МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра математики факультет інформатики

Кваліфікаційна робота

освітній ступінь – бакалавр

на тему: **“Розпізнавання сонячних панелей на супутникових зображеннях“**

Виконав: студент 4-го року навчання

спеціальності 113 Прикладна математика

Колінько Павло Володимирович

Керівник Жежерун О. П.

кандидат фіз.-мат. наук, доцент

Рецензент Франчук О.В.

Кваліфікаційна робота захищена

з оцінкою _____

Секретар ЕК _____

«____» _____20____ р.

Київ – 2022

Міністерство освіти і науки України

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра математики

ЗАТВЕРДЖУЮ
Зав.кафедри математики,
проф., д.ф-м.н., Б. В. Олійник
_____
(підпис)
„____”_____2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на кваліфікаційну роботу

студенту 4-го курсу, факультету інформатики__
Колінько Павлу Володимировичу
Розробити Алгоритм розпізнавання сонячних панелей на супутникових знімках
Вихідні дані:

- датасет для навчання алгоритма;
- датасет територій для розпізнавання;

Зміст ТЧ до кваліфікаційної роботи:

Зміст
Анотація
Вступ
1 Огляд існуючих методів обробки зображень
2 Підготовка даних
3 Навчання алгоритмів
4 Практичне використання алгоритмів
Висновки
Список літератури
Додатки

Дата видачі „___” _____ 2022 р. Керівник _____
(підпис)
Завдання отримав_____
(підпис)

Графік підготовки кваліфікаційної роботи до захисту

Графік узгоджено «_____» _____ 2022 р.

№ з/п	Перелік робіт	Термін	Підпис	Дата	Примітка
1.	Отримання теми кваліфікаційної роботи	16.10.2021			
2.	Ознайомлення з існуючою інформацією за темою курсової роботи	23.10.2021			
3.	Створення плану роботи	12.11.2021			
4.	Робота з науковою літературою	13.11.2021			
5.	Аналіз предметної області дослідження, аналіз існуючих рішень та алгоритмів	02.01.2022			
6.	Аналіз математичних методів реалізації алгоритмів	02.02.2022			
7.	Підготовка даних до аналізу, використання обраних алгоритмів, оцінювання точності та ефективності	05.04.2022			
8.	Аналіз практичної частини, її корегування	14.04.2022			
9.	Початок написання текстової частини	21.04.2022			
10.	Подання проміжної версії текстової частини	11.05.2022			
11.	Завершення написання текстової частини роботи	22.06.2022			
12.	Створення презентації	23.06.2022			
13.	Захист кваліфікаційної роботи	04.07.2022			

ЗМІСТ

Антоція

Метою проведеного дослідження є використання математичних алгоритмів класифікації для отримання “маски” зображень сонячних панелей на певній території.

У роботі було описано теоретичні аспекти аналізу предметної облаті - супутникові зображення безпосередньо та алгоритми класифікації, які були використані під час роботи. Практична частина складається з програмного застосунку, мета якого прочитати зображення та сформувати матрицю спектрів. На основі отриманих даних, був проведений аналіз за допомогою гістограм та описової статистики, з подальшим вибором алгоритма, які буде навчатись.

Результатом роботи є зображення, на яких жовтим кольором виділено необхідні нам ділянки та темно-фіолетовим всі інші території. Таким чином, ми маємо змогу проаналізувати не тільки територіальні дані, а й вирішити задачу алгоритмічним підходом, покращити його та оцінити результат.

Вступ

Кожні десять років людство робить більше, в контексті технологічно розвитку, ніж за минулі п’ятдесят. Прогрес ні на мить не зупиняється - космічні перельоти, супутники, швидкість передачі даних, зброя, медицина, - це все лише невелика частина сфер людського життя, які досягли неймовірних висот за останні десятиріччя.

Окремої уваги наразі набуває розвиток комп’ютерної оброки мультимедійної інформації, такі як : розпізнавання людскої мови, семантичний аналіз текстів та обробка зображень.

Обробка зображень може мати дуже різні цілі. Фактично, кінцевою метою навчання моделі - є наближення до людського сприйняття зоровим апаратом образів зовнішнього середовища.

Окремо необхідно виділити обробку та розпізнавання супутникових зображень, основна мета якої, автоматизувати та покращити роботу по розпізнавання різних об’єктів на знімках, задля подальшої побудови моделі, карти, тощо. Наразі дуже широко використовується розпізнавання певних об’єктів на супутников знімках, задля подальшої роботи з ними - розпізнавання місцевості, об’єктів інфраструктури, специфічних будівель і т.п. Такі методи обробки можуть бути використані при створенні кадастрових реєстрів та вирішеня інших схожих задач.

Актуальність теми розглянемо у двох аспектах - алгоритмічному та практичному. З точки зору розробленого алгоритму, вирішення задачі грунтується на методах математичної класифікації, які було застосовані для задачі розпізнавання сонячних панелей, оцінено ефективність роботи алгоритмів та обрано той, що надає найкращі результати.

З практичної точки зору, розпізнавання саме сонячних панелей на супутникових знімках - це задача оцінки потенціалу вироблення сонячної

енергії в окремих містах та територіях. Оцінка такого потенціалу напряму впливає на інвестиційну політику провідних компанії по виробленню екологічно чистої енергетики, такої як, наприклад, американської компанії CleanPower, та багатьох інших інтернаціональних та Українських компаній.

Для оцінки енергетичного сонячного потенціалу оцінюються значення таких параметрів:

1. Місто / Країна.

2. Азимут нахилу елемента даху.

3. Нахил елемента даху в градусах.

4. Площа елемента даху (в $м^2$).

5. Висота даху (в м).

6. Ідентифікатор елемента даху.

7. Координати центроїда елемента даху.

Потавлена задача полягала у розпізнаванні зазначених об’єктів на ділянках території.

1. ЗОБРАЖЕННЯ ТА РОЗПІЗНАВАННЯ

1.1 Задача обробки зображень

Процес обробки супутникових зображень дуже складна та об’ємна тема. Загалом, процес обробки супутникових зображень - це процес зчитування, перетворення та обробки зображень з метою подальшого аналізу та знаходження певних правил та послідовностей, як, наприклад, виявлення певних об’єктів на цих супутникових знімках.

За останні роки, тема обробки зображень зазнала великих змін. До широкого використання комп’ютерного зору (computer vision), зазвичай використовували методи аналізу “вручну”. Вся обробка зображень будувалась на класичному математичному апараті - статистичному аналізу результатів експериментів, опису різними математичним структурами, такими як графи, матриці, логічні послідовності.

Наразі, змінилась і сама постанова задачі. Задачі розпізнавання геометричних образів перейшли у пошуки об’єктів певного типу на супутникових знімках. Як приклад, може бути наступна задача - знайти на знімку і виділити контури всіх будинків, в котрих нахил елементів дахів знаходиться в межах 25-35 градусів і якщо таких елементів в одному об’єкті не менше 60%.

Метод обробки даних “вручну” є вкрай неефективним. Людина не може настільки швидко обробляти надзвичайно великі об’єми даних, як це може зробити машина. Саме тому зараз стало дуже поширеним використання нейроних мереж, методів математичної класифікації та інших підходів машинного навчання для вирішення данної проблеми. Коли мова йде про супутникові зображення - це сотні тисяч кілометрів території та тисячі об’єктів, розташованих на ній.

Як було зазначено, галузь, що займається обробкою зображень називається комп’ютерний зір або ж *Computer vision.* Комп’ютерний зір фактично призваний імітувати людський зоровий аппарат. Людське око здатне сприймати різні спектри світла та складати отриману інформацію в зображення, що передається на зоровий нерв та обробляються мізком людини.

Попри видимі переваги, комп’ютерний зір має певні недоліки. Людина отримує навички розпізнавання різних об’єктів під час знайомства з оточуючим середовищем. Для того щоб машина була спроможнна робити те ж саме, їй необхідно надати великий масив даних та “навчити робити висновки”. Саме від величини вибірки буде залежати кінцевий результат. Проте, отримані в процесі роботи алгоритми машинного навчання є достатньо вузьконаправленими - алгоритм, що вміє відрізняти собаку від кішки ніяк не допоможе при необхідності визначення кількості сонячних панелей на ділянці території.

Загалом, CV(Computer Vision) - це сукупність алгоритмів, зазвичай на базі машинного навчання, що має на меті не тільки розпізнавання, але й обробку отриманих даних та формулювання висновків, що залежить від поставленої задачі.

Наразі, обробка зображень проходить у декілька етапів:

- Підбір даних, на основі яких буде створена модель, що зможе проаналізувати та зробити корисні висновки;
- Підготовка даних для подальшої обробки;
- Аналіз отримани даних;
- Створення набору для тренування моделі;
- Підбір алгоритма класифікації та/або нейронної мережі;
- Аналіз отриманих результатів;

1.2 Зондування землі та супутникові зображення Santinel-2

Дистанційне зондування Землі за допомогою обладнання встановленого на штучних супутниках Землі має вже більше ніж піввікову історію. Все почалося з аерофотозйомки, яку в 1909 році було виконано під час демонстративного польоту Вілберта Райта. Наступним кроком була зйомка, яку виконали 24 жовтня 1946 року камерою, встановленою на борту ракети V-2 з висоти 65 миль. Але справжня ера дистанційного зондування Землі з космосу почалася нещодавно - 23 липня 1972 р. з запуском на орбіту супутника Landsat-1.

В наш час навколо Землі кружать супутники багатьох місій і можливостей:

- Супутники угруповання GPS – виконують роль визначення і позиціонування об’єктів;
- Супутники Sentinel-1 і Sentinel-2 виконують дуже об’ємну роботу по зйомці всієї поверхні землі в оптичному діапазоні
- Супутники серії RadarSat – знімають поверхню Землі в радіодіапазоні, що дає можливість зйомки 24 години на добу, не зважаючи на ніч і хмарність;
- Супутники серії Maxar і Airbus знімають Землю як в моно так і в стереорежимах, що дозволяє будувати точні 3-х мірні моделі міст;

Задачі які можна вирішувати за допомогою космічної зйомки можна класифікувати по сферам застосування:

- створення і оновлення карт різного масштабу;
- створення кадастру територій;
- екологічний та природоохоронний моніторинг території;
- оцінка стану сільськогосподарських культур, прогнозування урожаю;

- контроль стану лісів, спостереження за вирубкою;
- геологічні дослідження;
- дослідження атмосфери;
- контроль судноплавства, тощо;

В останні роки методи дистанційного зондування Землі за допомогою супутників почали, з одного боку, наздоганяти такі, традиційно, більш точні системи отримування даних, як аерофотозйомка, а з, іншого боку, почалися процеси, об’єднання результатів зондування з даними отриманими іншими методами, наприклад, Lidar-зйомки.

Моделі штучного інтелекту можуть вирішувати задачі, які ще декілька років тому вважалися неможливими через шалені об’єми інформації, що потребують обробки, кількістю параметрів, їх «нечіткістю», складнощами паралельної обробки інформації тощо.

