_spinner.setInnerRadius(8.0)
self.waiting_spinner.setRevolutionsPerSecond(1.0)
self.waiting_spinner.setColor(PLANET_COLOR)
self.waiting_spinner.start()
self.waiting_spinner.show()
search_model.start_api_search()
def _nix_waiting_spinner(self):
    self.waiting_spinner.stop()
    self.waiting_spinner.hide()

def checked_count(self):
    return self.results_tree.checked_count()

def checked_queue(self):
    return self.results_tree.checked_queue()

@pyqtSlot(int)
def checked_count_changed(self, count):
    if type(self._parent).__name__ == 'QTabWidget':
        tab_indx = self._parent.indexOf(self)

        if tab_indx != -1:
            txt = f'{self._request_type.capitalize()}'
            if count > 0:
                txt += f' ({count})'
            self._parent.setTabText(tab_indx, txt)

    self.checkedCountChanged.emit(count)

@pyqtSlot()
def _toggle_query(self):

```

```

if self.grpBoxQuery.isVisible():
    self.grpBoxQuery.hide()
    self.btnShowQuery.setIcon(
        QIcon('/:plugins/planet_explorer/expand-triangle.svg')
    )
else:
    self.grpBoxQuery.show()
    self.btnShowQuery.setIcon(
        QIcon('/:plugins/planet_explorer/collapse-triangle.svg')
    )

@pyqtSlot()
def _search_finished(self):
    self._nix_waiting_spinner()
    self.frameSearching.hide()
    self.results_tree.show()

@pyqtSlot()
def _search_no_results(self):
    self._nix_waiting_spinner()
    self.lblSearching.setText('No results found')
    self.btnCancel.hide()
    self.results_tree.hide()

@pyqtSlot()
def _search_cancelled(self):
    self._nix_waiting_spinner()
    self.lblSearching.setText('Search cancelled')
    self.btnCancel.hide()
    self.results_tree.hide()

@pyqtSlot(int)
def _search_timed_out(self, timeout):
    self._nix_waiting_spinner()
    self.lblSearching.setText(f'Search timed out ({timeout} seconds)')
    self.btnCancel.hide()
    self.results_tree.hide()

def _setup_request_aoi_box(self):
    if self._iface:
        log.debug('iface is available, adding aoi box support')
        self._aoi_box = QgsRubberBand(
            self._iface.mapCanvas(), QgsWkbTypes.PolygonGeometry)
        self._aoi_box.setFillColor(QColor(0, 0, 0, 0))
        self._aoi_box.setStrokeColor(SEARCH_AOI_COLOR)
        self._aoi_box.setWidth(2)
        self._aoi_box.setLineStyle(Qt.DashLine)
    else:
        log.debug('iface is None, skipping footprint support')
        self._aoi_box = None

@pyqtSlot()
def clear_aoi_box(self):
    if self._aoi_box:
        self._aoi_box.reset(QgsWkbTypes.PolygonGeometry)

@pyqtSlot()
def _zoom_to_request_aoi(self):
    if not self._iface:
        log.debug('No iface object, skipping AOI extent')
        return

    aoi_geom = None

```

```

if self._request_type == RESOURCE_DAILY:
    aoi_geom = geometry_from_request(self._request)

if not aoi_geom:
    log.debug('No AOI geometry defined, skipping zoom to AOI')
    return

qgs_geom: QgsGeometry = qgsgeometry_from_geojson(aoi_geom)
self._aoi_box.setToGeometry(
    qgs_geom,
    QgsCoordinateReferenceSystem("EPSG:4326")
)

self.show_aoi()

zoom_canvas_to_aoi(aoi_geom, iface_obj=self._iface)

self.zoomToAOIRequested.emit()

def hide_aoi_if_matches_geom(self, geom):
    if self._aoi_box is not None:
        color = (QColor(0, 0, 0, 0) if self._aoi_box.asGeometry().equals(geom)
                else SEARCH_AOI_COLOR)
        self._aoi_box.setStrokeColor(color)

def show_aoi(self):
    if self._aoi_box is not None:
        self._aoi_box.setStrokeColor(SEARCH_AOI_COLOR)

def aoi_geom(self):
    if self._aoi_box is not None:
        return self._aoi_box.asGeometry()

