МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет аеронавігації, електроніки та телекомунікацій
Кафедра авіаційних комп'ютерно-інтегрованих комплексів

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Віктор СИНЄГЛАЗОВ
"___"_____2024р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ **"БАКАЛАВР"**

Тема: Інтелектуальна система обробки зображень радара з синтезованою апертурою

Виконавець: студент групи КП-401Б Швидченко Андрій Олександрович

Керівник: завідувач каф. АКІК, д.т.н., професор Синєглазов Віктор Михайлович

Нормоконтролер: _____ Філяшкін М.К

Київ – 2024

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет аеронавігації, електроніки та телекомунікацій

Кафедра авіаційних комп’ютерно-інтегрованих комплексів

Освітній ступінь: бакалавр

Спеціальність 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Комп’ютерно-інтегровані технологічні процеси і виробництва»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віктор СИНЄГЛАЗОВ

“ ____ ” _____2024 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи студента

Швидченко Андрія Олександровича

1. Тема роботи: «Інтелектуальна система обробки зображень радара з синтезованою апертурою»

2. Термін виконання роботи: з 22.04.2024 р. до 12.06.2024 р.

3. Вихідні дані до роботи: Розробка інтелектуальної системи обробки зображень радара з синтезованою апертурою.

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):

1. Аналіз можливостей радара з синтезованою апертурою;

2. Розробка застосунку для аналізу великих наборів зображень отриманих радаром з синтезованою апертурою.

5. Перелік обов’язкового графічного матеріалу: опис набору даних, принцип роботи алгоритму, результати прогнозування, схеми роботи алгоритмів.

6. Календарний план-графік:

№ п/п	Завдання	Термін виконання	Відмітка про виконання
1.	Аналіз літературних джерел	22.04.2024	
2.	Збір інформації	23.04. 2024	
3.	Аналіз структури, параметрів радара з синтезованою апертурою	24.04.2024-15.05.2024	
4.	Аналіз згорткових нейронних мереж	16.05.2024 – 20.05.2024	
5.	Вибір інструментів для розробки застосунку	21.05.2024 – 25.05.2024	
6.	Розробка застосунку	26.05.2023 – 27.05.2024	
7.	Висновки по роботі	29.05.2024	
8.	Оформлення пояснювальної записки	31.05.2024	
9.	Створення презентації	01.06.2024	

8. Дата видачі завдання «____» _____ 20 __ р.

Керівник кваліфікаційної роботи: _____ Синєглазов В.М.
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання: ___ Швидченко А. О.
(підпис здобувача вищої освіти) (П.І.Б.)

РЕФЕРАТ

Кваліфікаційна робота «Інтелектуальна система обробки зображень радара з синтезованою апертурою» містить 58 ст., 28 рисунків, 1 табл., 18 використаних джерел.

Об’єкт дослідження – обробка зображень радара з синтезованою апертурою за допомогою штучного інтелекту.

Предмет дослідження – використання штучного інтелекту для сегментації та класифікації зображень.

Мета кваліфікаційної роботи – дослідити принцип дії радара з синтезованою апертурою, проаналізувати різні системи для виявлення аномальних об’єктів у ґрунті, розробити інтелектуальну систему для обробки отриманих радаром зображень та зробити висновок про перспективи використання розробленої системи для класифікації вибухонебезпечних об’єктів.

Методи дослідження – аналіз наявних літературних джерел, програмування на мові Python.

Матеріали кваліфікаційної роботи рекомендується використовувати при аналізі наявних систем виявлення аномальних об’єктів під землею, та перспектив використання штучного інтелекту і машинного навчання у процесі розмінування, а також для аналізу методів обробки радіолокаційних зображень.

РАДАР З СИНТЕЗОВАНОЮ АПЕРТОРОЮ, ІНТЕЛЕКТУАЛЬНА ОБРОБКА, ШТУЧНИЙ ІНТЕЛЕКТ МАШИННЕ НАВЧАННЯ, БЕЗПІЛОТНИЙ ЛІТАЛЬНИЙ АПАРАТ

The qualification work

"Intelligent Image Processing System for Synthetic Aperture Radar" contains 55 pages, 28 figures, 1 table, and 18 references.

Keywords: SYNTHETIC APERTURE RADAR, INTELLIGENT PROCESSING, ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, UNMANNED AERIAL VEHICLE.

Object of research: the processing of synthetic aperture radar images using artificial intelligence.

Subject of research: the use of artificial intelligence for image segmentation and classification.

The aim of the qualification work: to study the principle of operation of synthetic aperture radar, analyze various systems for detecting anomalous objects in the ground, develop an intelligent system for processing radar-obtained images, and conclude on the prospects of using the developed system for classifying explosive objects.

Research methods: analysis of available literature sources, programming in Python.

The materials of the qualification work are recommended for use in analyzing existing systems for detecting anomalous underground objects and the prospects of using artificial intelligence and machine learning in the demining process, as well as for analyzing methods of radar image processing.

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ

ШІ – Штучний інтелект

МН – Машинне навчання

CNN (ЗНМ) – Convolutional neural network (Згорткова нейронна мережа)

SAR (РСА) - Synthetic aperture radar (Радар із синтезованою апертурою)

БПЛА – Безпілотний літальний апарат

ЗМІСТ

Вступ…… **8**

1. Радар із синтезованою апертурою. Структура, параметри, та засоби застосування……..……...…….. **10**

1.1. Застосування радарів із синтезованою апертурою…… 10

1.2. Фізичні основи побудови радара із синтезованою апертурою……..… 12

1.3. Структура радара із синтезованою апертурою…….. 15

1.4. Параметри радара із синтезованою апертурою……. 16

1.5. Методи отримання даних радаром з синтезованою апертурою……... 22

1.6. Алгоритм обробки SAR зображень……..……... 24

1.7. Застосування радарів з синтезованою апертурою для виявлення вибухонебезпечних предметів……...……..... 25

1.7.1. Findmine Ground Penetrating Synthetic Aperture Radar (FM-GPSAR)…..... 25

1.7.2. Forward Looking Ground Penetrating SAR (FLGPSAR)……..……. 27

1.7.3. Система виявлення мін розтяжок з використанням радара з синтезованою апертурою……..……. 29

2. Інтелектуальна обробка зображень. Топологія згорткових нейронних мереж ……... **31**

2.1. Огляд методів інтелектуальної обробки зображень……..…… 31

2.2. Згорткові нейронні мережі……..……. 33

2.2.1. Структура згорткової нейронної мережі……..……...…. 33

2.2.2. Операція згортки……..…….. 35

2.3. Навчання згорткових нейронних мереж……...…….. 37

2.4. CNN архітектури для задач обробки зображень……...……. 38

3. Архітектура YOLO. інтелектуальна обробка зображень отраманих радаром з синтезованою апертурою…….…… **40**

3.1. Обґрунтування вибору архітектури YOLOv8…… 40

3.2. Аналіз архітектури YOLOv8…… 42

3.3. Інструменти для розробки програми…….. 45

3.4. Розробка програми…… 48

3.4.1. Навчальна вибірка……...……...…. 48

3.4.2. Процес підготовки даних для навчання моделі……..… 49

3.4.3. Створення файлу з конфігураційними параметрами……..… 50

3.4.4. Навчання моделі……..…….. 50

3.4.5. Створення інтерфейсу користувача…….…… 51

3.5. Огляд роботи програми…… 52

Висновки……..……..…… **55**

Список бібліографічних посилань використаних джерел ……..…….…… **56**

Додаток А……...……...…. 58

Додаток Б……...……... 59

Додаток В……...…. 60

Додаток Г……...…. 61

ВСТУП

Дослідження земної поверхні допомагає людям дізнатись більше про історію нашої планети та нас самих. З плином часу наші технології стають потужнішими, і дозволяють нам з більшою ефективністю проводити дослідження землі. Ми досліджуємо землю не тільки для поглиблення наших знань про історію, але і для практичних цілей наприклад, при будівництві складних споруд, таких як, мости, бурові платформи для видобутку нафти та газу, тощо.

Застосування РСА з використанням георадара, дозволяє безпечно інспектувати важкодоступні території без прямого контакту з ґрунтом. Георадри застосовують для перевірки цілісності труб або визначення порожнеч у ґрунті. На отриманих з георадара зображеннях можна побачити глибину на якій закопано об’єкт та його довжину. Також не менш важливим фактором є те, що такі зображення дозволяють нам відрізняти вибухонебезпечний предмет від інших металевих предметів, таких як труби, сміття тощо.

У сучасному світі розмінування великих територій стало актуальною проблемою. Розмінування — це комплекс заходів, спрямованих на виявлення та знешкодження вибухонебезпечних предметів, таких як міни та нерозірвані боєприпаси. Зазвичай цей процес здійснюється командами саперів, оснащених металошукачами. Хоча такий метод є досить ефективним, він також є дуже небезпечним для саперів, які ризикують своїм життям під час виконання завдань. Інженери активно працюють над створенням безпечніших рішень для розмінування, зокрема використанням наземних дронів для пошуку вибухонебезпечних предметів. Останні розробки в цій сфері показують, що ми рухаємося в правильному напрямку, створюючи технології, які зможуть зменшити ризик для людей.

Значним аспектом сучасних технологій є обробка отриманих зображень. Використання штучного інтелекту (ШІ) у цьому процесі є ефективним рішенням, оскільки ШІ має потужні можливості для автоматизації обробки та аналізу даних, отриманих за допомогою РСА. Штучний інтелект здатний швидко та точно обробляти великі обсяги інформації, що значно прискорює процес дослідження та зменшує ймовірність помилок.

Метою даної роботи є дослідження можливостей радара з синтезованою апертурою для виявлення об'єктів, закопаних на невеликій глибині, а також розробка штучного інтелекту для обробки отриманих зображень. У роботі розглядаються різні аспекти розмінування, технічні засоби, що використовуються для цього процесу, та класифікація вибухонебезпечних предметів. Це дослідження має на меті не тільки поглибити наші знання про підземні об'єкти, але й сприяти розвитку безпечніших та ефективніших методів виявлення та знешкодження вибухонебезпечних предметів.

РОЗДІЛ 1
РАДАР ІЗ СИНТЕЗОВАНОЮ АПЕРТУРОЮ. СТРУКТУРА, ПАРАМЕТРИ, ТА ЗАСТОСУВАННЯ

1.1. Застосування радарів із синтезованою апертурою

Радар з синтезованою апертурою (SAR) - це система радіолокації, яка використовує радіові хвилі для створення високоякісних зображень поверхні Землі. Основна ідея полягає в тому, щоб імітувати велику апертуру антени, комбінуючи сигнали, отримані від рухомої антени під час її руху. Радари із синтезованою апертурою мають широке коло застосувань в різних галузях. SAR дозволяє отримувати високоякісні зображення з роздільною здатністю, навіть при обмеженій фізичній апертурі антени. Це корисно для створення детальних карт, вивчення ландшафту, виявлення змін на поверхні та моніторингу природних ресурсів.