Супутникові знімки отримані від групи супутників Sentinel-1 та Sentinel-2 були обрані з декількох причин. [1] По-перше, вони є цілком безкоштовними та доступними, що дозволяє отримати велику вибірку даних для навчання і аналізу будь-якої ділянки землі. По-друге, зйомка проводиться безперервно і часто супутники роблять знімки однієї й тієї ж території під різними ракурсами, що дозволяє отримати більш точну та різноманітну вибірку даних. І наостанок, супутники Sentinel проводять зйомку у різних світлових спектрах, що дозволяє робити точний аналіз.

Рисунок 1.1 - Приклад супутникового знімка Sentinel-2

1.2.1 Просторова роздільність

Знімок, як будь який звичайний, так і супутниковий - це двовимірне зображення, на якому було зафіксоване власне та відбите світлове випромінення за допомогою спеціальної апаратури, в нашому випадку камерами, що розташовані на супутниках. Найважливішими характеристиками зображень є просторова, радіометрична, спектральна та часова роздільна здатність.

Просторова роздільна здатність — це величина пікселя на зображенні в просторових одиницях, яка характеризує розмір найменших об’єктів, помітних на зображенні. Тобто, чим вища просторова роздільність, тим якісніше та чіткіше буде знімок, та тим більше об’єктів можливо буде розрізнити на ньому.

Існують знімки різних масштабів, від якості яких напрями буде залежати отриманий результат.

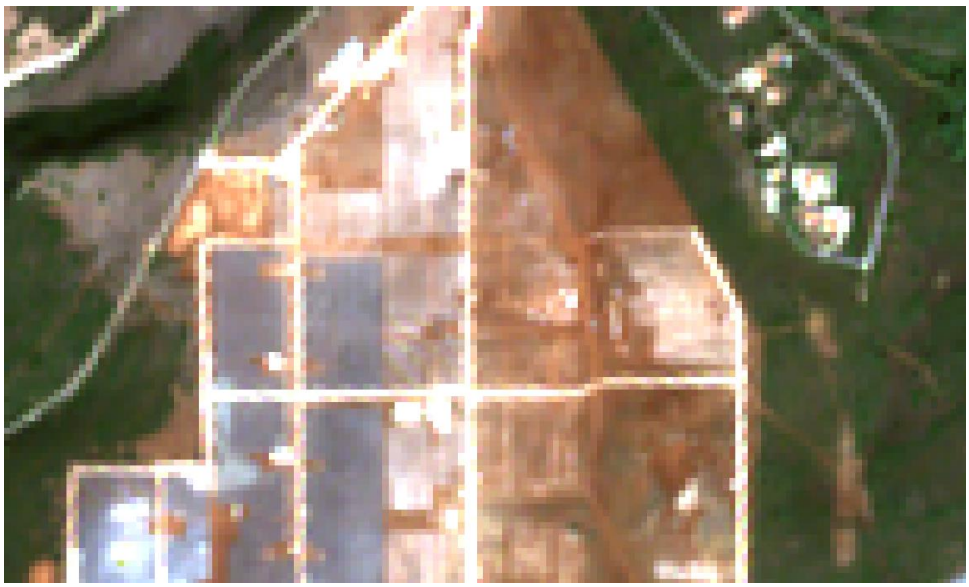

Рисунок 1.2 - Частина супутникового знімка Sentinel-2 в великому наближенні, де чітко можна роздивитись пікселі.

1.2.2 Зображення та їх формати

Є велика кількість форматів, у яких зберігається зображення. [2] Кожен з них має свої переваги та недоліки. Зазвичай, для зберігання супутникових знімків використовують растові зображення. Починаючи з загальновідомих, таких як JPEG(JPG), PNG так і меньш широковживаних, як наприклад той формат, що використовується для зберігання знімків супутника Sentinel-2 – TIFF(TIF).

Такі формати як JPEG(JPG), стискає зображення під час зберігання, що може вплинути на точність отриманих даних, а формат зберігання TIFF(TIF) не проводить операію стискання під час зберігання, що дозволяє

отримати найбільш точну картину, з урахуванням можливостей сьогоденої апаратури для проведення зйомок.

Хоча формат зберігання супутникових знімків дозволяє не втрачати точність данних, є і певні недоліки - вони дуже об’ємні, тому розмір на носіях буде чималим [2]. Як приклад, невелика ділянка території займає гігабайти місця на носіях, а цілой країни - терабайти.

1.2.3 Спектри зображень

Знімки Sentinel-2, як було зазначено вище, мають багато спектрів. Це дозволяє отримати максимально точні результати під час аналізу. Нижче буде наведена таблиця, де вони перераховані [1].

Sentinel-2 bands	Central wavelength (μm)	Resolution (m)
Band 1 – Coastal aerosol	0.443	60
Band 2 – Blue	0.490	10
Band 3 – Green	0.560	10
Band 4 – Red	0.665	10
Band 5 – Vegetation red edge	0.705	20
Band 6 – Vegetation red edge	0.740	20
Band 7 – Vegetation red edge	0.783	20
Band 8 – NIR	0.842	10
Band 8A – Vegetation red edge	0.865	20
Band 9 – Water vapour	0.945	60
Band 10 – SWIR – Cirrius	1.375	60
Band 11 – SWIR	1.610	20
Band 12 – SWIR	2.190	20

Рисунок 1.3 - Світлові спектри супутникових зображень Sentinel-2.

1.3 Методи обробки зображень

Обробка, аналіз та висновки з отриманих результатів шляхом роботи комп’ютерного зору можливо зробити багатьма способами - класифікація, математичні алгоритми, різні нейронні мережу, тощо. Варто звернути увагу на математичні алгоритми классифікації, яких існує чимала кількість. У даному розділі буде проведено опис теоретичної частини аналізу зображень, виділення сегментів, розбір алгоритмів класифікації.

1.3.1 Сегментація зображень

Сегментація зображень - це процес розділення зображення на меньші та більш значущі сегменти. Зазвичай, використовується об’єднання пікселів у группу, що також називається суперпікселем. Розподіл відбувається за допомогою об’єднання пікселів за візуальним параметром або ж іншою характеристикою, в певний об’єкт що має зміст [3].

Сегментація використовується в дуже різних галузях, першочергово задля виділення об’єктів та його меж (контурів). Це може бути корисним під час аналізу рентгенівських знімків для виявлення хвороб або ушкоджень. Загалом, це дозволяє розділити знімок за цікавлячими сутностями.

Цей процес буває різним. В загальному розумінні, сегмантація це більш широке поняття - як, наприклад, бінарна сегментація - це поділення сутності на певні бінарні класи. Повертаючись до вже наведеного прикладу сегментації зображень, для виявлення хвороб, бінарний тип сегментації може бути використаний у контексті виявлення наступних двох класів -

хворих та здорових людей, що мають певну ваду або травму та ті, що її позбавлені. У данному випадку, ми поділяємо зображення на об’єкти, де будуть присутні сонячні батареї, а де ні.

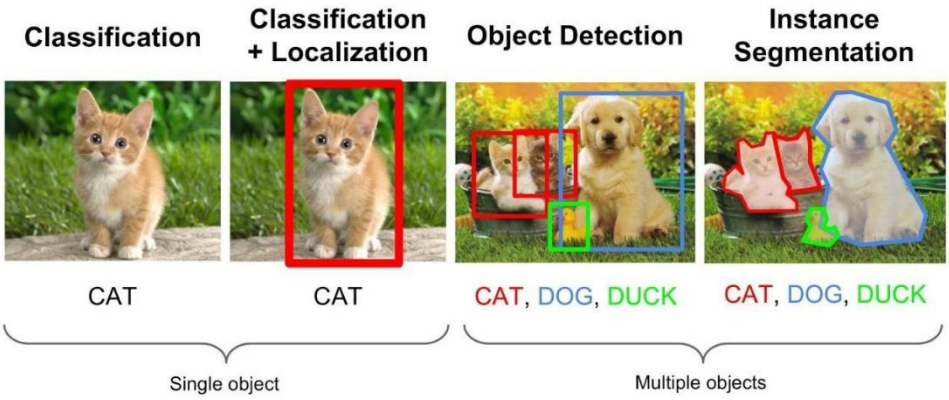

Рисунок 1.4 - Приклад сегменації за сутностями.

На даному малюнку ми визначаємо кожну групу пікселів, як окрему сутність, тобто собака, кіт, тощо. Семантичною сегментацією, у цьому прикладі було, б об’єднання всіх котів в окрему групу, а не тільки за сутностями [3].

Для сегментації зображень було розроблено декілька універсальних алгоритмів. Основні з них:

1) Визначення порогів;

Це найпростіший метод сегментування. Він базується на пороговомоу значенні, всі дані після якого відсікаються.

2) Методи, що засновані на кластеризації;

Цей метод заснований на ітерації по всьому датасету. Для початку необхідно обрати деяку N-кількість кластерів за допомогою деякого алгоритму оцінки, наприклад евристики, і шляхом ітерації приєднати кожен

піксель до цього кластера. Після приєднання знову порахувати центр, та повторювати до максимальної збіжності.

3) З використанням гістограм;

Дуже ефективний метод, прешочергово через економію часу. Для реалізації цього методу можливо лише один раз зробити ітерацію по массиву пікселів, та на основі описової статистики та візуального аналізу, зробити висновок про приналежність до того чи іншого кластера. Зазвичай, для порівняння може бути використаний колір або спектр.

4) Виділення країв;

Метод виділення країв(контурів) - це основа для використання інших методів сегментації.

5) Методи розростання областей;

Останній з наведених методів має на меті надходження певних початкових даних, та подальшого розширення області, шляхом “захоплення” пікселя в область початкових даних.

Наведені методи є загальними, та лежать в основі сегментації зображень, що відбувається під час роботи алгоритма або нейронної мережі.

1.3.2 Опис підходів до класифікації

Ці два поняття являються взаємодоповнюваними. Їх можна представити як сукупність правил, методів, алгоритмів вибору об’єктів (як метода розпізнавання) і сукупність правил опису взаємозв’язків вибраних об’єктів (як метода класифікації). Іншими словами, розпізнавання готує

матеріал для класифікації, хоча і класифікація є частиною процедури розпізнавання.

Класифікація – це сукупність правил створення систем класифікаційних угрупувань і їх взаємозв'язків. Серед найбільш поширених методів можна виділити ієрархічний, фасетний та дескрипторний. Вони розрізняються стратегією застосування класифікаційних ознак.

Ієрархічний метод класифікації визначається рівнями, які утворюють дерева об'єктів.

Касетний метод класифікації визначається як паралельний поділ множин об'єктів на незалежні підмножини.

Дескрипторний метод класифікації розділяє множини об'єктів згідно деякого словника ознак. Нас цікавлять більше методи і підходи, які застосовуються в розпізнаванні, але, по суті, ми хочемо в результаті розпізнавання провести класифікацію об'єктів.

Розглянемо задачу розпізнавання сонячних електростанцій, які і є предметом досліджень і опишемо це в теорії множин.

Хай Ω – множина пікселів на знімках, а $b_{i,j} \in \Omega$, i= 1,n , j=1,m , піксел, який знаходиться в i – рядку, j – колонці матриці. З кожним елементом $b_{i,j}$ зв'язано вектор V_k, k=1,e – вектор ознак, кожен з яких може приймати значення зі своєї множини A_p, p=1,e. В загальному вигляді A_p – це множини довільної природи: числа, поняття, рядки, тощо.