@pyqtSlot()
def _set_aoi_from_request(self):
    self.setAOIRequested.emit(self._request)

@pyqtSlot()
def _set_search_params_from_request(self):
    self.setSearchParamsRequested.emit(self._request, self.sort_order)

@pyqtSlot()
def clean_up(self):
    self.results_tree.clean_up()
    self.clear_aoi_box()

# noinspection PyPep8Naming
def closeEvent(self, event):
    self.clean_up()
    super().closeEvent(self, event)

def request_query(self):
    return self._request

if __name__ == "__main__":
    from qgis.PyQt.QtWidgets import QDialog, QVBoxLayout
    from qgiscommons2.settings import (
        readSettings,
    )

    sys.path.insert(0, plugin_path)

```



```

# noinspection PyUnresolvedReferences,PyPackageRequirements
from planet_explorer.gui.resources import qgis_resources

apikey = os.getenv('PL_API_KEY', None)
if not apikey:
    log.debug('No API key in environ')
    sys.exit(1)

try:
    argvb = list(map(os.fsencode, sys.argv))
except AttributeError:
    argvb = sys.argv

# Create a reference to the QgsApplication. Setting the
# second argument to False disables the GUI.
qgs = QgsApplication(argvb, True)

# Load providers
qgs.initQgis()

search_request = json.loads("""{"item_types":
["PSScene4Band", "PSScene3Band"],
"filter": {"type": "AndFilter", "config": [{"field_name": "geometry",
"type": "GeometryFilter", "config": {"type": "Polygon", "coordinates": [
[[-124.60010884388858, 36.207866384307614], [-119.61878664869495,
36.207866384307614], [-119.61878664869495, 39.705780131667844],
[-124.60010884388858, 39.705780131667844], [-124.60010884388858,
36.207866384307614]]]}}, {"field_name": "cloud_cover", "type":
"RangeFilter", "config": {"gte": 0, "lte": 100}}]}""")

readSettings(settings_path=os.path.join(plugin_path, 'settings.json'))
SETTINGS_NAMESPACE = None
wdgt = PlanetSearchResultsWidget(
    parent=None, iface=None, api_key=apikey,
    request_type=RESOURCE_DAILY, request=search_request,
    sort_order=('acquired', 'desc'))

dlg = QDialog()
layout = QVBoxLayout(dlg)
image_lbl = QLabel(dlg)
layout.addWidget(wdgt)
dlg.setMinimumHeight(700)
dlg.exec_()
qgs.exitQgis()
sys.exit(0)

```

ДОДАТОК Б.

Лістинг модулю «mosaic_layer_widgets.py»

```

# -*- coding: utf-8 -*-
import os
import json
from urllib.parse import quote

from PyQt5.QtCore import Qt, QRectF

from PyQt5.QtGui import (
    QPainter,
    QBrush,
    QColor
)

from PyQt5.QtWidgets import (
    QWidget,
    QSlider,
    QVBoxLayout,
    QHBoxLayout,
    QComboBox,
    QLabel,
    QStyle,
    QStyleOptionSlider,
    QStyleFactory
)

from qgis.core import (
    QgsMapLayer,
    QgsDataProvider,
    QgsLayerTreeLayer,
    QgsLayerTreeGroup,
    QgsProject
)

from qgis.gui import (
    QgsLayerTreeEmbeddedWidgetProvider,
    QgsLayerTreeEmbeddedWidgetRegistry
)

from ..pe_utils import PLANET_COLOR

from ..planet_api import PlanetClient

PLANET_CURRENT_MOSAIC = "planet/currentMosaic"
PLANET_MOSAICS = "planet/mosaics"
PLANET_MOSAIC_PROC = "planet/mosaicProc"
PLANET_MOSAIC_RAMP = "planet/mosaicRamp"
PLANET_MOSAIC_DATATYPE = "planet/mosaicDatatype"