Технологія SAR була великим проривом в обробці зображень, що дозволило отримувати зображення високої роздільної здатності з будь-якої відстані, надаючи SAR унікальну здатність серед усіх технологій зображення земної поверхні. Досягнення в комп'ютерних технологіях революціонізують обробку радіолокаційних даних, дозволяючи створювати те що, здавалося недосяжним лише кілька років тому.

Можливо, найкращим прикладом реалізації SAR є німецька місія TanDEM-X, яка створює об'єднану модель високої роздільної здатності висот по всій планеті (Інститут мікрохвиль та радіолокації Німецького аерокосмічного центру, 2013 р.). Ця місія включає в себе пару SAR супутників, які літають у формації на відстані до 200 м один від одного, випромінюючи та записуючи імпульси мікрохвильового випромінювання. Всі виміри орбітального руху, дальності, амплітуди та фази здійснюються з найвищою точністю, незалежно від хмар та відстані.

Роль SAR в дистанційному зондуванні важко недооцінити. Дистанційне зондування - це процес збирання та інтерпретації інформації про об'єкти або явища на земній поверхні без прямого фізичного контакту. Це особливо корисно в гуманітарному розмінуванні, в розділі 1.7 цієї роботи, описано системи з використанням SAR для виявлення мін.

SAR використовують в археології для виявлення археологічних об’єктів, структур, які приховані під землею, та геологічних формацій. Він може виявляти структури, такі як руїни, долини, гори та річки.

Завдяки тому, що технологія SAR створює детальні радарні зображення незалежно від погодних умов і часу доби, вона набула широкого використання в сфері безпеки і оборони. Отримані високоякісні зображення використовуються для військовими для планування операцій, розвідки, виявлення важливих інфраструктурних об’єктів противника, моніторинг кордонів та інших завдань військової сфери.

Дані зібрані таким радаром, також успішно застосовуються в сільському господарстві. Отримані забраження допомагають визначити вологість ґрунту, оцінити ріст та здоров’я тієї чи іншої культури. SAR дозволяє фермерам передбачати врожай. У випадку стихійного лиха, можна оцінити масштаб збитків для агрофірми. Такі можливості досить корисні для аграріїв. [1]

Тема екології набуває все більшої уваги науковців, вивчення зон клімату та виявлення та аналіз екологічних проблем потребує потужних інструментів. Саме таким інструментом є SAR, адже він допомагає спостерігати за змінами в рослинності, водних ресурсах, лісах та інших екосистемах, вивчати геологічні процеси, океанські струми, льодовики, вулкани, проводити моніторинг погоди, виявляти стихійні лиха та інші явища на Землі та інших планетах. Наприклад на рис. 1.1 можна побачити швидкості водного потоку в прибережній зоні, отримані за допомогою спільної комбінації двох SAR-зображень, отриманих літаком з двома радіолокаційними антенами, які були зсунуті вздовж треку. [2]

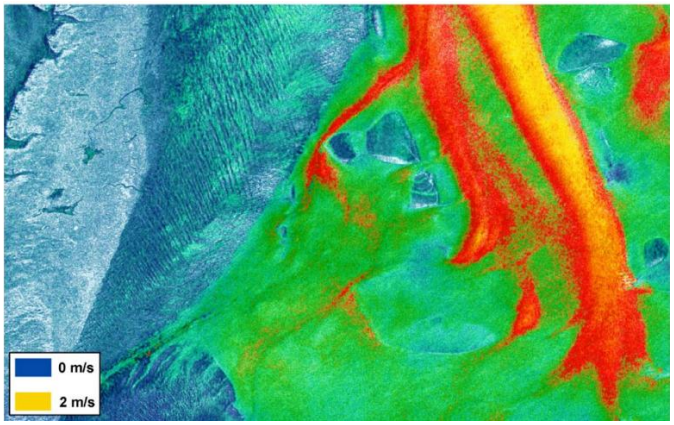

Рис. 1.1. Вимірювання швидкості водного потоку у Вадденському морі, на острові Амеланд, Нідерланди, за допомогою інтерферометрії вздовж треку (дані X-діапазону, отримані в режимі пінг-понг системою E-SAR німецького центру досліджень повітряного простору DLR)

1.2. Фізичні основи побудови радара із синтезованою апертурою

Вперше технологія SAR була представлена в 1957 році, вона використовувала фотоплівку для запису радіолокаційних даних і процесор зображень, виготовлений з лінз. Сьогодні цифрові пристрої та інше обладнання для збору даних можуть зберігати дані для обробки в автономному режимі або навіть обробляти зображення в режимі реального часу.

Зображення SAR здаються майже фотографічними, але це не фотографія, це двовимірна голограма (Рис. 1.2). На відміну від зображення з супутника, радар не вимірює цільову сцену зверху, він вимірює її з боку на досить значній відстані. Отримане зображення представляє собою вид з висоти пташиного польоту з багатьма тінями, де кожен піксель прямо відображає траєкторію польоту літака/супутника по дальності та азимутній координаті. [3]

Рис 1.2. РСА зображення Національної Лабораторії Сандія

Концепція роботи радара наступна - радар випромінює радіосигнали, а потім приймає зворотні відлуння імпульсів, формуючи у результаті зображення об’єкта, що відбиває сигнали. Як відомо, що більша апертура антени, то вищою буде роздільна здатність отриманого зображення. Отже, щоб отримати зображення високої якості, необхідно збільшити фізичну апертуру, тобто застосувати антену великого розміру. Кожному імпульсу притаманні унікальні характеристики. [4]

Імпульси відправляються з частотою повторення імпульсів у діапазоні 2000 на секунду і більше. Як показано на рис. 1.3, ці імпульси розсіюються в кожному прямому іоні, і невелика частина, яка називається радіолокаційним зворотнім розсіянням, повертається на антенну. Радар вимірює характеристики ехо-сигналів, включаючи час проходження імпульса від антени до землі і назад до антени, силу відбиття і фазу зворотної хвилі. Іншими словами, радар може визначити, чи повертається хвиля на вершині, у ямі або десь посередині. [5]

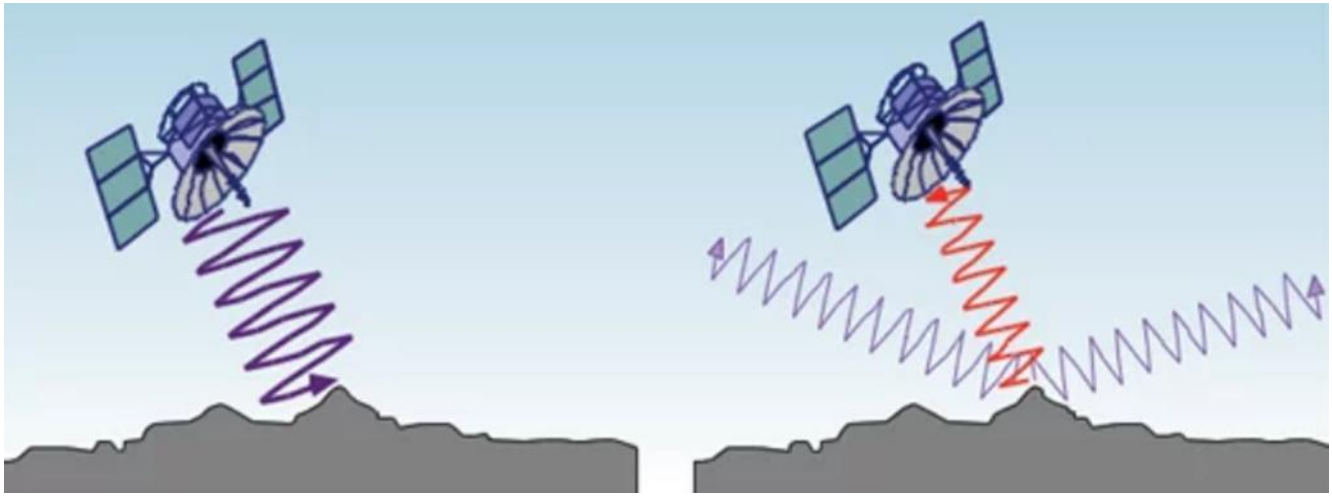

Рис. 1.3. Розсіювання імпульсів

Час, потрібний для проходження імпульсу, використовується для визначення відстані від антени до землі. Відстань (D) легко розрахувати - вона дорівнює швидкості імпульсу, яка представляє собою відому швидкість світла (Δc), помножену на час проходження туди і назад (T), поділене на два:

$$D = \frac{\Delta cT}{2}$$

При русі супутника вперед, об'єкти виявляються багато разів променем радару. Таким чином, розсіяні сигнали піддаються азимутному доплерівському зсуву, який є постійним по дальності і змінюється від вищих частот, коли об'єкт наближається, до нижчих частот, коли об'єкт поступово віддаляється. Антена супутника використовується в декількох точках вздовж азимутальної траєкторії і тим самим синтетично подовжується до кількох разів своєї фактичної довжини, що дозволяє системі вимірювати об'єкти з азимутальною та дальністю на метровому рівні роздільної здатності. У сучасних системах SAR були розроблені механізми керування антеною для подальшого зменшення цієї роздільної здатності до підметрового рівня [6].

Процес роботи SAR-системи можна коротко описати наступним чином, система випромінює радіосигнали в напрямку поверхні Землі, антена приймає відбиті сигнали від різних точок на поверхні. Час повернення сигналу залежить від відстані між радаром і об'єктом на поверхні.Отримані сигнали обробляються для отримання зображення. Для цього використовуються різні методи обробки сигналів,

такі як фільтрація, компенсація зміщення, компенсація руху платформи тощо.З оброблених даних формується високоякісне зображення поверхні Землі, яке може бути використане для різних застосувань, таких як картографія, зондування, військові операції тощо.

Зазвичай ширина радіохвиль, які застосовуються радаром цього типу, складає від декількох метрів до пари сантиметрів, що набагато більше ніж довжина хвилі видимого світла для створення звичайних оптичних зображень. Довжини хвиль SAR входять до мікрохвильової частини спектру (див. рис. 1.4).