Якщо розглядати результат розпізнавання, як першого етапу класифікації, то задача розпізнавання сонячних електростанцій на супутникових знімках може бути розглянута як задача пост класифікації

для будь-якого метода. Нижче наведені підходи по класифікації відносно заданої задачі.

1) Ієрархічний підхід

Розбити множину Ω на сукупність підмножини Ω_t, t=1,n $\in \Omega$ таких, що підмножини Ω_t утворюють між собою дерево (відносно значень вектора ознак V_k, k=1,e).

Наприклад, якщо Ω_t – це розпізнані об’єкти, що є сонячними електростанціями, то ієрархія може бути задана ієрархією адміністративного ділення: область, район, територіальна громада. Тоді серед параметра вектора V повинна бути ознака: V_e – адміністративне ділення, що приймає значення із множини значень ознак (область, район, територіальна громада).

2) Фасетний підхід

Якщо результатом досліджень повинні бути три підмножини Ω_1, Ω_2, $\Omega_3 \in \Omega$, що відповідають поняттям розміру електростанції, то об’єкти, які будуть розпізнані як електростанції, будуть сегментовані на три класи за кількістю пікселей, що належать об’єкту.

3) Дескрипторний підхід.

Якщо результатом досліджень має бути розділення розпізнаних об’єктів за словниками дескрипторів, наприклад, орієнтування по сторонам світу, наявності додаткових сонячних елементів орієнтування тощо, то це можна трактувати як розділення множини Ω за словниками ознак.

Хоча дослідження, яким присвячена дипломна робота, стосуються практичній задачі розпізнавання, але коректний результат має бути оформлений як класифікаційний поділ об’єктів, які були сформовані алгоритмами розпізнавання.

1.3.3 Алгоритми класифікації

Як було зазначено, існує велика кількість алгоритмів класифікації, що засновані на математичному апараті та реалізовані програмно.

З точки зору розпізнавання та обробки супутникових знімків, класифікація - це процес дешифрування космічних знімків, з метою виявлення та розвізнавання шуканих об’єктів на даних знімках. Фактично, існують два типи класифікації - без навчання (некерована класифікація, кластеризація) та з навчання (керована класифікація) [4].

Можна виділити наступні етапи класифікації під час обробки знімків:

- визначення кількості класів, їхнього змісту
- створення навчальних вибірок, у данному випадку обранння еталоних даних, де гарантовано будуть присутні шукані об’єкти, та порожні ділянки
- вибір алгоритму для проведення класифікації з навчанням
- виконання класифікаційного алгоритма
- обробка всіх даних за допомогою алгоритма
- оцінка точності результатів

Дуже важливим етапом є вибір правильного алгоритма класифікації. І хоча для задачі вибори сонячних панелей на супутниковом знімку, здавалося б є чітке логічне розподілення на два класи - безпосередньо сонячні батареї та всі інші об’єкти, тільки більш точний подальший аналіз може допомогти підібрати правильний алгоритм для обробки [5].

Рисунок 1.5 - Схема вибору алгоритма

Ця схема, на жаль, не є дуже точною, бо частіше за все, як було зазначено, лише більш глибокий аналіз вхідних даних покриває потреби обрання алгоритма. Краще розподіляти ці методи за складністю реалізації, адже бувають випадки, коли одного алгоритма навчання не вистачає для отримання задовільного результата.

Наразі, варто розглянути шість основних алгоритмів бінарної классифікації. [6] Деякі з них будуть використані та порівняні ефективністю під час виконання практичної частини.

Бінарна класифікація поділяється на наступні алгоритми найбільш розповсюджені алгоритми:

- Логістична регресія

Цей метод працює на основі припущення незалежної лінійності і фактично є удосконаленої версією лінійної регресії. Логістична регресія використовує сигмоїдну функція, що також називається логістичною, для визначення приналежності до того чи іншого класу. Це крива, що може

приймате будь яке число та у висновку відображати його як нуль або одиницю, фактично присвоюючи вирогідність.

Якщо існує деяка випадкова величини *Y*, така що може приймати значення {0} або {1} залежно від деякого вектора значень $X=\{x_1, x_2, \dots x_n\}$ в такому випадку залежність можна проінтерпретувати ввівши змінну y^*:

$$y^* = \theta^T x = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n + \varepsilon \tag{1.1}$$

Де x_i – значення з вектора *X*;

θ – параметр правдоподібності, що можливо знайти шляхом максимізації функції правдоподібності на вибірці.

$$\hat{\theta} = \operatorname{argmax}_\theta L(\theta) = \operatorname{argmax}_\theta \prod_{i=1}^{m} \Pr\{Y = Y^{(i)} | x = x^{(i)}\} \tag{1.2}$$

А шукане значення Y набуває значення вирогідності за наступним правилом.

$$Y = \begin{cases} 0, & y^* \leqslant 0 \\ 1, & y^* > 0 \end{cases}$$

Маючи вже визначений певний набір даних з чітким розділенням на класи, сигмоїда визначає вирогідність приналежності кожного пікселя до того чи іншого класу.

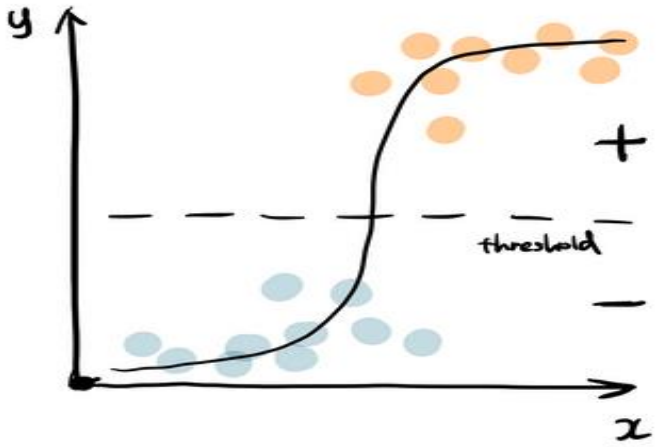

Рисунок 1.6 - Логістична регресія

- Дерево рішень

Дерево рішень - проста структура даних на основі графу(дерева). Фактично, у нас є набір правил, у якому є глибинне розгалуження на підкласи, що кожен раз зменшує область даних, на яких проводиться класифікація. Точність отриманих даних залежить від параметризуємих значень, таких як маштаб розгалужень, глибина.

У випадку класифікації спектру зображень необхідно вирахувати вирогідність приналежність пікселя з навчальної вибірки до одного з двох класів.

$$G(node) = \sum_{k=1}^{n} p_k(1 - p_k) \tag{1.3}$$

G(node) – шукана вирогідність для кожної вершини графа, що дорівнює вирогідності приналежності пікселя до класа. Він називається індекс різноманіття Джинні.

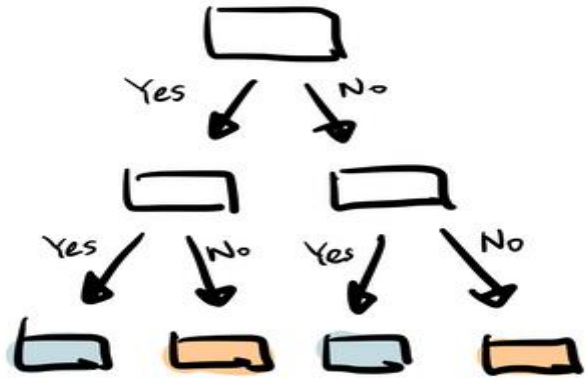

Рисунок 1.7 - Дерево рішень

- Випадковий ліс

Цей метод схожий на алгоритм дерева рішень та, фактично, складається з великої кількості останніх. Порівняно з ним він є більш узагальненим, проте гірше інтерпретуємим, адже має більше слоїв, доданих до моделі класифікації. Висновок робиться на основі умовних “голосів” дерева, на основі теорії, що більшість не може помилятись.

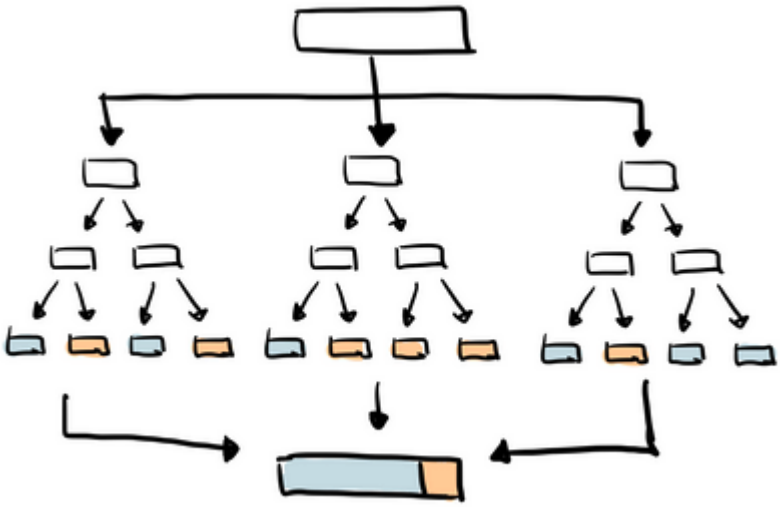

Рисунок 1.8 - Випадковий ліс

- Метод опорних векторів

Метод опорних векторів - ще один метод класифікації. Його принцип роботи полягає у наданні значенню певного класу за наступним алгоритмом - обрані дані з тренувального набору відносяться до певного класу, а метод опорних векторів будує модель нові зразки котрої відносить до тієї чи іншої категорії, роблячи це неймовірнісним бінарним лінійним класифікатором. Тоді пікселі з вектора $X=\{x_1, x_2 \ldots x_n\}$, що лежать у площині значень будуть задовольняти умову,

$$wx + b = 0 \tag{1.4}$$

а їх відношення до певного класу обраховується шляхом виконання умови (1.4) для кожного пікселя, в залежності від значення класу, тобто вектора $Y=\{y_1, y_2 \ldots y_n\} \mid y_i \in \{-1, 1\}$.

$$y_i(x_i w + b) - 1 \geq 0 \quad \forall i \,. \tag{1.5}$$

Отримана оцінка дозволяє визначити до якої "сторони" гіперплощини відноситься певний піксель. Таким чином, метод буде працювати ітераційно для кожної одиниці даних по всій вибірці.

Цей метод, як і лінійна регресія добре працює за умови що дані є лінійно розподіленими, що буває доволі рідко в практичних задачах.

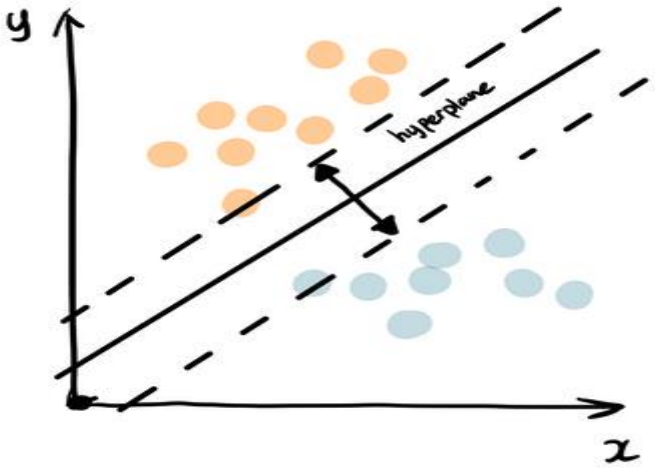

Рисунок 1.9 - Метод опорних векторів

- “K” найближчих сусідів

Метод найближчих сусідів - це метод класифікації, що відносить дані до певного класу, на основі впорахованої відстані від своїх сусідів на площині. Для виявлення приналежності точки до класу необхідно проводити наступні операції, в залежності від поставленої задачі:

- Інтерпретація для одного наближчого сусіда

  Він є найпростішим та найбільш інтуєтивно зрозумілим, що інтерпретується таким підрахунком:

$$C_n^{1nn}(x) = Y_{(1)} \tag{1.6}$$

Де Y – значення навчальної вибірки, що складається з$\{0\}$ та $\{1\}$.