TILE_URL_TEMPLATE = "https://tiles.planet.com/basemaps/v1/planet-
tiles/%s/gmap/{z}/{x}/{y}.png?api_key=%s"

class CustomSlider(QSlider):

    def paintEvent(self, event):

```

```

# based on
# http://qt.gitorious.org/qt/qt/blobs/master/src/gui/widgets/qslider.cpp

painter = QPainter(self)
style = self.style()
opt = QStyleOptionSlider()
self.initStyleOption(opt)

groove_rect = style.subControlRect(
    style.CC_Slider, opt, QStyle.SC_SliderGroove, self)
handle_rect = style.subControlRect(
    style.CC_Slider, opt, QStyle.SC_SliderHandle, self)

slider_space = style.pixelMetric(style.PM_SliderSpaceAvailable, opt)
range_x = style.sliderPositionFromValue(
    self.minimum(), self.maximum(), self.value(), slider_space)
range_height = 4

groove_rect = QRectF(
    groove_rect.x(),
    handle_rect.center().y() - (range_height / 2),
    groove_rect.width(),
    range_height)

range_rect = QRectF(
    groove_rect.x(),
    handle_rect.center().y() - (range_height / 2),
    range_x,
    range_height)

if style.metaObject().className() != 'QMacStyle':
    cur_brush = painter.brush()
    cur_pen = painter.pen()
    painter.setBrush(QBrush(QColor(169, 169, 169)))
    painter.setPen(Qt.NoPen)
    # painter.drawRect(groove_rect)
    painter.drawRoundedRect(groove_rect,
        groove_rect.height() / 2,
        groove_rect.height() / 2)
    painter.setBrush(cur_brush)
    painter.setPen(cur_pen)

cur_brush = painter.brush()
cur_pen = painter.pen()
painter.setBrush(QBrush(QColor(18, 141, 148)))
painter.setPen(Qt.NoPen)
painter.drawRect(range_rect)
painter.setBrush(cur_brush)
painter.setPen(cur_pen)

opt = QStyleOptionSlider()
self.initStyleOption(opt)

opt.subControls = QStyle.SC_SliderHandle

if self.tickPosition() != self.NoTicks:
    opt.subControls |= QStyle.SC_SliderTickmarks

if self.isSliderDown():
    opt.state |= QStyle.State_Sunken
else:
    opt.state |= QStyle.State_Active

```

```
opt.activeSubControls = QStyle.SC_None
```

```
opt.sliderPosition = self.value()
```

```
opt.sliderValue = self.value()
```

```
style.drawComplexControl(
```

```
    QStyle.CC_Slider, opt, painter, self)
```

```
class MosaicLayerWidget(QWidget):
```

```
    def __init__(self, layer):
```

```
        super().__init__()
```

```
        self.current_mosaic_name = layer.customProperty(PLANET_CURRENT_MOSAIC)
```