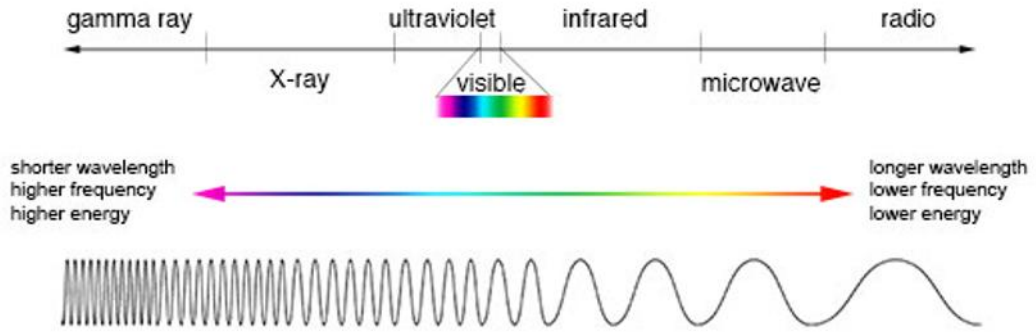

Рис. 1.4. Порівняння довжини хвилі, частоти та енергії для електромагнітного спектра

Найважливішою властивістю радара для візуалізації є те, що його відносно довгі хвилі проникають крізь шари, пил і навіть вулканічний попіл, і він може отримувати зображення незалежно від більшості погодних умов. Зважаючи на те, що Земля це океанічна планета, з великою кількістю водяної пари, яка постійно конденсується в хмари над великими областями поверхні Землі, радар є основною технологією дистанційного зондування і стає основним джерелом зображень, коли хмарний покрив заважає іншим засобам отримання даних.

1.3. Структура радара із синтезованою апертурою

Основні частини системи SAR зображені на рис. 1.5. Блок формування імпульсів генерує імпульси з шириною смуги, відповідно до бажаної дальньої роздільної здатності. Вони будуть посилені передавачем і передані на антену через циркулятор. Приймач отримує вихідний сигнал антени (екхо сцени), посилює їх до

відповідного рівня і застосовує полосопропускний фільтр. Після демодуляції і A/D-перетворення сигналів процесор SAR починає обчислювати зображення SAR. Додаткова інформація про рух буде надана системою вимірювання руху. Радарний блок управління організовує послідовність операцій, зокрема графік часу.[7]

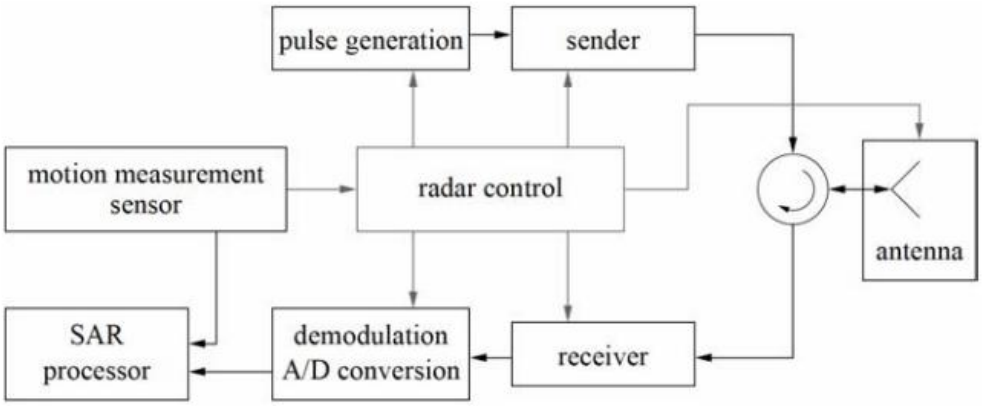

Рис. 1.5.Структурна схема радара з синтезованою апертурою

Можна виділити основні компоненти SAR-систем:

– Антена – це,найважлива частина радару це антена, вона відповідає за випромінювання радіосигналів і приймання ехо сигналів від поверхні Землі. У системі SAR антена може бути рухомою для забезпечення необхідної апертури.

– Трансмітер та приймач, вони відповідають за генерацію радіосигналів, їх відправлення через антену, а також за прийом відбитих сигналів.

– Платформа зі зв'язком Для створення зображень поверхні Землі, антена має бути рухомою відносно об'єкта. Це може бути досягнуто за допомогою рухомої платформи (літак, супутник, БПЛА), на якій розташований радар.

1.4. Параметри радара із синтезованою апертурою

Геометрія отримання радіолокаційних даних. Радари можуть збирати дані під різними кутами щодо напрямку польоту. Як правило, дані збираються під прямим кутом до напрямку польоту. Але деякі радари можуть змінювати кут збору даних, щоб охопити райони перед або позаду основного напрямку.

Кут, під яким радар спрямований вниз від горизонту, називається кутом припливу. Цей кут визначається як різниця між вертикальним вектором у точці

спостереження та лінією, що з'єднує цю точку з антеною. Кут ковзання дорівнює доповненню кута припливу. Наприклад, кут злиття 55° відповідає куту ковзання 35° і навпаки.

Процес збору даних радаром показаний на рисунку 1.6. Під час цього процесу радар випромінює сигнали під широким кутом і збирає дані з безлічі окремих імпульсів. Це дозволяє створювати радіолокаційні зображення з інформацією про дальність і азимут. Діапазон відноситься до розміру вздовж широкого кута, а азимут відноситься до напрямку, ортогонального до цього кута.

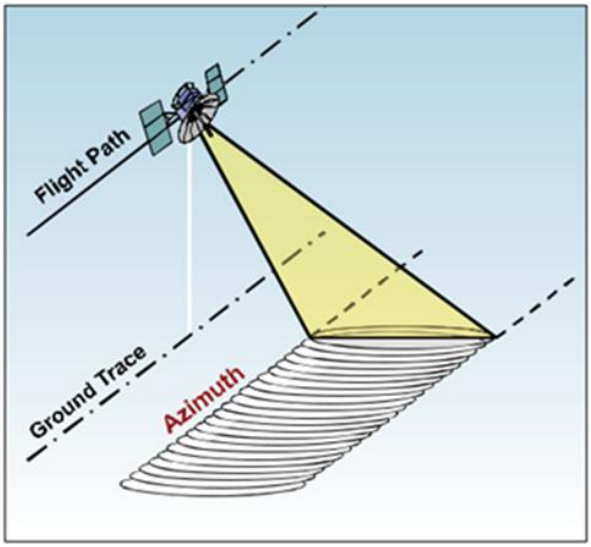

Рис. 1.6. Процес збору даних радаром

Довжина хвилі. Мікрохвилі - це та частина електромагнітного спектра, довжина хвиль якої значно більша, ніж у видимій світловій області, тобто в діапазоні десятків сантиметрів. Проникнення є ключовим фактором для вибору довжини хвилі: чим довша довжина хвилі (коротша частота), тим сильніше проникнення в рослинність і ґрунт. В таблиці 1.1 можна побачити які довжини хвиль використовують SAR системи.

Таблиця1.1

Основні характеристики хвиль які використовуються

Найменування	Діапазон частот	Довжина хвилі	SAR системи
P-band	250—500 МГц	~ 65 cm	AIRSAR
L-band	1—2 ГГц	~ 23 cm	JERS-1 SAR, ALOS PALSAR
S-band	2—4 ГГц	~ 10 cm	Almaz-1
C-band	4—8 ГГц	~ 5 cm	ERS-1/2 SAR, RADARSAT-1/2, ENVISAT ASAR, RISAT-1
X-band	8—12 ГГц	~ 3 cm	TerraSAR-X-1, COSMO-SkyMed
K-band	18—26,5 ГГц	~ 1.2 cm	Для військових потреб

Поляризація. Незалежно від довжини хвилі, радар може передавати горизонтальні (H) або вертикальні (V) вектори електричного поля і приймати як горизонтальні (H), так і вертикальні (V) повернені сигнали, або обидва (Рис. 1.7). Основні фізичні процеси, відповідальні за повернення з однаковою поляризацією (HH або VV), - це квазі-дзеркальне відображення поверхні. Перехресне повернення (HV або VH) зазвичай слабкіше і часто пов'язане з різними відбиттями через, наприклад, нерівності поверхні.[8]

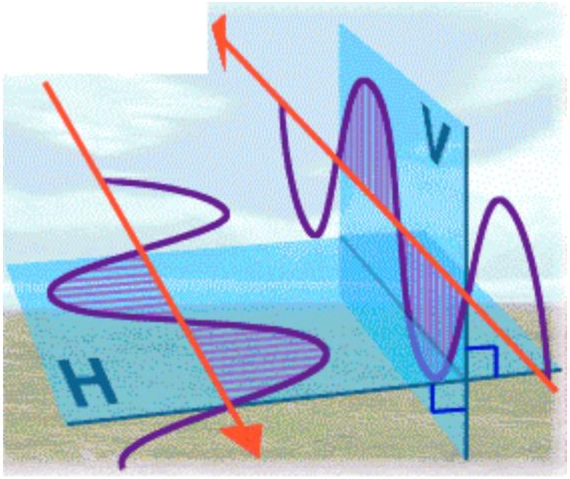

Рис. 1.7. HV поляризація

Кут падіння. Кут падіння (θ) визначається як кут, утворений радіолокаційним променем і лінією, перпендикулярною до поверхні. Мікрохвильові взаємодії з поверхнею складні, і різні відбиття можуть відбуватися в різних кутових областях. Повернення зазвичай мають велику силу при низьких кутах падіння і зменшуються зі збільшенням кута падіння (Див. рис 1.8).

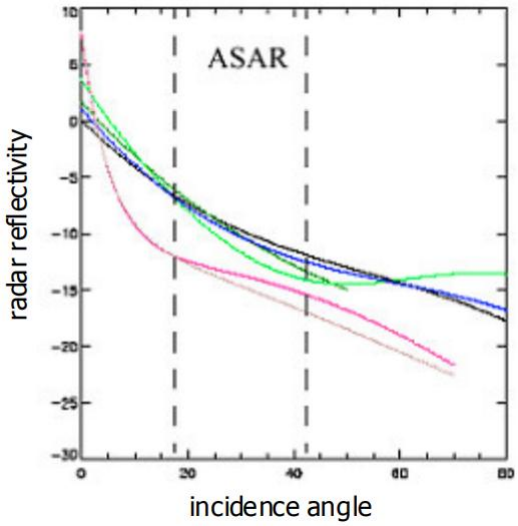

Рис. 1.8. Діаграма показує зміну радіорефлективності для різних класів покриття землі (кольорові лінії), тоді як пунктирні лінії виділяють діапазон області для даних ENVISAT ASAR

Те якими будуть радіолокаційні зображення визначає геометрія отримання сигналів, характером відображення об’єкта та способом випромінювання. На рис. 1.9 можна побачити види відображень таких як, кутові, дзеркальні та дифузні.