- Модель зваженого класифікатора N найближчих сусідів

  Ідея полягає у тому, що *k* найближчим сусідам призначається вага *1/k*, а всім іншим вага *0*. Тобто найближчим сусідам присвоюється значення з вектора w_i, де $\sum_{i=1}^{n} w_{in} = 1$.

  Оптимальна тема зважження буде мати наступний вигляд:

$$w_{ni}^{*} = \frac{1}{k^{*}}\left[1 + \frac{d}{2} - \frac{d}{2k^{*2/d}}\{i^{1+2/d} - (i-1)^{1+2/d}\}\right] \quad (1.7)$$

  За умови що $w_{in}^{*} = 0$, для $i = k^{*} + 1, \ldots n$.

- K-NN викиди

  Необхідно також враховувати так звані викиди, тобто аномалії даних. K-найближчих сусідів також можна розглядати як підрахунок щільності розподілу даних, і чим нижча локальна щильність, тим більша вирогідність, що дані є аномалією.

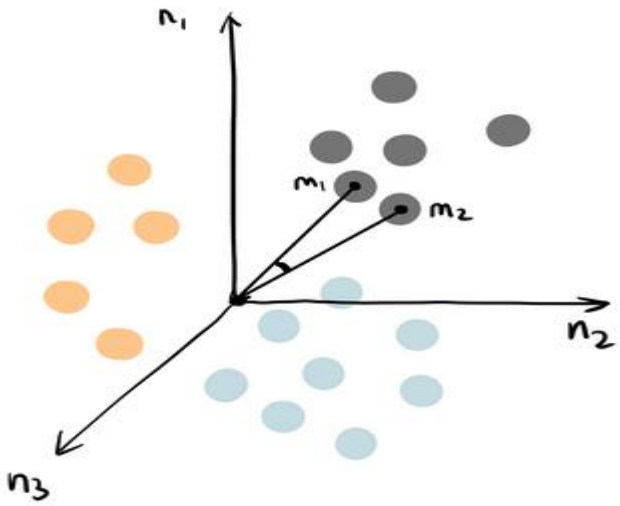

Рисунок 1.10 - “K” найближчих сусідів

- “Наївний” Баєс

Метод грунтується на теоремі Баєса. Теорема Баєса це одна з основних теорем у теорії йомовірностей. Вона полягає у вирахуванні того, що одна подія відбулась, на основі того що відбулись деякі статистично взаємопов’язані з ним події. Формула Баєса наступна,

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)} \tag{1.8}$$

P(A) - Апріорна вирогідність настання події *A*;

P(B | A) - Вирогідність настання події *A*, за умови настання події *B*;

P(A | B) - Вирогідність настання події *B*, за умови настання події *A*;

P(B) - Повна вирогідність настання події *B*;

Суть методу полягає у вирахуванні вирогідності базуючись на апріорних знаннях. Метод називається наївним, бо під час рахування “наївно” робиться припущення, що події незалежні.

Перевага такого метода полягає у невибагливості до об’єму тренувального датасету, також надання невеликої вирогідності невідомим даним, тобто не залишається некласифікованих даних. Це досягається методом адитивного згладжування(згладжування Лапласса).

Під час опису процесу навчання моделей, цей алгоритм буде формалізовано під умови поставленої задачі, адже сам він надає найбільш точний результат.

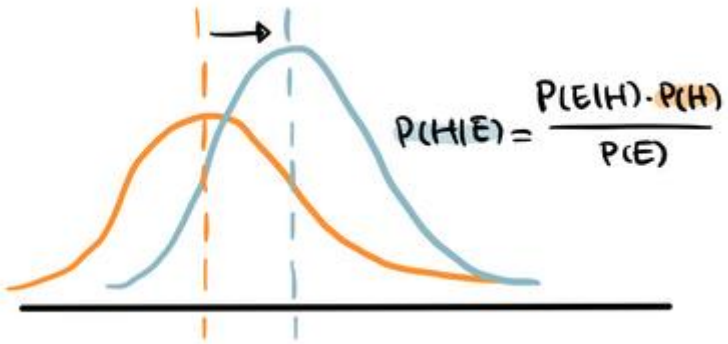

Рисунок 1.11 - “Наївний” Баєс

Надалі розглянемо більш складні алгоритми класифікації. Варто зазначити, що гарною практикою є поєднання декількох алгоритмів, з метою отримання максимально точного результата, бо одного може “не вистачити”. Розглянемо 4 алгоритма, що наведені у схемі 1.5.

1) Метод паралелепіпеда

Варто почати з метода параллелепіпеда, так як він є найпростішим в реалізації. Його використання є доречним у відмінній ситуації від використання бінарних алгоритмів классифікації - коли у задачі необхідно розпізнати більше ніж два класи, за умови, що дані цих класів не перетинаються. Тобто, дані, що належать до одного шуканого класу, не лежать у площині даних іншого.

Цей метод є доволі простим для інтуїтивного розуміння, та його реалізацію у контексті заданої задачі легко проінтерпретувати графічно.

Для роботи по розпізнаванню класів на супутникових знімках, цьому методу необхідно знати два критерії - середнє значення яскравості вибірки та похибку, стандартне відхилення від значення по всьому класу [7].

Аналіз проходить у три етапи:

- Вирахування центра даних кожного класу
- Знаходження крайної точки коного класу
- Опис точок класу в паралелепіпед по векторам, що проведені до крайніх точок та подальше відношення до цього класу всіх значень що опинились в цьому паралелепіпеді

Цей метод праює ітераційно та повторює ці три етапи кожен раз, допоки не закінчаться точки [7].

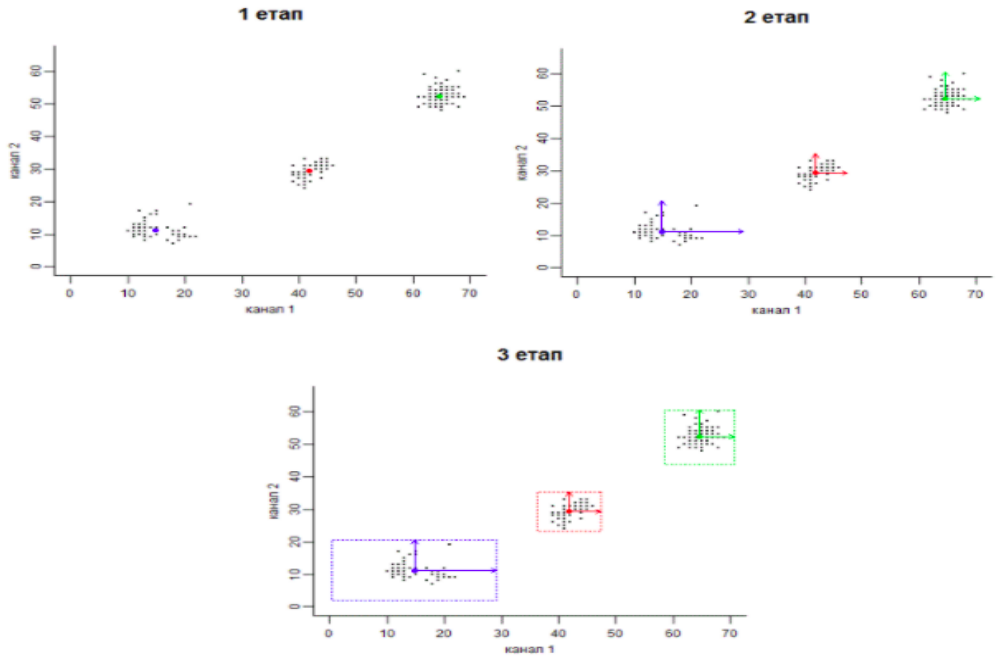

Рисунок 1.12 - Метод паралелепіпеда

2) Метод мінімальної відстані

Наступний метод слушно використовувати, якщо дані, що належать певним классам, перетинаються. [8] Як і попередній метод, він має три основні етапи та зручно візуалізується за допомогою графіків.

Три єтапи роботи данного методу:

- Вирахування центру кожного класу
- Другий етап вираховує відстань кожної точки даних, у даному випадку пікселя, до центра ввього розподілу даних
- На останньому етапі, йде порівняння кожної відстані для кожної точки і до якого вона найменша, до того класу і буде зараховано дані

Для вирахування відстані використовується евклідова відстань та метод повторюється ітераційно [8]. В загальному вигляді Евклідова відстань визначається наступної формулою,

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}. \quad (1.9)$$

проте в задач аналізу спектра вона має наступний вигляд:

$$F(R) = \left[\sum_{l=1}^{L} A_l \cdot (f_{l(ij)} - f_{l(km)})^2 \right]^{\frac{1}{2}}, \quad (1.10)$$

Де A_l – вагові коефіцієнти, що враховують залежність яскравості в одному спектрі;

$f_{l(ij)}$, $f_{l(km)}$ – значення яскравості пікселей

Ця формула справедлива як і для метода паралелепіпеда, так і для метода максимальної відстані.

У більш складних варіантів для аналізу відстані можуть бути використані різні скалярні характеристики текстурної матриці.

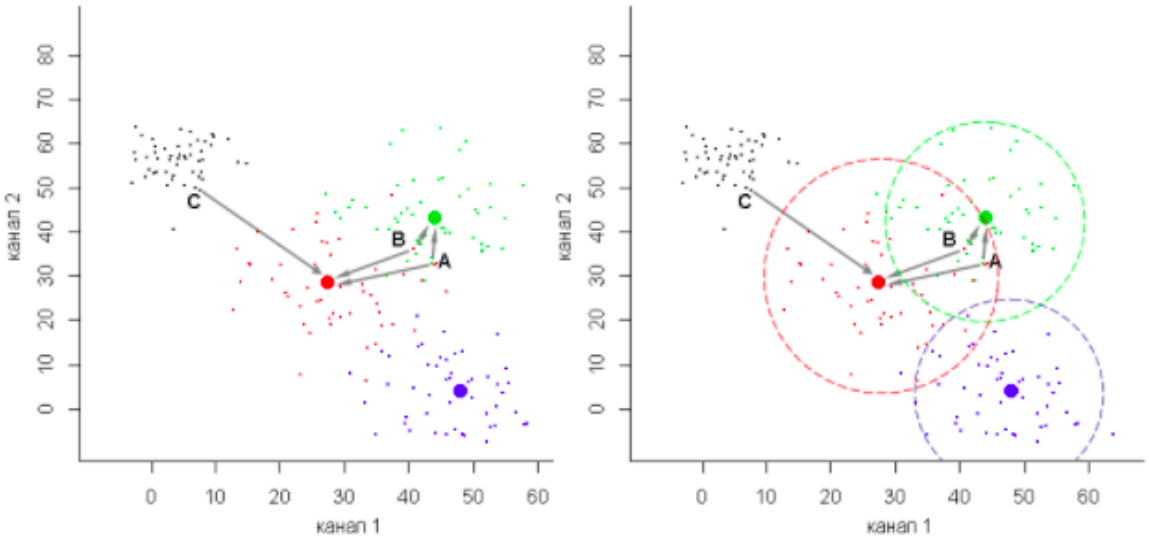

Рисунок 1.13- Метод найменшої відстані

3) Метод відстані Махалонобіса

Метод використовується за умови складних форм “хмар” значень, тобто їх дуж складно чітко вписати в одну конкретну геометричну фігуру.