```
        proc = layer.customProperty(PLANET_MOSAIC_PROC)
```

```
        ramp = layer.customProperty(PLANET_MOSAIC_RAMP)
```

```
        self.datatype = layer.customProperty(PLANET_MOSAIC_DATATYPE)
```

```
        self.layer = layer
```

```
        self.mosaics = json.loads(layer.customProperty(PLANET_MOSAICS))
```

```
        self.mosaicnames = [m[0] for m in self.mosaics]
```

```
        self.mosaicids = [m[1] for m in self.mosaics]
```

```
        self.layout = QVBoxLayout()
```

```
        self.labelId = QLabel()
```

```
        self.layout.addWidget(self.labelId)
```

```
        if len(self.mosaics) > 1:
```

```
            idx = self.mosaicnames.index(self.current_mosaic_name)
```

```
            self.labelName = QLabel(f"{self.current_mosaic_name}")
```

```
            self.slider = CustomSlider(Qt.Horizontal)
```

```
            self.slider.setRange(0, len(self.mosaics) - 1)
```

```
            self.slider.setTickInterval(1)
```

```
            self.slider.setTickPosition(QSlider.TicksAbove)
```

```
            self.slider.setPageStep(1)
```

```
            self.slider.setTracking(True)
```

```
            self.slider.setEnabled(True)
```

```
            self.slider.setValue(idx)
```

```
            self.slider.valueChanged.connect(self.on_value_changed)
```

```
            self.slider.sliderReleased.connect(self.change_source)
```

```
            self.layout.addWidget(self.labelName)
```

```
            self.layout.addWidget(self.slider)
```

```
        else:
```

```
            idx = 0
```

```
        self.labelId.setText(f'<span style="color: grey;">{self.mosaicids[idx]}</span>')
```

```
        self.hlayoutProc = QHBoxLayout()
```

```
        self.labelProc = QLabel("Processing:")
```

```
        self.comboProc = QComboBox()
```

```
        self.comboProc.addItem(self.processes_for_datatype())
```

```
        self.comboProc.setCurrentText(proc)
```

```
        self.hlayoutProc.addWidget(self.labelProc)
```

```
        self.hlayoutProc.addWidget(self.comboProc)
```

```
        self.load_ramps()
```

```
        self.hlayoutRamp = QHBoxLayout()
```

```
        self.labelRamp = QLabel("Color ramp:")
```

```
        self.comboRamp = QComboBox()
```

```
        for r in self.ramps:
```

```
            self.comboRamp.addItem(r)
```

```
        self.comboRamp.setCurrentText(ramp)
```

```
        self.comboRamp.setVisible(self.can_use_indices())
```

```
        self.labelRamp.setVisible(self.can_use_indices())
```

```
        self.hlayoutRamp.addWidget(self.labelRamp)
```

```
        self.hlayoutRamp.addWidget(self.comboRamp)
```

```
        self.layout.addLayout(self.hlayoutProc)
```

```
        self.layout.addLayout(self.hlayoutRamp)
```

```
        self.labelWarning = QLabel('<span style="color:red;"><b>No API key available</b></span>')
```

```

self.layout.addWidget(self.labelWarning)
self.setLayout(self.layout)

self.comboProc.currentIndexChanged.connect(self.change_source)
self.comboRamp.currentIndexChanged.connect(self.change_source)

PlanetClient.getInstance().loginChanged.connect(self.login_changed)

self.change_source()

def load_ramps(self):
    path = os.path.join(os.path.dirname(os.path.dirname(__file__)),
                        "resources", "mosaics_caps.json")
    with open(path) as f:
        caps = json.load(f)

    self.ramps = list(caps["colors"].keys())

def on_value_changed(self, value):
    self.labelId.setText(f'<span style="color: grey;">{self.mosaicids[value]}</span>')
    self.labelName.setText(f'{self.mosaicnames[value]}')
    if not self.slider.isSliderDown():
        self.change_source()

def can_use_indices(self):
    return self.datatype == "uint16"