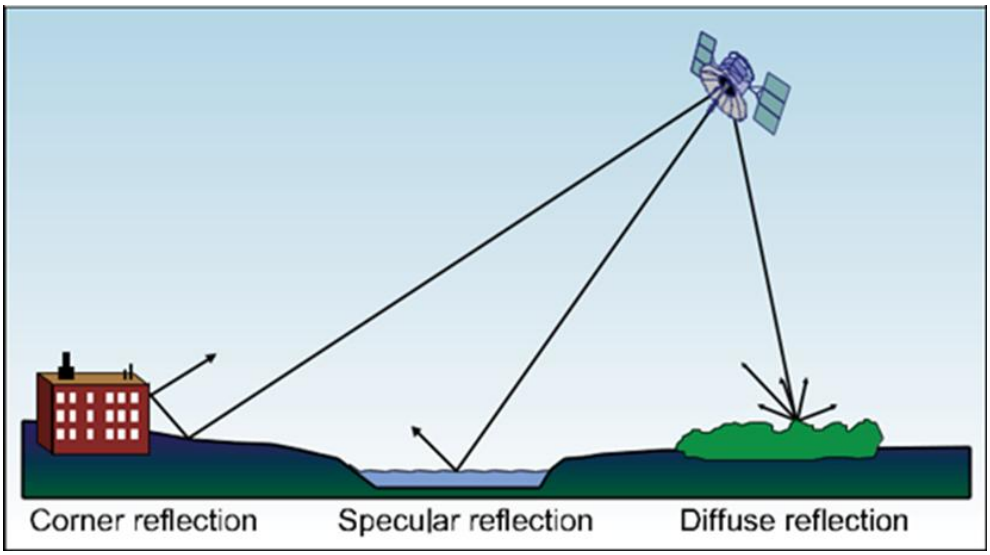

Рис. 1.9. Види відображень від різних об’єктів

Дерева створюють дифузне відображення, воно призводить до помірного зворотнього розсіювання і сірим пікселям на зображенні. На рис. 1.10 можна побачити, як довжина хвиль впливає на розсіювання сигналу. Наприклад, сигнал на С-діапазоні проникає лише в верхні шари крони лісу, тому він переважно зазнає розсіювання на нерівності з додаванням обмеженої кількості об'ємного розсіювання. Але довгі хвилі, сигнал на діапазоні L або P, проникає набагато глибше, відбиваючись від ґрунту та стовбура дерева. Використання довгих хвиль може бути корисним коли потрібно знайти схований у гущавині лісу об’єкт.

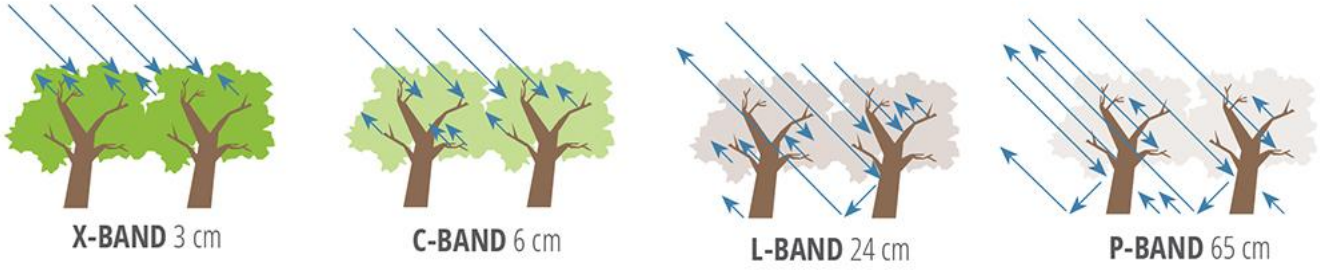

Рис. 1.10 Вплив довжини хвилі на проникнення та розсіювання сигналу через крони дерев

Вода в озерах або спокійних річках, створює дзеркальне відображення, і на зображеннях з радара вона зазвичай досить темна. Вода в океанах має більш нерівну поверхню і відображення від хвиль будуть досить яскравими і помітними на готовому зображенні (Рис. 1.11).

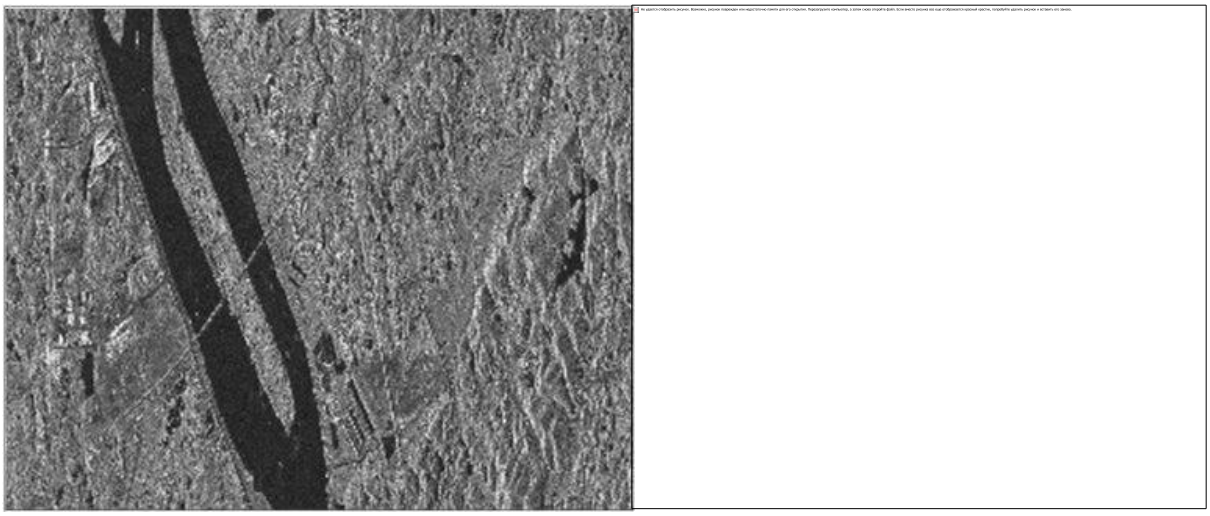

Рис. 1.11. Відображення річок та океану

Будівлі та інші вертикальні об’єкти, мають сильну віддачу, на зображеннях виглядають білими.

На рис. 1.12 можна побачити комбінований ефект зеркальних, дифузних і кутових відображеннях. Можна побачити як контрастують між собою річки, земна поверхня, будівлі та мости.

Рис. 1.12. Зображення Вашингтона, округ Колумбія

Аналіз параметрів радара з синтезованою апертурою допомагає визначити оптимальні характеристики для конкретного застосування. Наприклад, для

високошвидкісних місій може бути важливою швидка реакція та висока роздільна здатність, тоді як для зондування Землі на великих відстанях буде більш важливою велика ширина смуги та довгий часовий розрив.

1.5. Методи отримання даних радаром з синтезованою апертурою

Принцип роботи методу Stripmap. При роботі в режимі Stripmap антена антена залишається нерухомою у певному напрямку. Промінь антени сканує поверхню Землі з постійною швидкістю, створюючи безперервне зображення області.Процес передачі та прийому сигналу відбувається паралельно до руху платформи, що дозволяє отримати високоякісне зображення з високою роздільною здатністю. Після прийому сигналу використовуються алгоритми обробки, такі як обернена фільтрація, для утворення кінцевого зображення.

Варто зауважити, що режим Stripmap є найбільш поширеним режимом. У випадку CAP ERS-1/2 та JERS-1 антена була нерухомою, що робить неможливим вибір області зображення. Останнє покоління SAR-систем, таке як RADARSAT-1/2, ENVISAT ASAR, ALOS PALSAR, TerraSAR-X-1, COSMOSkyMed та RISAT-1, надають можливість вибору різних режимів смуги.

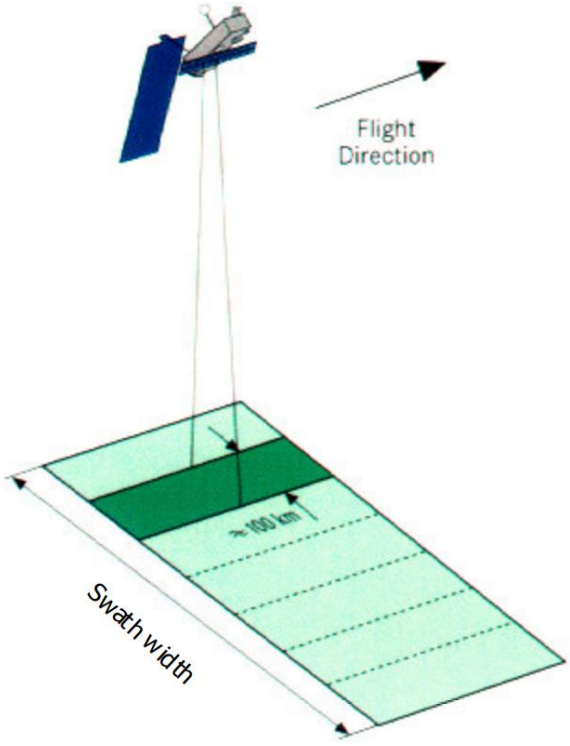

Рис. 1.13. Отримання даних методом Stripmap

Принцип роботи у режимі ScanSAR. В той час коли метод Stripmap обмежується вузькою смугою, ScanSAR досягає розширення смуги за допомогою керованого антенного променю. Замість того, щоб сканувати одну смугу високої роздільної здатності, радар сканує кілька смуг з меншою роздільною здатністю, тим самим покриваючи більшу площу за менший період часу. Радіолокаційні зображення можуть бути синтезовані шляхом сканування кута падіння та послідовного синтезування зображень для різних положень пучка. Зона, зображена з кожного конкретного пучка, утворює під-смугу. Цей метод дозволяє робити зображення широких областей земної поверхні. Основним недоліком є низька азимутальна роздільна здатність.Принцип ScanSAR полягає в тому, щоб поділити час роботи радару між двома або більше окремими під-смугами таким чином, щоб отримати повне охоплення кожної.

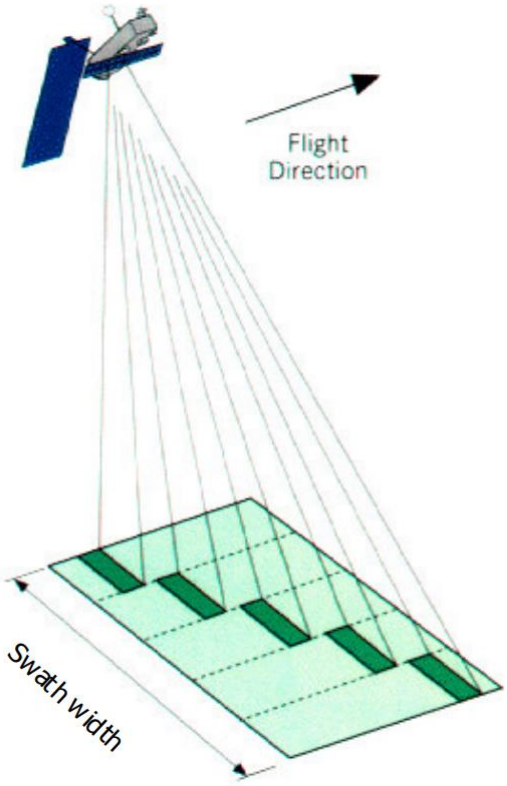

Рис. 1.14. Робота у режимі ScanSAR

Принцип роботи методу Spotlight. Під час збору даних методом Spotlightантена спрямовується на конкретну точку на поверхні Землі, забезпечуючи високу роздільну здатність в цій області. Цей метод потребує точного визначення

позиції та орієнтації платформи, оскільки радар повинен точно визначити положення кожного пікселя на зображенні. Після збору даних застосовуються алгоритми утворення зображення, такі як обернене проектування, для отримання високоякісного зображення.