По підходу до классифікації, метод дуже схожий на метод описаний у пункті 2, але замість Евклідової відстані між точками, цей метод використовує відстань Махалонобіса між векторами.

$$d(X,Y;S) = \sqrt{(X-Y)^T S^{-1}(X-Y)} \tag{1.11}$$

Y, X - вектори;

S - коваріаційна матриця, тобто квадратна матриця, яка складена з попарних коваріацій та дисперсій нашої випадкової величини;

Коваріаційна матриця у даному випадку є матрицею ознак, за якою проводиться характеристика. Таким чином, у загальному випадку класифікації за яскравостями маємо наступну формулу підрахунку відстані Махалонобіса.

$$D = (X - M_m)^T \times (COV_m^{-1}) \times (X - M_m) \quad (1.12)$$

D – відстань Махалонобіса;

m – визначений клас;

X – вектор виміру класифікованого піксела;

M_m – значення сигнатури класу m;

COV_m – коваріаційна матриця пікселів у сигнатурах класу m;

4) Спосіб максимальної правдоподібності

В основі методу лежить оцінювання невідомого параметра класифікації, шляхом максимазіції функції правдоподібності, для подальшого відношення даних до того чи іншого класу [9]. Вона рахується як розподіл величини, за параметром θ.

$$\mathcal{L}(\theta \mid x) = p_\theta(x) = P_\theta(X = x) \quad (1.13)$$

Як і в логістичний регресії, параметр необхідно максимізувати аналогічним способом, як у формулі (1.2). В загальному вигляді, для задачі класифікації за спектрами, формула буде мати наступний вигляд:

$$D = \ln(a_m) - [0.5 \ln(|COV_m|)] - [0.5(X - M_m)^T (COV_m{}^{-1})(X - M_m)] \quad (1.12)$$

D – вагова відстань (вірогідність);

a_m – відсоток вірогідності належності класифікованого пікселя до класу m (дорівнює 1,0 або вводиться на основі апріорних даних);

$|COV_m|$ – детермінант матриці COV_m

Метод припускає, що всі дані про вибірку лежать саме у цій функції. Отримані результати методу є доволі точними, адже він не залишає некласифіковані дані та враховує дисперсію.

2. АНАЛІЗ ТА НАВЧАННЯ АЛГОРИТМІВ

Розробка программного забезпечення, тобто алгоритм розпізнавання об’єктів на супутникових знімках проводилось у наступні декілька кроків:

- Підготовка даних з учителем
- Зчитування, перетворення та аналіз даних для навчання
- Навчання алгоритма
- Аналіз отриманих результатів та ефективності обраного методу
- Використання моделі на цікавлячій ділянці знімка (всьому об’єму даних)

2.1 Опис використаних інструментів

Починати розробку програмного продукту необхідно з правильного підбори інструментів. У наш час існує безліч готових інструментів для реалізації необхідного програмного забечпечення. Але, кожен з них має як свої переваги, так і свої недоліки.

Першочергово, необхідно зазначити програмуQGIS, яка раніше називалась Quantum GIS. Створена у 2002 році американским геологом Гаррі Шерманом для проведення робот по геодезії. Надалі до розробки кросс-платоформенного продукта приєднались інші розробники. Цей програмний продукт, що має як і десктопну так і онлайн версії, підходить під операційну систему Windows(останіх поколінь), під більшість Unix систем, такі як Linux та Mac OS. QGIS - це програма створена для роботи, аналізу та перетворення геопросторової інформації, такими як супутникові

знімки або ж бази даних, що наповнені геопросторовими даними. Має широкий набор інструментів для роботи з зазначеним типом даних. Цей інструментарій надає наступні можливості при роботі:

- Перегляд зображень
- Компоновка зображень, карт
- Створення, редагування та експорт даних
- Аналіз даних
- Публікація створених карт в мережі Інтернет
- Розширення програмного продукту за допомогою власноруч створених модулів

Під час роботи та, пер шочергово під час підготовки даних до подальшого аналіза, QGIS був використаний як засіб для візуального аналізу супутникових зображень, виділення контурів об’єктів та експорту отриманих шматків зображень для їх інтегрування в розроблену програму.

Не меньш важливим для якісної і зручної роботи є правильне обрання мови програмування. Як вже було зазначено, існує безліч інструментів та мов програмування на яких можливо реалізувати поставлену задачі. Для роботи було обрано мову програмування Python. Цей вибір був зроблений через кількість готового набору бібліотек, на відміну від переважної більшості інших мов програмування, наприклад Java, яка зазвичай використовується для вирішення іншого роду задач. І хоча інструментарій, що вже розроблений у мові програмування Python, для рішення задачі по розпізнавання об’єктів на зображеннях є дуже широким, є і свої недоліки, такі як швидкість роботи вказаної мови, що суттєво вплине на час виконання коду, та як наслідок, на швидкість навчання моделі.

Як середовище розробки було обрано програмне забезпечення міжнародної компанії Google – Google Colabortory, більш відомий як

скорочення Google Colab. Це дуже потужний інструмент для розробки програм, що підтрмує мову програмування Python, та має вже встановлені всі загальновідомі та необхідні модулі для рішення багатьох задач. Вона дає доступ розробнику до великої кількості оперативної та дискової пам’яті, що є дуже корисним, особливо у випадку якщо розробник не має достатньо потужного комп’ютера або ноутбука.

Формат зберігання супутникових зображень - TIFF(TIF), як стандатний формат для супутникових знімків, що дозволяє уникнути втрати даних під час стиснення та імпортування зображень.

Наостанок, варто описати бібліотеки, які було використано під час розробки програмного забезпечення. Було використано такі модулі:

- GDAL

Це безкоштовна бібліотека, що створена засобами мови програмування Python. Її основний функціонал зазвичай використовується для роботи з растовими зображеннями. Цей програмний модуль також підтримується і іншими мовами програмування. Він був обраний як дуже поширене рішення при роботі з растовим типом зображень, також він дозволяє зчитувати зображення по спектрам, тобто за рівнями його яскравості, що є дуже корисним для задачі розпізнавання сонячних ппанелей.

- Numpy

Стандартний модуль для роботи з різними математичними операціями, має розширений функціонал для роботи з масивами та багато інших допоміжних функції. Написаний на мові програмування C.

- Pandas

Це високорівнева бібліотека, що має а меті роботу з даними, створена на основі модуля Numpy, і так як Numpy написаний на C, дає високі показники у швидкості роботи.

- Mathplot

Популярна бібліотека для роботи по створенню і модифікації графіків.

- SKlearn

Дуже потужна бібліотека, що найчастіше використовується під час роботи у сферах Data Science(аналіз даних) и Machine Learning(машинне навчання). Надає можливості для попередньої обробки даних, їх стисненню, обрані моделі навчання, регресії, класифікації та кластеризації.

2.2 Підготовка та аналіз даних

Як було зазначено у розділі 2.1, для підготовких даних було використане програмне забезпечення під назною QGIS. Першочергово, необхідно виділити ті дані на яких ми будмо навчати обрані моделі з подальшим аналізом ефективності, використанні їх на всій вибирці дани, тобто шматку зображення, де будуть як і цікавлячі нас об’єкти сонячних панелей, так і порожніх територій.

2.2.1 Підготовка даних

Після обрання супутникового знімка з відкритих джерел, що підходить під необхідні критерії задачі, а саме наявність на ньому сонячних

батарей, необхідно “нарізати” зображення на шматки, на яких ми будемо навчати алгоритм [10].

Цей крок є необхідним, задля спрощення навчання. Готові еталонні дані дозволяють точно знати кількість нулів та одиниць (пікселів, що відносяться до класів, що розпізнаються, без розробки додаткового програмного продукта для їїх виокремлення, що сильно впливає на час розробки програми та на її безпосередній об’єм та інформаційну ємність, адже людина здатна швидше розпізнавати об’єкти в довколишньому середовиші або ж зображеннях.

Після візуального аналізу знімка у програмному забезпеченні QGIS було прийнято рішення виділити вручну на ньому так званні “еталонні нулі” та “еталонні одиниці”.

Рисунок 2.1 - Контури сонячних панелей та порожніх територій

Об’єкти “еталонних одиниць” – сегменти зображень сонячних панелей, фактично кінцева мета пошуку, а “еталонні нулі” – порожні території.

Інструменти цієї програми надають змогу експортувати лише ті частини зображень, що було обведені контуром, фактично, розділити великий знімок на багато маленьких, які є вхідними даними [10].

Як наслідок, ми отримали 32 зображення сонячних панелей, та 4 великі зображення територій, де вони гарантовано не присутні. Ці знімки було переміщено на Google Disk, як хмарне сховище даних. Це дуже зручно, так як Google Colab та Google Disk є програмним продуктом єдиної компанії Google, що спрощує імпортування даних з одного застосунка до іншого.

2.2.2 Побудова гістограм спектрів та їх аналіз

Першим кроком є обов'язковий імпорт всіх бібліотек зазначених у розділі 2.1.

API Google Collab дозволяє напряму підключитись до Google Disk, після дозволу заходити у свій аккаунт. Надалі, перейдемо до створення необхідних констант:

1) *sp_path* - шлях до місцерозташування даних на гугл диску знімків сонячних панелей
2) *nulls_path* - шлях до місцерозташування даних на гугл диску знімків територій без сонячних панелей
3) *sp_len* - кількість знімків сонячних панелей
4) *nulls_len* - кількість знімків порожніх територій
5) *band_names* - массив, що має назвих всіх спектральних каналів, що є необхідним для підпису гістограм

Також, створено декілька допоміжних функції, що будуть використані при подальшій роботі:

1) *flat* - Функція перетворення двомірного массиву в одномірний
2) *filter_arr* - Функція фільтрації массиву за певним обмежувачем, задля відкадання безмістовних даних
3) *choise* - Функція рандомного вибору даних з массива, необхідна для вирішення проблеми неоднорідності вибірок нулів та одиниць

Після створення допоміжного функціоналу, необхідно прочитати зображення та перетворити його у зрозумілу машині структуру.

Sentinel-2 використовує растові зображення, які складаються з пікселей. Піксель - це найменша структурна одиниця растового зображення, що є неподільною. Кожен піксель, під час відбиття на фоточутливій матриці апарату за допомогою якого було зроблено знімок, набуває певного кольору через змішання трех основних RGB (Red, Green, Blue) кольорів.