def change_source(self):
    has_api_key = PlanetClient.getInstance().has_api_key()
    self.labelWarning.setVisible(not has_api_key)
    self.labelId.setVisible(has_api_key)
    self.labelProc.setVisible(has_api_key)
    self.labelRamp.setVisible(has_api_key and self.can_use_indices())
    if len(self.mosaics) > 1:
        self.labelName.setVisible(has_api_key)
        self.slider.setVisible(has_api_key)
        self.comboProc.setVisible(has_api_key)
        self.comboRamp.setVisible(has_api_key and self.can_use_indices())
        value = self.slider.value() if len(self.mosaics) > 1 else 0
        proc = self.comboProc.currentText()
        ramp = self.comboRamp.currentText()
        name, mosaicid = self.mosaics[value]
        tile_url = TILE_URL_TEMPLATE % (mosaicid, str(PlanetClient.getInstance().api_key()))
        protparam = quote(f'&proc={proc}') if proc != "default" else ""
        rampparam = quote(f'&color={ramp}') if self.can_use_indices() else ""
        uri = f"type=xyz&url={tile_url}{protparam}{rampparam}"
        self.layer.dataProvider().setDataSourceUri(uri)
        self.layer.setCustomProperty(PLANET_CURRENT_MOSAIC, name)
        self.layer.setCustomProperty(PLANET_MOSAIC_PROC, proc)
        self.layer.setCustomProperty(PLANET_MOSAIC_RAMP, ramp)
        self.layer.triggerRepaint()

def login_changed(self):
    self.change_source()
    self.ensure_correct_size()

def ensure_correct_size(self):
    def findLayerItem(root=None):
        root = root or QgsProject.instance().layerTreeRoot()
        for child in root.children():
            if isinstance(child, QgsLayerTreeLayer):
                if self.layer.id() == child.layer().id():
                    return child

```

```

        elif isinstance(child, QgsLayerTreeGroup):
            return findLayerItem(child)
    item = findLayerItem()
    if item is not None:
        if not PlanetClient.getInstance().has_api_key():
            item.setExpanded(True)
        isExpanded = item.isExpanded()
        item.setExpanded(not isExpanded)
        item.setExpanded(isExpanded)

    def processes_for_datatype(self):
        if self.datatype == "uint16":
            return ["default", "rgb", "cir", "ndvi", "mtvi2", "ndwi",
                    "msavi2", "tgi", "vari"]
        elif self.datatype == "byte":
            return ["default", "tgi", "vari"]
        else:
            return ["default"]

WIDGET_PROVIDER_NAME = "planetmosaiclayerwidget"

class MosaicLayerWidgetProvider(QgsLayerTreeEmbeddedWidgetProvider):

    def __init__(self):
        QgsLayerTreeEmbeddedWidgetProvider.__init__(self)
        self.widgets = {}

    def id(self):
        return WIDGET_PROVIDER_NAME

    def name(self):
        return "Planet Mosaic Layer Widget"

    def createWidget(self, layer, widgetIndex):
        widget = MosaicLayerWidget(layer)
        self.widgets[layer.id()] = widget
        return widget

    def supportsLayer(self, layer):
        return PLANET_CURRENT_MOSAIC in layer.customPropertyKeys()