Переваги методу Spotlight над методом Stripmap:

- Метод Spotlight пропонує більшу роздільну здатність по азимуту, ніж можна отримати в режимі Stripmap, використовуючи ту ж саму фізичну антену.
- Зображення у режимі Spotlight надають можливість отримання зображення сцени з різних кутів огляду під час одного прольоту.
- Метод Spotlight дозволяє ефективно зображувати кілька менших сцен, тоді як режим Stripmap природно зображує довгу смугу місцевості.

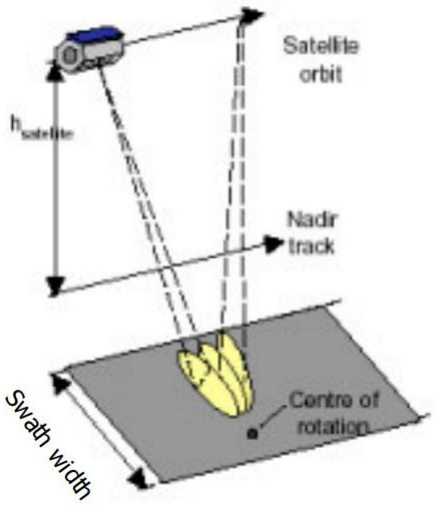

Рис. 1.15. Робота методу Spotlight

1.6. Алгоритм обробки SAR зображень

Щоб отримати готове зображення, отримані радаром дані необхідно обробити. Існує безліч алгоритмів обробки так званих “сирих” даних, про деякі сучасні, одним з найпоширеніших в контексті SAR є Range-Doppler algorithm (R-D). Винайдений в 1978 році,цей алгоритм і досі використовується для утворення високоякісних радарних зображень, що відображають рельєф та об'єкти на земній поверхні з високою роздільною здатністю та точністю.R-D алгоритм

використовується для обробки зображень з радірів, що працюють за методом Stripmap.

Основна ідея алгоритму Range-Doppler полягає у використанні швидкого перетворення Фур’є для утворення радарного зображення.

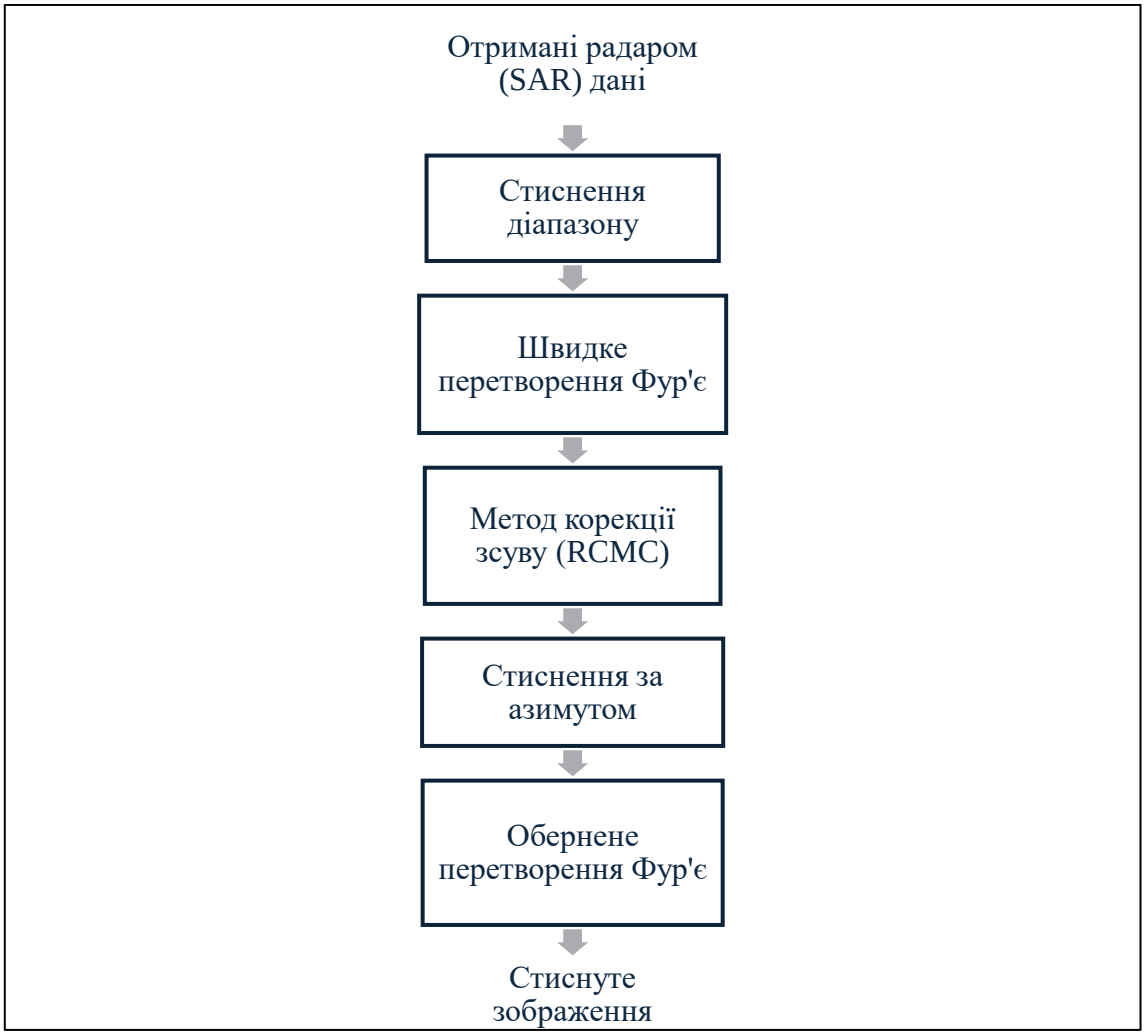

Рис. 1.16. Послідовність виконання алгоритму Range-Doppler

1.7. Застосування радарів з синтезованою апертурою для виявлення вибухонебезпечних предметів

Радари з синтезованою апертурою стають все більш доступними для широкого кола дослідників, що в свою чергу розширює можливості та сфери застосування цих радарів. Мета цього підрозділу ознайомити читача з наявними системами виявлення та локалізації вибухонебезпечних предметів на основі SAR.

1.7.1. Findmine Ground Penetrating Synthetic Aperture Radar (FM-GPSAR)

«FM-GPSAR» – це комплекс набазі БПЛА (Рис. 1.17) для виявлення та локалізації закопаних на невелику глибину об’єктів таких як міни та нерозірвані боєприпаси. GPSAR дозволяє виявляти об'єкти під поверхнею землі або приховані в густому рослинному покриві. Глибина проникнення, однак, значно зменшується в умовах вологого середовища. За допомогою системи GPSAR не потрібно, щоб оператор системи безпосереднью піддавав себе значному ризику. Основні проблеми системи - це точність локалізації, динамічний діапазон радару, синхронізація датчиків та обчислювально інтенсивна обробка даних, схожа на томографію. Підозрілі місця від GPSAR можуть бути подальше класифіковані за допомогою нашого дрону, що несе металошукач.[9] Комплекс розроблений некомерційною компанією «Findmine», заснованою в Ілертіссені, Баварія. Компанія «Findmine» займається розробками на базі БПЛА в сфері безпечного розмінування.

Рис. 1.17. Зовнішній вигляд системи

На сайті компанії виробника [9] наведено такі технічні характеристики та функції системи:

–БПЛА з радаром для проникання в грунт у частотному діапазоні 1 ГГц - 4 ГГц;

–Роздільна здатність дальності на землі за шириною смуги радару;

– Роздільна здатність поперечного зміщення за синтетичною апертурою;

–Роздільна здатність глибини за моделюванням ґрунту на основі ЦМВ та кількома перспективами;

– Глибокі зображення до -20 см отримані після інтенсивної офлайн обробки;

– Алгоритми вимагають точної (+/-1 см) реконструкції траєкторії польоту;

– Швидкість пошукового польоту 2700 м²/год;

– Час обробки 3–30 годин на кожну годину пошукового польоту;

–Обробка залежно від роздільної здатності вокселів та відстані в спостереженні;

– Типова роздільна здатність вокселів 1 см – 5 см.

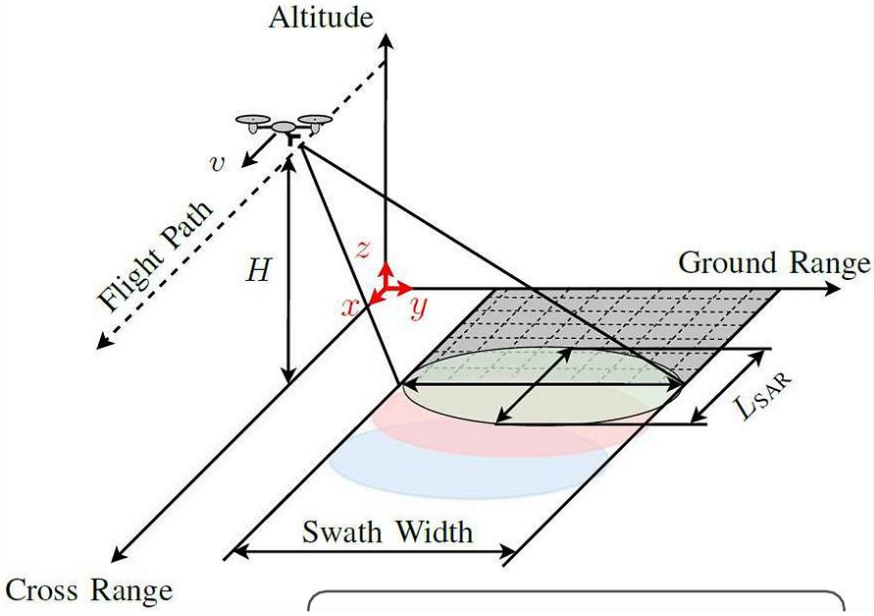

Рис. 1.18. Геометрія отримання даних системою

1.7.2. Forward Looking Ground Penetrating SAR (FLGPSAR)

FLGPSAR – це система на базі газонного транспортного засобу (turf vehicle) (Рис. 1.18), яка здатна виявляти як металеві так і пластикові міни на шляху транспортного засобу. Це стає можливим завдяки напрямленому вперед широкосмуговому GPSAR, який працює в діапазоні частот від 400 МГц до 4 ГГц. [10]

Отримані системою дані обробляються використовуючи різну геометрію цільдатчика. Одночасно з рухом платформи система формує 2D або 3D зображення, когерентно їх об’єднуючи, такий метод називається Multi-lookprocessing, цей метод буде також використано при обробці зображень системою описаною в наступному розділі. Завдяки такому підходу отримані зображення стають чіткішими, адже зменшується шум.