Нарізані знімки, що ми отримали на етапі підготовки даних з учителем ми маємо прочитати та трансформувати у двомірну матрицю розміру *NxM*. Кожен піксель є відображенням змішання кольорів що потрапили на світочутливу матрицю знімального апарату. Тому кожному пікселю буде поставлене у відповідність число, яке умовно опише його яскравість [11].

Використовуючи готові інструменти бібліотеки GDAL, прочитаємо знімки, та запишемо результат у матрицю. Наразі, зчитає тільки переший канал спектра, наприклад Coastol Aerosol, адже це нам буде необхідно задля проведення аналізу першого етапу.

За допомогою створеної функції *read_tiff_and_transform* прочитаємо перший спектр каналу. Дані необхідно зчитати для кожного знімка та покласти їх у одновимірний масив. Цей крок є необхідним для побудови гістограм і їх аналізу. На виході з методу ми отримаємо два масива значень

різних розмірностей - масив “еталоних нулів”, та масив з фотознімками сонячних батарей.

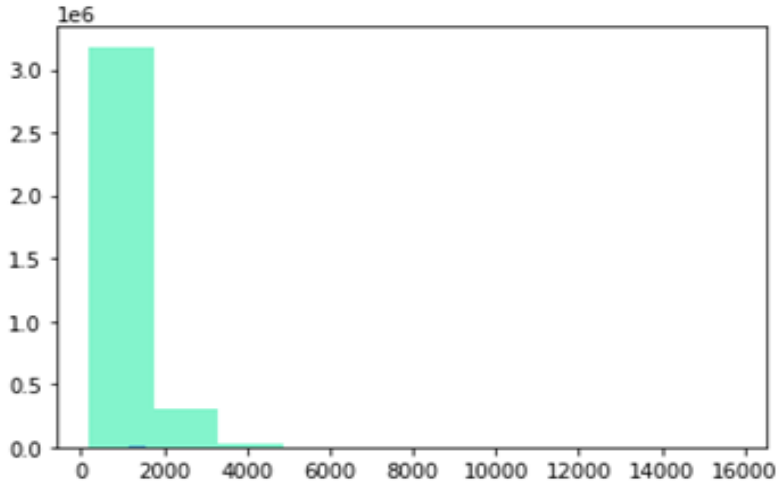

Рисунок 2.2 - Гістограма для аналізу першого спектра фотознімків

Як ми бачимо на рисунку 2.2 - гістограми настільки відрізняються, що на перший погляд їх навіть невидно *(заув. - дивитись внизу осі ординат).* Це пов’язано з дуже розповсюдженою проблемою в цій галуззі - неоднорідність вибірки даних [11].

Звичайно, можна “вручну” нарізати зображення умовно однакового розміру, але такий підхід буде мати багато недоліків. По-перше, при написанні програмного продукту дані все одно необхідно буде урівнювати в точні значення довжини масивів, адже це необхідно для правильної роботи алгоритма. По-друге, в контексті даної задачі, сонячні панелі мають відносно чітке спектральне відображення та є по суті единим об’єктом, що нас цікавить, на відміну від території, де сонячні панелі не наявні. На таких територіях присутня величезна кількість об’єктів, що нам необхідно відсіяти на результуючій “масці” - дерева, кущі, дороги, будівлі, різні водойми, тощо. Саме тому, більш коректним підходом є вибір великих

шматків порожніх території, де наявні найрізноманітніші об’єкти ландшафту та інфраструктури.

Саме такий підхід був обраний для рішення посталеної задачі. Задля зведення масиву до однокової довжини, була створена функція *choise*, що за допомогою функціоналу бібліотеки Numpy випадково обирає пікселі з масиву “еталоних нулів”, задля того, що у цей масив даних потрапило якумога більше пікселів, що відображають різні об’єкти.

Після прирівнювання масивів даних, необхідно побудувати гістограми та провести описову статистику з метою аналізу та обрання методів класифікації.

Першочергово, необідно прочитати всі світлові спектри, що представлені на супутникових знімках Sentinel-2, та візуалізувати отримані дані за допомогою бібліотеки Mathplotlib.

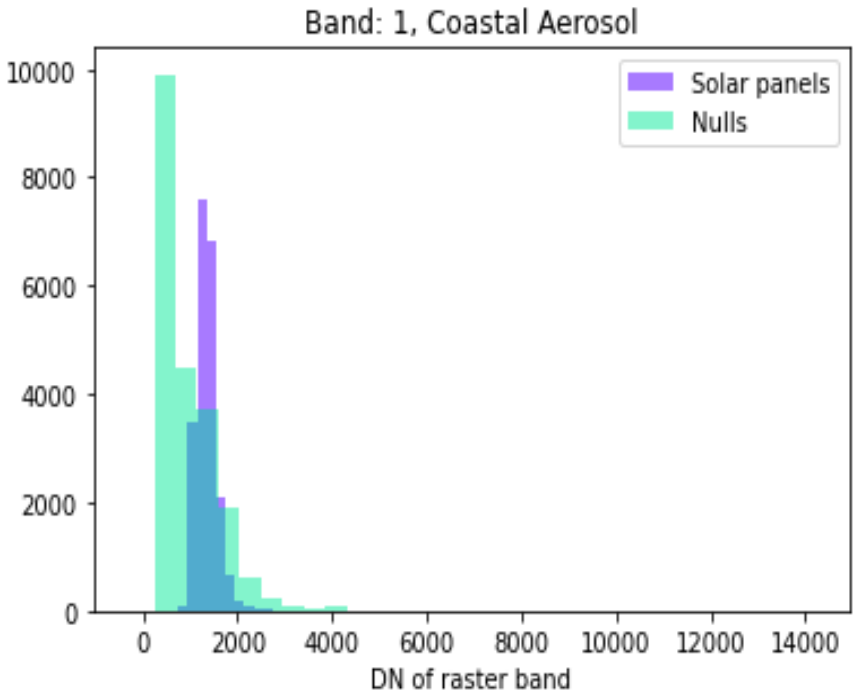

Рисунок 2.3 - Гістограма для аналізу першого світлового спектра

Solar pannels description

	0
count	21027.000000
mean	1344.566985
std	218.459188
min	746.000000
25%	1196.000000
50%	1329.000000
75%	1450.000000
max	2744.000000

Nulls description

	0
count	21027.000000
mean	887.074143
std	653.903783
min	234.000000
25%	270.000000
50%	742.000000
75%	1305.500000
max	5862.000000

Рисунок 2.4, 2.5 - Описова статистика даних спектра

З отриманих візуальних та статистичних даних, та проаналізувавши всі отримані гістограми можемо зробити наступні висновки - хоча дані і перетинаються, навіть візуально їх можливо чітко розділити на два класи [11]. Також, середні значення двох вибірок відрізняються у кожному спектральному каналі, тому беззмістовно відкидати будь-який з них. У випадку, якщо б дані дуже сильно перетинались на одному з каналів, їх варто було б відкинути, адже для подальшого навчання моделі вони не мають жодного змісту.

2.3 Використання алгоритмів класифікації

Після проведення аналізу отриманих даних, було прийнято рішення використати моделі бінарної класифікації, а саме:

- Логістична регресія;
- Дерево рішень;
- Випадковий ліс
- Гауссівський наївний Баєс

Було обрано саме ці моделі, з урахування специфіки даних та поставленої задачі, адже нам необхідно розділити дані саме на два класи. Більше однієї моделі було обрано з декількох причин,таких як порівняння результатів роботи методів, оцінка статичстичної точності, та головної - найкраща перевірка точності, це використання алгоритма на практиці.

2.3.1 Розділення вибірки даних

Обраний підхід до розподілення вибірки даних дуже полегшує навчання обраних моделей. Адже, через поділ об’єктів на шукані та ті, що нас не цікавлять, ми можело точно поставити у відповідність масиву пікселів сонячних панелей значення {1}, а масиву значень порожніх територій - {0}.

Для того, щоб правильно передати в модель дані, необхідно зробити наступні декілька кроків:

- Об’єднати масиви панелей та порожніх територій у один за кожним спектром
- У відповідність до створеної матриці створити вектор що буде мати кількість значень {1}, відповідну до довжини масиву сонячних панелей
- У відповідність масиву всього, що є порожніми триторіями - значення {0}

Для цього було створено функцію *concat_dicts,* що приймає на вхід словники *(заув. - назва структури даних у мові програмування Python)* з даними. На виході ми отримаємо дві структури:

- Матрицю X розмірності *mxn, m* - це кількість даних у кожному спектральному каналі, *n* - безпосередньо кількість спектральних каналів
- Вектор Y довжини *j*, де *j = m * m*

Наступним кроком є розділення вибірки на так звану тренувальну та навчальну, після чого у масив моделей передаєм отримані дані. Було прийнято рішення розділення у відсотковому співвідношенні *20/80*, адже це доволі стандартний підхід.

2.3.2 Тренування моделей

Тренування моделей проходить у дуже простий спосіб. Необхідно передати в модель розділену вибірку даних. Тренування проводиться для кожної з обраних моделей послідовно. Після того, як агоритм робить перші висновки з переданого набору даних, алгоритм повертає передбачення моделі у вигляді масиву зі значеннями {0}, {1}. Фактично, під час роботи, алгоритм визначає вирогідність приналежності даних на вході до певного класу. Кожен з обраних алгоритмів робить це у свій спосіб.

Дерево рішень або рандомний ліс будують структуру на основі графа. У данному випадку випадковий ліс - це велика структура з дерев класифікації, що працює “методом голосування” за теорією, що більшість не може бути не права. Точність данної моделі залежить від кількості листів, яку необхідно правильно підбирати [12].

Логістична регресіє використовує функцію детермінатор (сигмоїду) для відокремлення класів, а метод наївного баєсу - теорему баєса як основу, та метод апостеріорного максимума для віднесення даних до певного класу.

Варто більш глибоко роздивитись та формалізувати роботу методу наївного Баєса для поставленої задачі, адже саме він надає найбільш точний результат для вирішення поставленої задачі.

Як було зазначено у розділі 1.3.3 метод наївного Баєса грунтується на теоремі Баєса з “наївним” припущенням, що події незалежні [13]. Конкретизуємо формулу для випадка даної задачі:

$$P(class|sp) = \frac{P(sp|class)P(class)}{P(sp)} \tag{2.1}$$

P(class|sp) - вирогідність, яку нам необхідно розрахувати, що означає вирогідність того, що сонячна панель *(sp)* належить деякому классу *(class)*

P(sp|class) - вирогідність зустріти сонячну панель поміж класів

P(sp) - безумовна вирогідність зустріти шукану *sp* поміж усіх об’єктів

P(class) -безумовна вирогідність зустріти об’єкт, що належить до класу, поміж всіх об’єктів. Його ми можемо оцінити як відношення всіх елементів класу до загальної вибірки даних.

Ми отримали вирогідність яку нам необхідно порахувати, але оскільки мета метода полягає не у тому, щоб порахувати його вирогідність, а знайти приналежність до певного класу будемо використовувати оцінку апостеріорного максимума. Іншими словами, нам необхідо розрахувати вирогідність всіх класів та методом максимізації обрати найбільшу. Цей метод також тісно пов’язаний з методом максимальної правдоподібності.

$$c_{map} = \arg\max \frac{(P(sp|class)P(class)}{P(sp)} \tag{2.2}$$

Безумовна вирогідність сонячної панелі, що стоїть у знаменнику, є константою, і тому не можу вплинути на ранжування класів цього алгоритма, тому для спрощення підрахунків, надалі їм можна нехтувати.