    def logoutLayerWidgets(self):
        PlanetClient.getInstance().log_out()
        for widget in self.widgets.values():
            widget.login_changed()

```

ДОДАТОК В.

Електронні плакати

*Міністерство освіти і науки України
Східноукраїнський національний
університет ім. В.Даля
Кафедра комп'ютерних наук та інженерії*

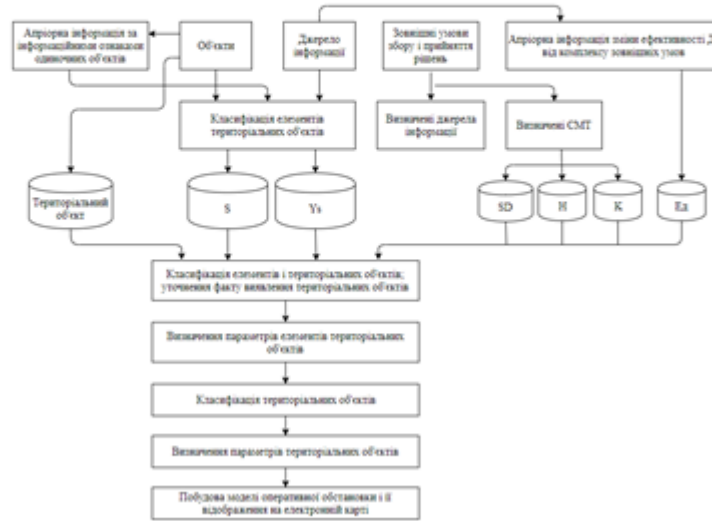
*Магістерська робота
«Моделі і методи системи
моніторингу територій»*

*ст.гр. КН-19дм Карташов М.В.
Керівник: Щербаков Є.В.*

Актуальність роботи

- **Об'єктом дослідження** магістерської роботи є системи моніторингу території. Предметом дослідження – моделі і методи розпізнавання геоінформаційних ситуацій в системах моніторингу.
- **Метою дослідження** є покращення науково-методичного апарату обробки картографічної інформації про об'єкти моніторингу для підвищення ефективності розпізнавання геоінформаційних ситуацій.
- **Наукова новизна роботи** полягає в тому, що в ній розроблено узагальнену модель геоінформаційних ситуацій на основі їх інформаційних ознак з урахуванням контексту зовнішніх умов та на її основі запропонований метод їх розпізнавання по нечіткій і неповній вихідній інформації.

Узагальнена модель процесу обробки інформації в СМТ



Метод розпізнавання геоінформаційних ситуацій в залежності від зовнішніх умов



Формування опису основних лінгвістичних змінних територіальних об'єктів у системах моніторингу територій

$$I_{\text{обп}} = (m_{i_1} a_{i_1} m_{i_2} a_{i_2} \dots m_{i_p} a_{i_p}), \quad (1)$$

де m_i – довжина поля однорідних символів виду a_i ;
 p – число полів у груповому об'єкті. Поля розділені не
більше ніж трьома символами a_o .

$$I_{\text{обп}} = (2G 2K 4G 3T 4G 4G), \quad (2)$$

$$m = (m_1 m_2 m_3 m_4 m_5 m_6) = (2 2 4 3 4 4),$$

$$s = (s_1 s_2 s_3 s_4 s_5 s_6) = (G K G T G G). \quad (3)$$

Вибір інструментів для вирішення поставлених задач