Рис. 1.18. Газонний транспортний засіб зі встановленим FLGPSAR

Систему випробовували як на металевих так і на пластикових мінах, які були закопані на різній глибині. З металевими мінамине виникає жодних проблем, наприклад для проведення тестування системи було використано протитанкову міну ТМ62, закопану на глибині 5см. Отримані зображення охоплювали площу 4,5 м на 5 м, на відстані 4 м від транспортного засобу. В той же час пластикові міни виявлялися системою важче, цьому сприяло те, що такий тип мін має низький дієлектричний контакт з ґрунтом.

Основна перевага цієї системи це здатність виявляти міни спрямованого типу, такі як МОН-100 та інші. Зазвичай ці міни розміщуються біля дороги або стежкои, приховані за густою рослинністю, і представляють серйозну небезпеку для автомобілів та особового складу. Активація мін відбувається за командою або через розтяжку. Для перевірки ефективності системи FLGPSAR проти мін спрямованого типу було створено муляж такої міни.На інженерному об'єкті PSI у Лонг-Біч, штат

Міссісіпі змоделювали ситуацію де виготовлений муляж, металева пластина, приблизно такого ж розміру, як MON-100, була закріплена на дереві за 10 метрів від траєкторії FLGPSAR та за 1 метр від густої рослинності. Цю пластину не було можливо візуально виявити. Проте, система чудово впроралася з поставленою задачею та без проблем виявила імпровізовану міну.

1.7.3. Система виявлення мін розтяжок з використанням радара з синтезованою апертурою

Система розроблена студентами Маркусом Шартелем, Ральфом Буром з Інституту мікрохвильової інженерії Університету Ульма, і їхніми керівниками, Вінфрідом Майєром та Крістіаном Вальдшмідтом, призначена для виявлення протипіхотних мін-розтяжок PROM-1. Їхній виріб це БПЛА зі встановленим на нього РСА з бістатичним частотним-модулятором постійної хвилі (FMCW SAR), що працює в діапазоні від 1 до 4 ГГц. Антени радара можуть бути повернуті вручну на 360°. Радар може знаходитись у повітрі до 23 хвилин. Отримані радаром данні зберігаються на одноплатному комп’ютері для подальшої обробки після приземлення. [11]

Систему було протестовано з метою виявлення муляжу міни PROM-1, яку було встановлено у вкритій росою траві (Рис. 1.19). Через траву, натягнуті розтяжки погано видно на готовому зображенні (Рис. 1.20). Для поліпшення чіткості зображення, було вирішено зробити більше зображень сцени та об'єднати їх в одне, такий метод називається Multi-look processing.Отриманий результат можна побачити на рис. 1.21.

Рис. 1.19. Встановлений в траві муляж міни та 4 розтяжки

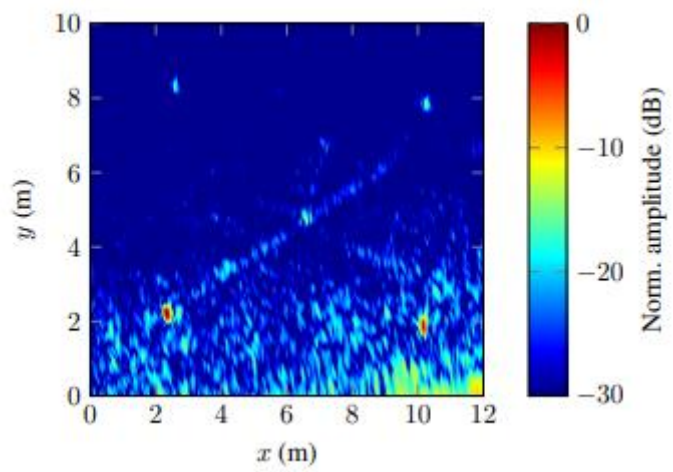

Рис. 1.20. Отримане зображення

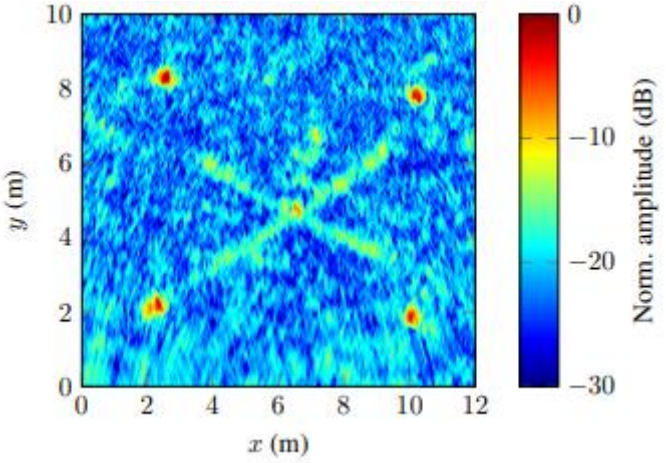

Рис. 1.21. Зображення після використання методу Multi-look processing

Отже, підсумовуючи перший розділ, можна сказати що, радари з синтезованою апертурою є потужним інструментом не тільки для отримання високоякісних зображень з великої відстані, а і для безпечного виявлення вибухонебезпечних предметів. Вони мають різноманітні параметри, які можуть бути адаптовані під конкретні завдання. Аналіз та класифікація цих параметрів допомагає розуміти особливості роботи радарів з синтезованою апертурою та вибирати найкращі рішення для конкретних застосувань.

РОЗДІЛ 2
ІНТЕЛЕКТУАЛЬНА ОБРОБКА ЗОБРАЖЕНЬ. ТОПОЛОГІЯ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

2.1. Огляд методів інтелектуальної обробки зображень

Сьогодні штучний інтелект як ніколи розвивається, що розширює межі його застосування. Штучний інтелект застосовується для розпізнавання голосу, генерації текстів, як інструмант навчання, за допомогою якого ми можемо за лічені секунди отримати потрібну нам інформацію, але однією з основних його галузей застосування на маю думку є інтелектуальна обробка зображень.

Інтелектуальна обробка зображень — це галузь науки, що займається розробкою та застосуванням алгоритмів та методів для автоматичної обробки та аналізу зображень з використанням штучного інтелекту та машинного навчання. Завдяки поєднанню передових технологій комп'ютерного зору, обробки сигналів, статистичного аналізу та нейронних мереж, інтелектуальна обробка зображень стає важливим інструментом у різних галузях, від медицини та робототехніки до автоматизації виробництва та досліджень в галузі науки.

На рис. 2.1 наведено основні типи задач які вирішуються алгоритмами комп’ютерного зору. Для вирішення цих задач застосовують згорткові нейронні мережі (Convolutional neural network).

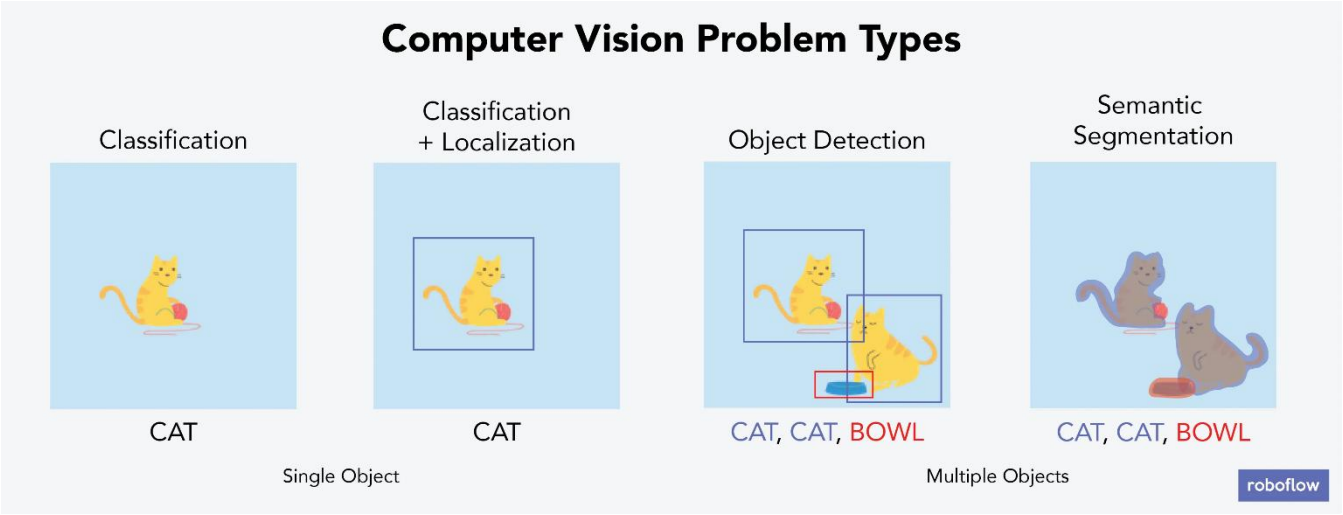

Рис. 2.1. Основні типи задач комп’ютерного зору

Базовою задачею алгоритмів комп’ютерного зору, є класифікація. Класифікація (Classification) – це процес віднесення об'єктів або регіонів зображення до певних категорій чи класів на основі їх характеристик, які визначаються попередньо встановленими критеріями. По суті, це завдання призначення тегів чи міток до окремих областей зображення з метою їх ідентифікації або класифікації. Застосовуючи методи машинного навчання, наприклад, нейронні мережі, можна створити моделі, які автоматично визначають типи об'єктів на зображеннях. Це дозволяє автоматизувати процеси аналізу великих обсягів зображень та швидше отримувати інформацію з даних. Класифікація також може додатково включати в себе локалізацію класифікованого об’єкта. Це означає визначення положення об’єкта на зображенні, та виділенні його рамкою.

Виявлення об'єктів (Object Detection) - це один з ключових напрямків в комп'ютерному зорі, який включає виявлення і класифікацію об'єктів на зображенні або в відео. На відміну від задачі класифікації, де система визначає лише наявність певного класу об'єктів на зображенні, об'єктне виявлення також визначає координати рамки навколо кожного знайденого об'єкта. Це дозволяє не лише ідентифікувати об'єкти, але й точно визначити їх місцезнаходження.

Однією з основних задач інтелектуальної обробки зображень є автоматичне визначення об'єктів та паттернів на зображеннях. Це включає в себе сегментацію зображень, яка полягає в розділенні зображення на окремі області або об'єкти з метою подальшого аналізу.

Сегментація - це процес розділення зображення на окремі сегменти або регіони, які мають спільні характеристики або властивості. Ці сегменти можуть представляти об'єкти, структури або області зображення з певними властивостями, такими як текстура, колір, яскравість тощо. Сегментація може бути використана для розпізнавання об'єктів, виявлення аномалій, а також для підготовки даних для інших задач, наприклад, для класифікації об'єктів на зображенні. У випадку з зображеннями SAR, сегментація може використовуватися для виділення різних типів місцевості або об'єктів на земній поверхні, таких як вода, ліси, забудови, дороги тощо. Це дозволяє аналізувати інформацію, яка міститься на зображенні, в

більш зрозумілому та корисному форматі. Наприклад, сегментація може допомогти в ідентифікації зон з підвищеним ризиком природних лих або в аналізі змін у природних довкіллях.