У задачах комп'ютерного зору, всі об'єкти складаються з пікселів, тому *P(sp|class)* розкладається у перемноження вирогідності кожного пікселя належати класу.

$$P\left(sp|class\right) = P\left(pix_1|class\right)P\left(pix_2|class\right)...P\left(pix_n|class\right) \tag{2.3}$$

Підставивши у формулу (2.2), отримаємо,

$$c_{map} = \arg\max\left(P(c)P\left(pix_1|class\right)P\left(pix_2|class\right)....P\left(pix_n|class\right)\right) \tag{2.4}$$

На данному етапі, ми стикаємось з першою проблемою - проблемою арифметичного переповненння. Так як вхідні дані мають дуже великий обсяг, то комп'ютеру доведеться перемножувати велику кількість достатньо малих ймовірностей. Для вирішення цієї проблеми, прологарифмуємо формулу (2.4) за будь-якої основи логарифма. У данному випадку це вплине лише на числові значення а не на ранжування класів, адже функція логарифма монотонна.

$$c_{map} = \arg\max\left(\ln\left(P(c)\right) + \sum_{i=1}^{n}\ln\left(\left(P\left(pix_i|class\right)\right)\right)\right) \tag{2.5}$$

Оцінити вирогідність того, чи належить піксель до класу можливо наступним шляхом.

$$P\left(pix_i \middle| class\right) = \frac{pix_{iclass}}{\sum_{i=1}^{n} V_i} \tag{2.6}$$

pix_{iclass} - це піксель, що належить класу

V_i - це унікальний піксель, який ми не зустрічали в жодному класі. У знаменнику має бути сумма по всім унікальним елементам вибірки

Проблема унікальних значень є доволі актуальною. Під час навчання моделі нерідко бувають ситуації, коли ми зустріли значення, якого не було у навчальний вибірці. За таких умов, виникають ситуації, коли піксель має нульову вирогідність, тобто його неможливо віднести до жодного з класів. У розділі 2.2.2 було зазначено, що під час підготовки даних, було урівняно вибірки сонячних панелей та порожніх територій шляхом випадкового вибору пікселів, тому описана вище ситуація циклом має місце бути [13].

Для вирішення проблеми, доречним є використання методу аддитивного згладження, також відомого як згладження Лапласса. Ідея полягає у додавання одиничної константи до кожного пікселя, уявляючи, що ми бачили цей піксель хоча б один раз. Це дозволить отримати значенню дуже маленьку, але не нульову вирогідність.

Зібравши всі формули разом, ми отримуємо кінцевий варіант, за яким і буде рахуватись приналежність до класу.

$$c_{map} = \arg\max\left(\ln(P(class)) + \sum_{i=1}^{n} \ln\left(\frac{pix_{iclass} + 1}{\sum_{i=1}^{n}(V_i + 1)}\right)\right) \tag{2.7}$$

2.3.3 Аналіз точності моделей

Паралельно з навчанням кожної з моделей, ми отримуємо її проміжне передбачення на тестовій вибірці. Ці дані передаються в готовий функціонал, що на їх основі підрахує точність отриманого результата. Ці дії є важливими, адже на основі отриманих точностей ми можемо зробити висновки, чи є сенс використовувати той чи інший алгоритм [14].

Існує декілька стандартних підходів до оцінки точності отриманих результатів. Першочергово, необіхдно ввести важливе поняття для її оцінки, так звану матриця помилок *(confusion matrix)*. У випадку для двох класів, вона буде мати наступний вигляд:

True Positive (TP)	False Positive (FP)
False Negative (FN)	True Negative (TN)

Рисунок 2.6 - Структура матриці помилок

Де значення правого стовпчика *TP, FN* - відповідь алгоритма на даних, а лівого *FP, TN* - дійсні значення. Ці метрикі є необхідними для підстановки їх у формули та оцінки точності.

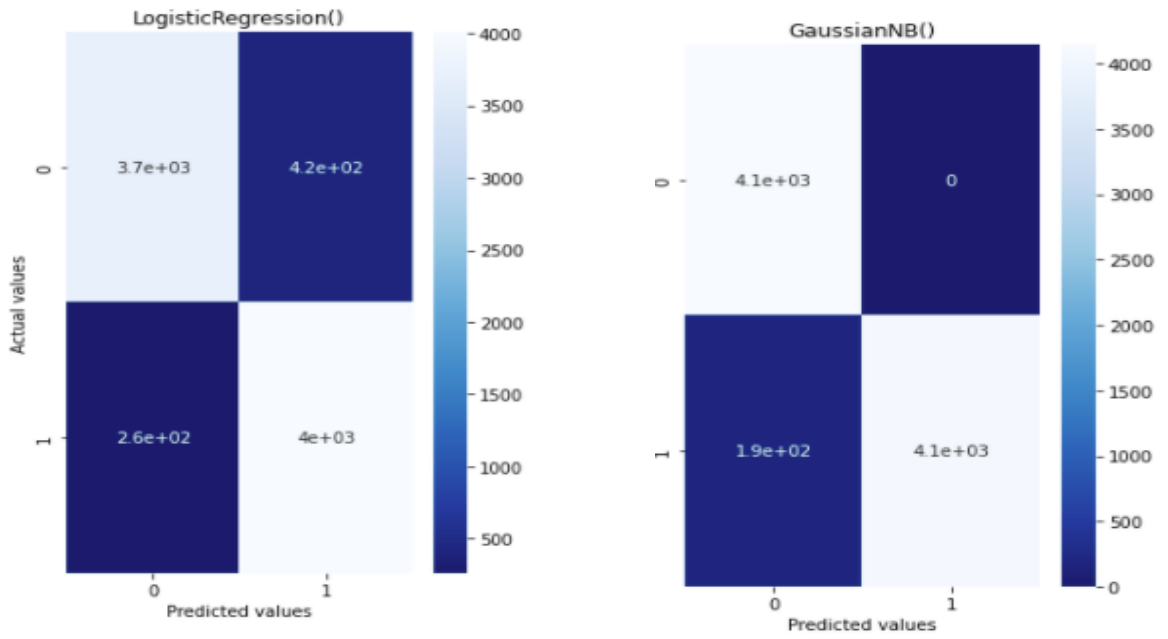

Рисунок 2.7, 2.8 - Приклади матриць помилок методів логістичної регресії та наївного Баєса

Надалі, ми маємо порахувати точність. Найочевиднішою є метрика *accuracy* - кількість правильних відповідей алгоритма.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.8}$$

Ця формула не є точною для класів з нерівною кількістю даних. [14] Наприклад за цією формулою, якщо віднести всі дані до вибірки більшої розмірності, то суто математично ми отримаємо більш високу оцінку, аніж при реальному підрахунку.

Так як у нас класи вже були приведені до однієї розмірності, ми можемо зважати на результат цієї метрики.

Model	Accuracy	AUC
1) LogisticRegression	0.919510	0.92
2) Random_Forest	0.999168	1.00
3) GaussianNB	0.977292	0.98
4) DecisionTreeClassifier	0.997979	1.00

Маємо високі результати точності роботи алгоритма. На жаль, часто він неспівпадає з реальною роботою алгоритма. Існує ще багато методів оцінки алгоритма. Під час написання програми, було використано ще такі метрики, як precision (точність), recall (повнота), F, та інші.

$$precision = \frac{TP}{TP + FP} \tag{2.9}$$

$$recall = \frac{TP}{TP + FN} \tag{2.10}$$

Ці значення дуже зручно рахувати за допомогою методу *create_classification_report*, що надається засобами бібліотеки *Sklearn*. Метод приймає на вхід передбачення алгоритма, та його дійсні значення, тобто дані з тестової вибірки.

За результатами використання наведених формул, було отримано наступні значення [14].

1) Логістична регрессія

	precision	recall	f1-score	support
Solar panel	0.93	0.90	0.92	4143
Nulls	0.91	0.94	0.92	4268
accuracy			0.92	8411
weighted avg	0.92	0.92	0.92	8411

Таблиця 1.1 - Метрики точності алгоритма логістичної регресії

2) Наївний Баєс

	precision	recall	f1-score	support
Solar panel	0.96	1.00	0.98	4143
Nulls	1.00	0.96	0.98	4268
accuracy			0.98	8411
weighted avg	0.98	0.98	0.98	8411

Таблиця 1.2 - Метрики точності алгоритму наївного Баєса

Отримані метрики співпадають з оцінкою точності по формулі (2.8), тому ми можемо зробити висновок, що метрики дійсно мають місце бути. У таблицях 1.1 та 1.2 також зазначено кількість даних, що належать кожному з обраних класів - сонячних панелей та “еталонних нулів”. Але, як

було зазначено, оцінка точності може не співпадати з реальним результатом роботи алгоритма, тому необіхдно зробити передбачення для повної вибірки даних.

РОЗДІЛ 3. ПРАКТИЧНЕ ЗАСТОСУВАННЯ АЛГОРИТМІВ

Цей розділ присвячений практичному застосуванню обраних алгоритмів. У результаті як і проведення теоретичного дослідження, так і навчання алгоритмів та оцінка їх точності, ми отримали змогу обрати алгоритми, та передбачити їх потенційну ефективність в умовах даної задачі. Безпосереднє використання алгоритмів у робочому середовищі дозволить обрати найкращий з них, що може бути використаний для вирішення реальних робочих задач.

3.1 Розпізнавання панелей на знімках

Після того, як алгоритми є навченими, останнім етапом є їх використання на цікавлячих ділянках територій, тобто їх практичне застосування. Під час розпізнавання територій було використано датасет, що складався з нарізаних супутникових знімків, на яких представлено різні ділянки території. Навіть відносно “легковісний” метод розпізнавання, як у випадку даної задачі – алгоритм класифікації, використовує чималий об’єм оперативної пам’яті та забирає багато часу, тож для прикладу роботи алгоритму представлено невелику ділянку території, яку можливо обробити засобами Google Colab.

Першочергово, необхідно “прочитати” всю область даних аналогічним методом, що був обраний для навчання моделей – перетворити зображення в матрицю яскравостей розмірністю NxM, де M – десять спектрів.