The logo for QGIS (Quantum GIS) features a stylized orange and green map icon to the left of the letters 'QGIS' in a bold, green, sans-serif font.

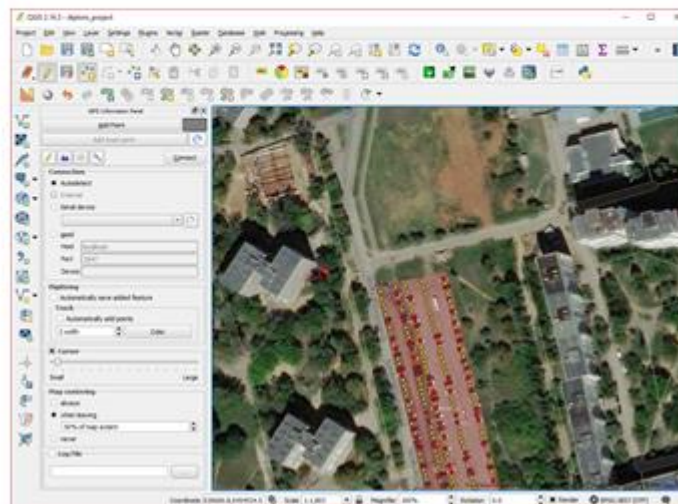


The Python logo, consisting of two interlocking snakes (one blue, one yellow), is centered within a teal rectangular box. To the left of the logo, the word 'python' is written in a light blue, lowercase, sans-serif font, enclosed in curly braces.

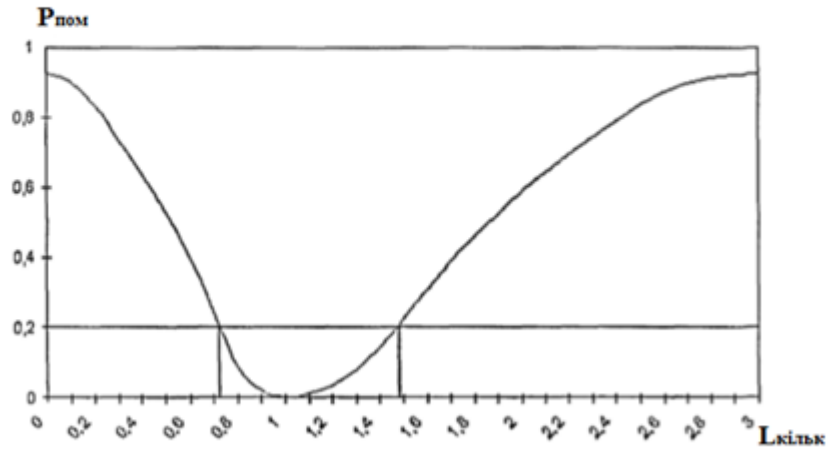
Структура системи моделювання нечіткого геоінформаційного розпізнавання об'єктів та ситуацій



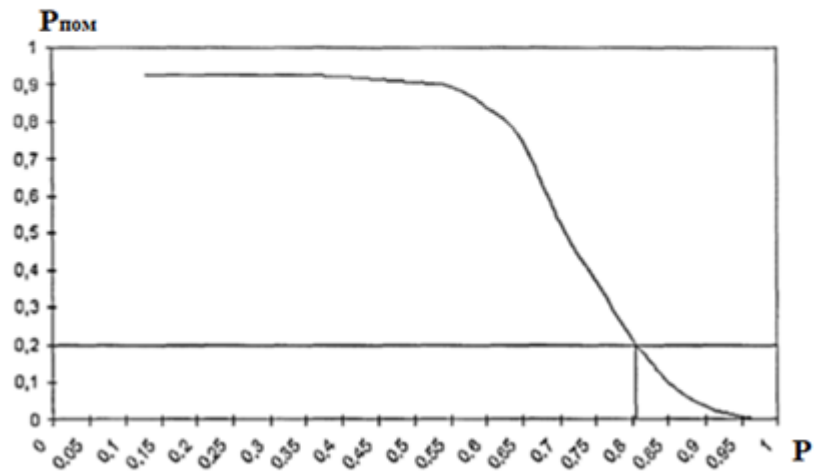
Реалізація пошуку вільних місць на паркові з використанням пакету програм QGIS



Графік стійкості методу до зміни довжини лінійного об'єкта відносно встановленої



Графік стійкості методу до зміни значень на головній діагоналі матриці змішування





ВИСНОВКИ

Розроблено узагальнену модель процесу обробки інформації в системі моніторингу територій в припущенні ідеального каналу зв'язку, що відрізняється урахуванням впливу контексту зовнішніх умов на результати багаторівневого прийняття рішень у СМТ. Це дозволяє розробити процедури, що компенсують його негативний вплив.

Розроблено узагальнену модель геоінформаційних ситуацій на основі їх інформаційних ознак, яка відрізняється способом організації простору ознак, сформованого у вигляді комбінації векторів, що описують структурні і нечіткі параметричні властивості територіальних об'єктів, а також правил їх багаторівневого узагальнення, що дозволяє описувати геоінформаційні ситуації довільної складності з урахуванням контексту зовнішніх умов.

Розроблено метод і алгоритми розпізнавання геоінформаційних ситуацій по нечіткій і неповній вихідній інформації на основі нечіткої штучної нейронної мережі, що відрізняються адаптацією функцій приналежності нечітких інформаційних ознак і їх числа відповідно до змін контексту зовнішніх умов, що дозволяє підвищити якість розпізнавання.