2.2. Згорткові нейронні мережі

Згорткова нейронна мережа (Convolutional neural network) – це нейронна мережа, основна задача якої, розпізнавання та аналіз образів на зображеннях. CNN використовує операцію згортки для виділення опорних ознак зображення, таких як контури, ребра, грані. Іншими словами ця мережа виділяє характерні ознаки об’єкта для подальшого застосування, це працює подібно до зорової кори головного мозку, коли ви дивитесь на якийсь об’єкт, наприклад комп’ютер, ваш мозок фокусується на його ключових ознаках, порівнюючи їх з комп’ютерами баченими раніше.

На сьогоднішній день існує безліч архітектур CNN для різних застосувань. Наприклад для класифікації зображень використовують архітектури LeNet-5, AlexNet, VGGNet та інші. Для виявлення та сегментації об’єктів використовують Faster R-CNN, YOLO, Mask R-CNN.

2.2.1. Структура згорткової нейронної мережі

Типову стурктуру CNN показано на рис. 2.2, вона містить в собі вхідний шар, додаткові шари у вигляді фільтрів та згорток та вихідний шар.

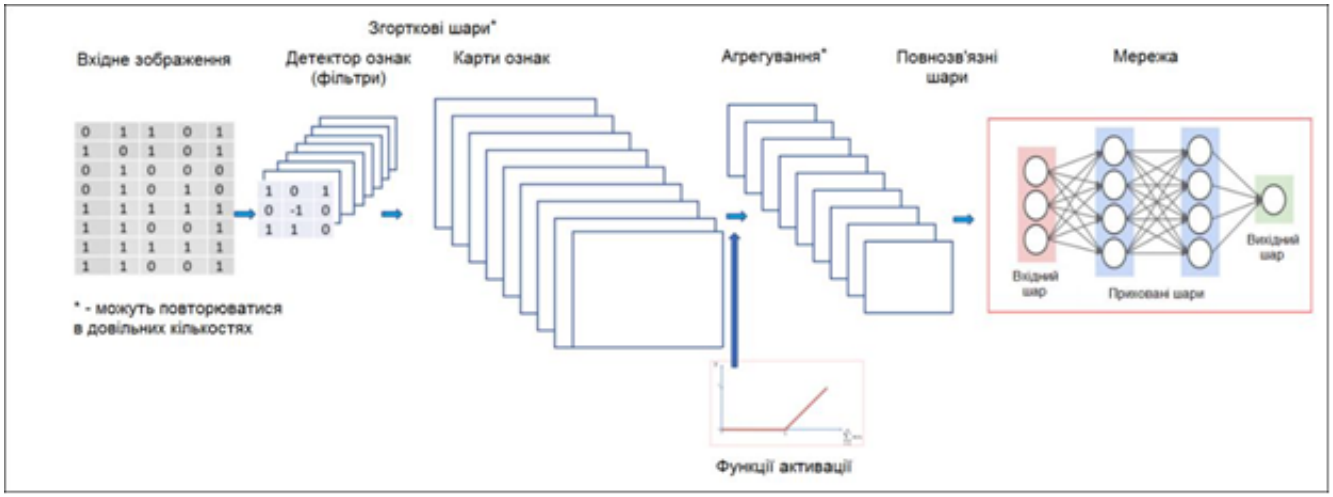

Рис. 2.2. Загальна структура згорткової нейронної мережі

Детально розглянемо структуру CNN зображену на рис. 2.2. Першим етапом обробки є представлення вхідного зображення у вигляді матриці чисел де кожне число відповідає своєму пікселю на зображенні. У кольорових зображеннях числа позначають комбінацію значень трьох основних кольорів, червоного, зеленого та синього (RGB). У чорно-білому зображенні, кожен піксель представлений значенням інтенсивності світла, 0 – чорний, 255 – білий.

Згорткові шари містять в собі певний набір фільтрів або детекторів ознак, які можна розглядати як двовимірні числові матриці. Згорткові шари займають провідну роль у структурі CNN, адже виконують операцію згортки розглянуту у розділі 3.3.2.

Фільтри (детектори, ядра) – представлені у вигляді квадратних матриць непарної довжини(наприклад, 3x3 або 5x5) вони проходять по всьому зображенню, зберігаючи просторову інформацію. На кожному кроці вони обчислюють згортку з частиною зображення, генеруючи карту ознак. Ці карти ознак підкреслюють різні аспекти зображення, такі як краї, текстури або інші важливі ознаки.

Агрегування – операція зменшення карт ознак, зберігаючи при цьому важливу інформацію. Існують два підходи до агрегування максимальне агрегування (Max Pooling) та Середнє агрегування (Average Pooling). Max Pooling є найпоширенішим, основна ідея цього методу полягає у розділенні карти ознак на невеликі області (наприклад, 2x2) та виборі максимальних значеннь в кожній області. Такий підхід дозволяє зменшити розмір карти ознак, зберігаючи найважливіші ознаки. Average Pooling так само розділяє карту ознак на невеликі області, але замість вибору найбільшого значення, обчислює середнє значення для кожної області. Основним недоліком такого методу є неефективне виділення ключових ознак, саме через це метод не здобув широкого використання.

Функція активації – застосовується після кожної операції згортки, робить модель нелінійною, що дозволяє НМ моделювати більш складні дані. Найпопулярнішою функцією активації у CNN є ReLU (Rectified Linear Unit). Вона замінює всі від'ємні значення в карті ознак на нулі, залишаючи всі додатні значення

незмінними. Це допомагає моделі бути більш стабільною і уникати проблеми зникаючих градієнтів.

Після кількох шарів згорток та агрегування, CNN зазвичай має один або більше повнозв'язаних шарів. Ці шари працюють аналогічно нейронам у традиційних нейронних мережах. Вхідний шар - риймає вхідні дані у вигляді векторів, створених з карт ознак після попередніх етапів обробки. Вхідні дані можуть бути підготовлені шляхом розгортання (флатенінгу) карт ознак у один довгий вектор. Приховані шари - складаються з набору нейронів, кожен з яких з'єднаний з нейронами попереднього шару. Кожне з'єднання має свою вагу, яка коригується під час навчання. Вихідний шар генерує остаточний результат класифікації або регресії. Для задач класифікації часто використовується функція активації softmax, яка перетворює вихідні значення на ймовірності, що сумуються до 1.

Зключним етапом є начання мережі, це ключає в себе оптимізацію ваг усіх з'єднань у мережі, щоб мінімізувати помилку на тренувальному наборі даних. Це досягається шляхом зворотного поширення помилки (backpropagation) і використання алгоритмів оптимізації, таких як стохастичний градієнтний спуск (SGD) або його варіації (наприклад, Adam).

2.2.2. Операція згортки

Операція згортки – це основна складова будь якої ЗНМ, згортка дозволяє ідентифікувати важливі ознаки вхідного зображення, це можуть бути такі ознаки, як контури, текстури, форми об'єктів та інші візуальні характеристики. Використовуючи згортку, мережа може ефективно зменшити розмірність вхідних даних, зберігаючи при цьому важливу інформацію, і тим самим зменшити кількість параметрів моделі, що робить її більш ефективною та менш схильною до перенавчання.

Для операції згортки нам потрібно визначити фільтр (ядро), як вже було сказано в розділі 3.3.1, ядро це матриця ваг. Ці ядра можуть бути різного розміру (наприклад, розміром 3x3, 5x5 або 7x7) і використовуватися для виявлення різних

аспектів вхідних даних. Наприклад, перші шари мережі можуть мати ядра для виявлення простих ознак, таких як горизонтальні та вертикальні границі, тоді як подальші шари можуть мати ядра для виявлення більш складних ознак, таких як форми або об’єкти. Для виявлення країв використовують ядро собеля:

$$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

Вертикальний фільтр Собеля виявляє вертикальні контури зображення, а горизонтальний фільтр Собеля виявляє горизонтальні контури. Для згладжування, зменшення шуму зображення, використовують гаусове ядро:

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Кожен елемент ядра має своє вагове значення, яке оновлюється під час навчання моделі. У CNN кожен шар має свій набір ядер, які навчаються під час процесу навчання мережі.

Ядро поступово проходить по всьому зображенню, на кожному кроці накладаючись на частину зображення. Елементи ядра множаться на відповідні пікселі зображення. Отриманий скалярний добуток цих значень підсумовується, утворюючи одне значення на вихідній карті ознак. Цей процес повторюється для кожної позиції ядра на зображенні.

Важливим параметром операції згортки є крок згортки, він визначає на скільки пікселів зсувається ядро на кожному кроці згортки. Збільшення кроку зменшує розмірність вихідної карти ознак, оскільки зображення проходиться швидше.

Іншим важливим параметром є відступ(padding), вінозначає додавання певних значень (найчастіше нулів) навколо країв вхідного зображення перед застосуванням операції згортки. Це робиться для того, щоб контролювати розмір вихідних карт ознак та забезпечити коректне оброблення пікселів на краях зображення. Наприклад якщо ми хочемо, щоб розмір вихідної карти ознак залишався таким же, як і вхідне зображення, необхідно застосувати same padding, додавання додаткових рядів і

колонок нулів навколо країв зображення. Формула для визначення відступу з кожного боку:

$$P = \frac{K - 1}{2}$$

Де K – розмір ядра.

Значення отримані в результаті згортки проходять через функцію активації, наприклад, ReLU (Rectified Linear Unit), яка замінює всі від'ємні значення нулями. Це допомагає моделі вловлювати нелінійні взаємозв'язки у даних.

2.3. Навчання згорткових нейронних мереж

Навчання нейромережі – це процес налаштування параметрів (ваг та зміщень) нейронної мережі для того, щоб вона могла виконувати конкретне завдання, таке як класифікація, регресія, сегментація тощо. Процес навчання включає кілька ключових етапів і методів, які допомагають нейромережі вчитися з даних, розпізнавати патерни та робити точні передбачення.

Для навчання нейромережі, застосовують заздалегідь підготовлений набір даних – датасет. Датасет (Dataset) — це структурована колекція даних, яка використовується для навчання, тестування та оцінки моделей машинного навчання, зокрема нейронних мереж. В контексті CNN таким набором даних є колекція зображенн, наприклад одним з популярних датасетів для задачі класифікації зображень є CIFAR-10, CIFAR-10, який містить 60 000 кольорових зображень розміром 32x32 пікселів, розподілених на 10 класів (по 6 000 зображень на кожен клас). Інший приклад – датасет MNIST, який містить 70 000 зображень рукописних цифр від 0 до 9. Зазвичай для навчання один датасет розділяють на 3 типи:

- Навчальний датасет використовується для навчання моделі. Це основна частина даних, з якої модель вчиться розпізнавати структури та патерни.