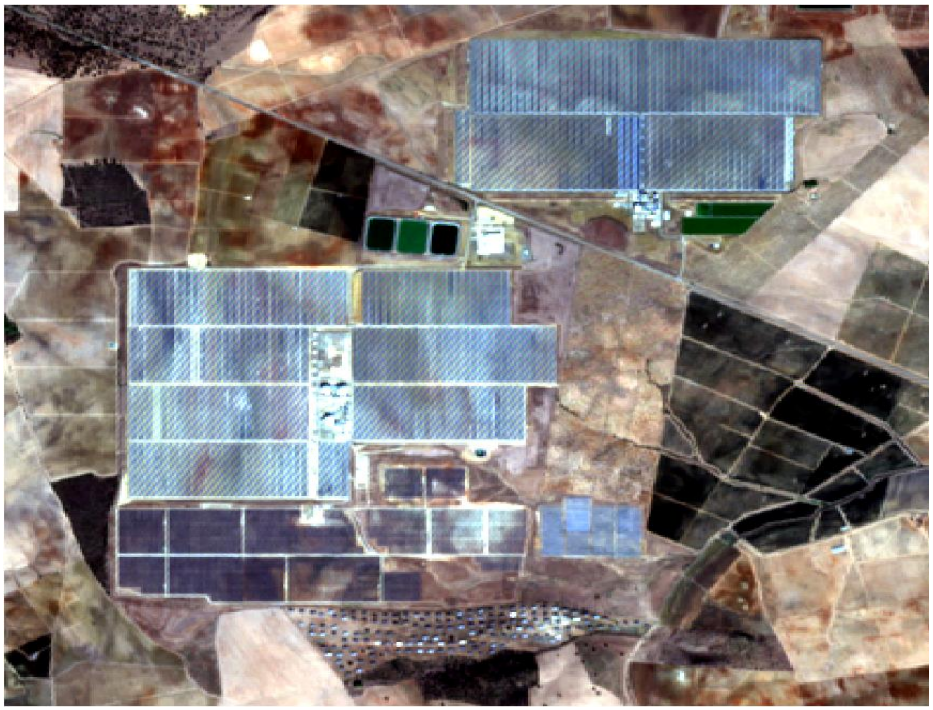

Рисунок 3.1 - Зображення для розпізнавання

На кожній з моделей необхідно викликати метод *pred (заув. - від англійського слова predict)* куди передаємо матрицю пікселів.

- *pred_tree* - передбачення методу дерева рішень
- *pred_forest* - передбачення методу випадкового лісу
- *NB_pred* - передбачення методу наївного Баєса
- *logistic_pred* - передбачення методу логістичної регресії

Як результат роботи алгоритма, ми отримаємо вектор значень {0}, {1} який необхідно перетворити у матрицю, що буде записана як зображення засобами бібліотеки Mathplotlib. Отриману картинку називають маскою зображення.

У спекрті, який за замовчування використовує вищезазначена бібліотека, всі дані, що прийняли значення одиниці інтерпретуються

жовтим кольором. Всі дані що отримали нульове значення представлені темно-фіолетовим кольором.

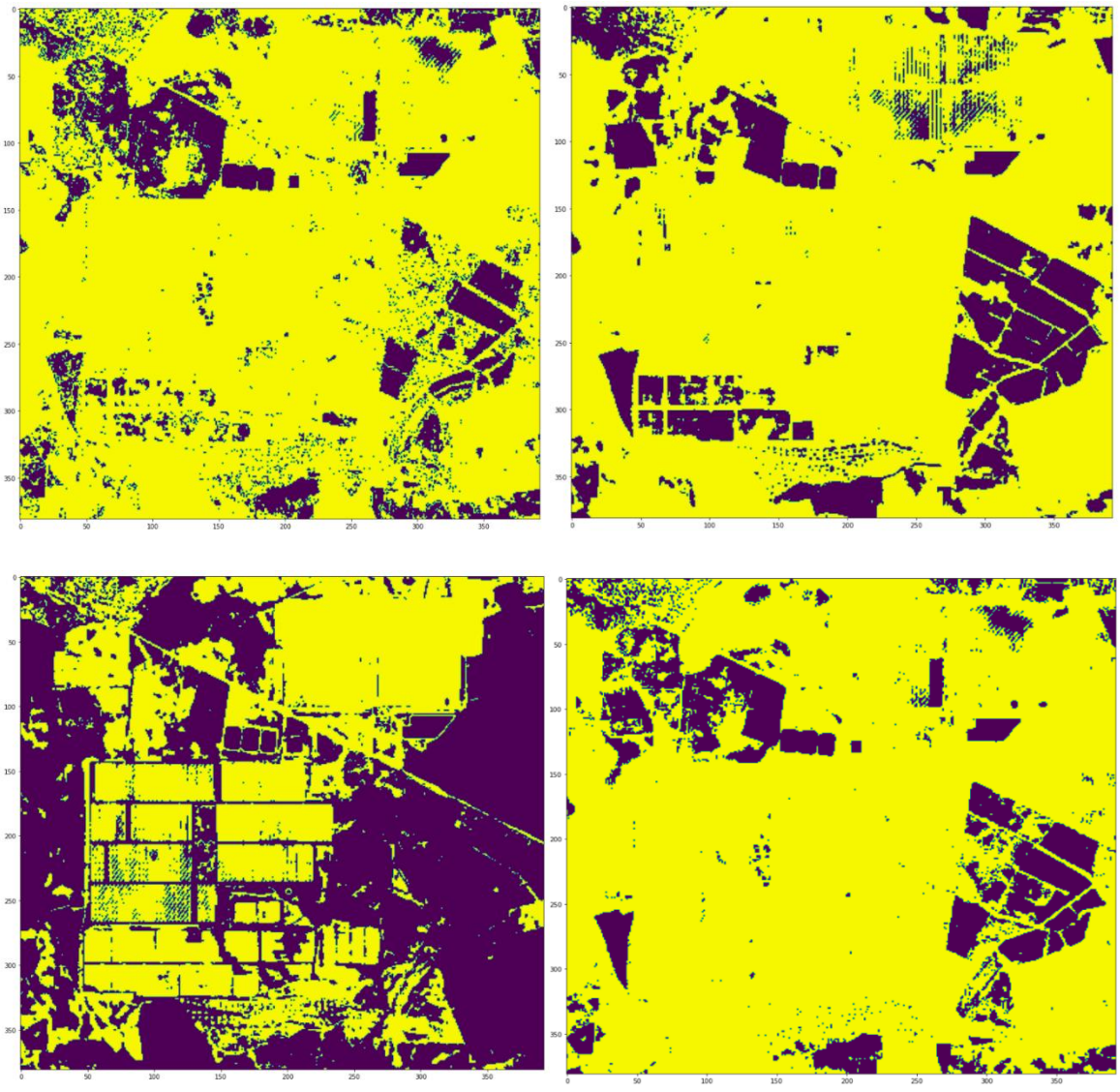

Рис 3.2 – Маски передбачень моделей – Логістична регресія, Дерево рішень, наївний Баєс, випадковий ліс

Як ми бачимо три алгоритми захопили багато інших, не цікавлячих нас територій, виокремивши лише дуже темні ділянки, наприклад ліси, тоді як метод наївного Баєса показує бажаний результат, позначивши як сонячні панелі лише дуже світлі ділянки. Попри це, можливо чітко виокремити

шукані об’єкти на отриманій “масці”, що дозволяє обрати найкращий з представлених алгоритмів для робочих задач.

3.2 Аналіз ефективності

Чому саме цей метод відпрацював краще всіх, попри високі показники точності і у інших алгоритмах? Классифікатор наївного Баєса дає високий результат, незважаючи на те, що при формулюванні методу ми припускаємо незалежність події та той факт, що оцінка ймовірності приналежності до того чи іншого класу не є надто точною.

Основна причина задовільного результата є те, що під час класифікації більш важливу роль грає співвідношення між оцінками йомовірності, а не їх абсолютна величина. Відносні оцінки працюють добре навіть коли дані не є незалежними [15].

У даній задачі дані є залежними, що було видно на аналізу гістограм. Поки залежності або “співпрацюють одна з одною” або повністю пригнічують вплив метод буде надавати гарні результати. Іншими словами, він надає деякий компроміс незалежності даних для подальшого спрощення підрахунків. Також, метод наївного Баєса добре працює за умови не дуже великого обсягу навчальної вибірки.

Таким чином, задача класифікації об’єктів за спектральними ознаками може бути вирішена в методами бінарних алгоритмів. Модель наївного Баєса показав найкращий результат поміж усіх варіантів.

Висновки

У даній роботі було досліджено методи математичної класифікації для обробки супутникових зображень та виокремлення сонячних панелей на них. Було проведено аналіз предметної області та сформульовано підхід до вирішення задачі спектрального аналізу, що дозволило обрати алгоритми класифікації, оцінити їх точність та обрати найкращу модель для вирішення поставленої задачі.

Програмний застосунок надав бажаний результат - на отриманих “масках” можливо чітко розрізнити цікавлячі нас об’єкти сонячних панелей. Було виявлено переваги та недоліки у даному підході. Метод є простим в реалізації, та може бути використаний для вирішення схожих задач пов’язаних з спектральним аналіз зображень, тобто аналізом за кольоровими спектрами - знаходження водойм, хмар та інші специфічних об’єктів. Метод не потребує занадто великої кількості вхідних даних для навчаня, на відміну від нейроних мереж. Але, присутні і недоліки - метод охоплює певну кількість зайвих територій, що схожі за спектральними ознаками, та такий метод неможливо використати для виявлення більш складних об’єктів, наприклад будівель, у яких необхідно виділяти контури.

Отримані результати є корисними при аналізі території задля розвинення зеленої енергії, проведення інвестиційної політики, тощо. Також, розроблений програмний застосунок показує можливість вирішення даної задачі методами математичної класифікації, і може бути розвинений у майбутньому, шляхом оптимізації обраного математичного алгоритма, або поєднання декількох з них.

Список використаних джерел

1. Sentinel-2 [Електронний ресурс]: https://uk.wikipedia.org/wiki/Sentinel-2
2. Растрові та векторні зображення, їхні властивості. Формати файлів растрових і векторних зображень [Електронний ресурс]: https://informatik.pp.ua/uroky/6-klas/konspekty-uchnia/urok-2-rastrovi-ta-vektorni-zobrazhennia-formaty-failiv
3. Бинарная сегментация изображений методом фиксации уровня (Level set method) [Електронний ресурс]: https://habr.com/ru/post/332692/
4. Binary classification of small satellites telemetry data based on deep learning approach http://aait.ccs.od.ua/index.php/journal/article/view/123/180[Електронний ресурс]:
5. Як обрати найкращий метод классифікації з навчанням? [Електронний ресурс]: http://www.50northspatial.org/ua/pick-best-supervised-classification-method/
6. Top 6 Machine Learning Algorithms for Classification [Електронний ресурс]: https://towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501
7. Керована класифікація за допомогою метода паралелепіпеда [Електронний ресурс]: http://www.50northspatial.org/ua/supervised-image-classification-using-parallelepiped-algorithm/
8. Керована класифікація за допомогою метода мінімальної відстані [Електронний ресурс]: http://www.50northspatial.org/ua/supervised-image-classification-using-minimum-distance-algorithm/
9. Практическое руководство по методу максимального правдоподобия [Електронний ресурс]: https://habr.com/ru/company/otus/blog/585610/

10. Reading and Visualizing GeoTiff | Satellite Images with Python [Електронний ресурс]: https://towardsdatascience.com/reading-and-visualizing-geotiff-images-with-python-8dcca7a74510
11. Using Histograms to Understand Your Data [Електронний ресурс]: https://statisticsbyjim.com/basics/histograms/
12. Leo Braimen, Statistics Department University of California Berkeley – C. 1 – 15 - RANDOM FORESTS https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf
13. Наивный байесовский классификатор [Електронний ресурс]: http://bazhenov.me/blog/2012/06/11/naive-bayes.html
14. Метрики в задачах машинного обучения [Електронний ресурс]: https://habr.com/ru/company/ods/blog/328372/
15. Why Do Naive Bayes Classifiers Perform So Well? [Електронний ресурс]: https://highdemandskills.com/naive-bayes-perform/