- Валідаційний датасет використовується для налаштування гіперпараметрів моделі та перевірки її продуктивності під час навчання. Це допомагає уникнути перенавчання.

- Тестовий датасет використовується для остаточної оцінки продуктивності моделі після завершення навчання. Це дає змогу оцінити, наскільки добре модель узагальнює нові, невідомі дані.

Важливим параметром навчання є епоха. Епоха — це один повний прохід через увесь навчальний датасет. Під час епохи всі приклади з навчального датасету передаються через модель один раз для обчислення та оновлення ваг мережі. Кількість епох визначається залежно від складності задачі та розміру датасету. Наприклад, великі датасети потребують більше епох, адже модель має засвоїти всі можливі патерни.

Оскільки обробка всього датасету за раз може бути обчислювально затратною, датасет розбивається на менші підмножини, звані батчами (batches). Кожен батч обробляється окремо під час однієї ітерації. Наприклад, якщо датасет містить 1000 зразків, а розмір батчу становить 100, то одна епоха буде складатися з 10 ітерацій. Після кожної ітерації ваги моделі оновлюються на основі градієнтів, обчислених для поточного батчу. Це дозволяє моделі поступово покращувати свої передбачення, зменшуючи функцію втрат. Процес повторюється протягом декількох епох.

2.4. CNN архітектури для задач обробки зображень

В цьому розділі мова піде про архітектури згорткових мереж для класифікації та сегментації. Класифікація зображень передбачає віднесення зображення до однієї з декількох категорій. CNN досягають цього шляхом автоматичного виділення та навчання ознак, що характеризують кожну категорію. Існуючи на сьогднішні день архітектури задовольняють буктично будь які потреби.

Архітектура CNN для задач класифікації зазвичай повторює архітектуру описану у розділі 3.3.1, за вийнятком застосування додаткового шару Softmax, та різних невеликих відмінностей. Шар Softmax використовується для генерації ймовірностей приналежності до кожної категорії.

LeNet — одна з перших архітектур CNN, розроблена для класифікації рукописних цифр (наприклад, у базі даних MNIST). Вона включає кілька згорткових

і пулінгових шарів, за якими слідують повнозв’язні шари. LeNet показала, що CNN можуть успішно вирішувати задачі класифікації зображень.

AlexNet — більш складна архітектура CNN, яка виграла конкурс ImageNet у 2012 році. Вона складається з п’яти згорткових шарів з шаром ReLU після кожного, трьох шарів пулінгу та трьох повнозв’язних шарів. Вона продемонструвала значне покращення в порівнянні з попередніми методами, завдяки використанню глибоких згорткових шарів та методів регуляризації, таких як dropout.

VGGNet відрізняється використанням послідовних згорткових шарів з невеликими ядрами (3x3) та глибокою архітектурою. Це дозволяє моделі виявляти складні ознаки на різних рівнях абстракції. VGGNet показала, що збільшення глибини мережі може значно покращити точність класифікації.

ResNet вводить ідею залишкових (residual) зв'язків, які дозволяють створювати дуже глибокі мережі без проблем з градієнтним зниканням. Це дозволило створювати моделі з сотнями шарів, значно підвищуючи їхню точність.

Для задач сегментації найчастіше використовують U-Net та Fully Convolutional Networks, через їхню простоту та широкі можливості застосування. Кожна з цих архітектур має свої особливості, розуміння цих особливостей допоможе нам визначити яка архітектура краще підійде для виконання нашого завдання.

U-Net є однією з найпопулярніших архітектур для задач сегментації зображень. Вона складається з двох частин: контрактної (contracting path) та експансивної (expansive path). Контрактна частина складається з послідовних згорткових і пулінгових шарів, які зменшують розмірність і витягують ознаки. Експансивна частина включає деконволюційні шари, які відновлюють розмірність, та пропускні зв'язки, які передають деталі від контрактної частини до відповідних деконволюційних шарів. Це дозволяє зберігати просторову інформацію та покращує якість сегментації.

FCN — це архітектура, яка використовує тільки згорткові шари, замість повнозв’язних шарів. Це дозволяє обробляти зображення будь-якого розміру і повертати карти ознак тієї ж розмірності, що і вхідне зображення. FCN також

використовують деконволюційні шари для відновлення розмірності зображення та пропускні зв'язки для збереження деталізованої інформації.

Досить популярною на сьогоднішній день, є YOLO архітектура. YOLO (You Only Look Once) – призначена для виконання великого спектру операцій над зображенням, класифікація, сегментація, детекція об’єктів. Завдяки своїй простоті у використанні, ця архітектура часто застосовується для визначення типу об’єкту на зображенні чи відео.

РОЗДІЛ 3
АРХІТЕКТУРА YOLO. ІНТЕЛЕКТУАЛЬНА ОБРОБКА ЗОБРАЖЕНЬ ОТРАМАНИХ РАДАРОМ З СИНТЕЗОВАНОЮ АПЕРТУРОЮ

3.1. Обґрунтування вибору архітектури YOLO

Коли стоїть задача обробки будь якого зображення, то при виборі архітектури НМ для цієї задачі ми впершу чергу звертаємо увагу на такі показники як:

- швидкодія;
- простота використання;
- наявність детальної документації яка описує модель;
- підтримка розробниками свого продукту.

YOLO відповідає всім цим критеріям, забезпечуючи розробнику безкоштовний доступ до своїх продуктів, та головне це те, що YOLO надає рішення для виконання задач обробки різного типу. Особливість цієї моделі це можливість виконання задачі за один прохід вхідного зображення через НМ, це суттєво скорочує час необхідний для обробки однієї фотографії, також це робить можливим використання НМ в режимі реального часу. Додатковим аргументом на вибір саме YOLO, Є відкритість архітектури та програмного коду.

В практичній частині цієї роботи мною було прийнято рішення використовувати модель YOLOv8, хоча це не найновіша модель, вона досить точна та ефективна для простих задач сегментації, класифікації, детекції.

YOLOv8 є універсальною моделлю, яка може бути використана для різних задач детектування об'єктів. Її гнучкість та можливість адаптації до різних

масштабів об'єктів дозволяють використовувати модель у різноманітних областях, від медицини до розваг. Покращені методи екстракції ознак та мультискалярне навчання дозволяють YOLOv8 досягати високих показників точності. Це робить модель здатною точно визначати об'єкти навіть у складних сценах з великою кількістю деталей та різними умовами освітлення.

Незважаючи на оптимізовану архітектуру, навчання YOLOv8 все ще потребує значних обчислювальних ресурсів, особливо для навчання на великих наборах даних. Використання потужних графічних процесорів (GPU) та розподілених систем обробки може допомогти вирішити цю проблему.

Як і більшість сучасних моделей глибокого навчання, YOLOv8 потребує великої кількості даних для ефективного навчання. Збирання та анотування таких даних може бути трудомістким та дорогим процесом. Однак, використання методів transfer learning та data augmentation може значно знизити ці витрати.

YOLOv8 здатна виконувати різноманітні задачі, такі як класифікація, детекція, сегментація, трекінг, та розпізнавання рухів і поз (Рис. 3.1). [12]

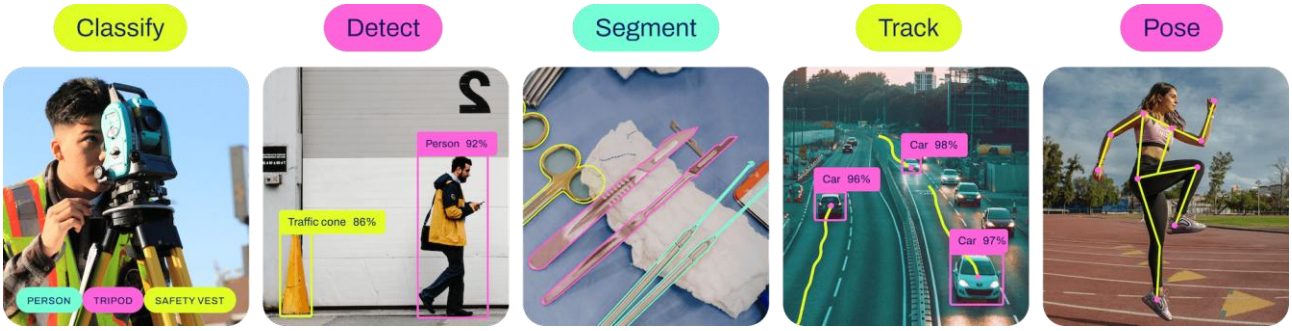

Рис. 3.1. Задачі які виконує YOLOv8

Щоб оцінити цей вибір можна порівняти цю модель з іншими моделями YOLO. На рис. 3.2 можна побачити два графіки, на них представлена порівняльна оцінка продуктивності різних моделей для обробки зображень за метрикою COCO AP (Average Precision), яка використовується в задачах розпізнавання об'єктів. На лівому графіку показана залежність COCO AP від затримки (latency), а на правому — від кількості параметрів моделі (number of parameters).

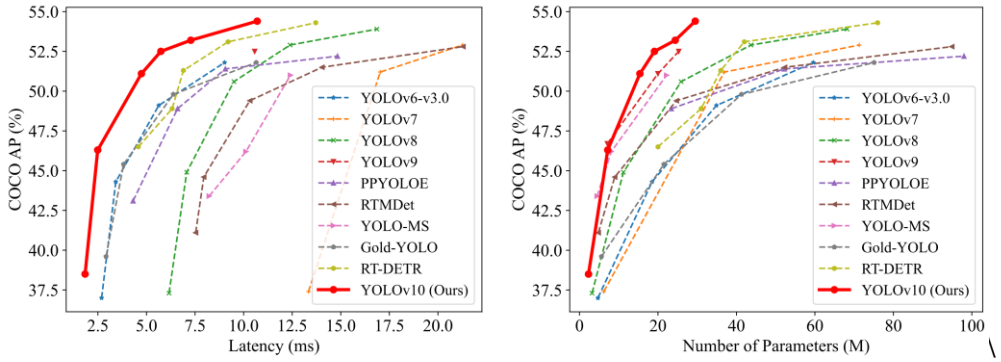

Рис. 3.2. Порівняльна оцінка продуктивності різних моделей YOLO для обробки зображень за метрикою COCO AP

Показані на графіках моделі за багатьма пареметрами поступаються новій YOLOv10, але YOLOv8 все ще показує високі результати, особливо при низьких значеннях затримки.

За ідеальних обставин, звісно варто було б використовувати останню модель серії YOLO, але функціоналу та можливостей v8 достатньо для виконання задачі цієї роботи.

3.2. Аналіз архітектури YOLOv8

YOLOv8 продовжує концепцію одностадійного об'єктного детектування, де зображення розбивається на сітку, і кожна сітка відповідає за передбачення об'єктів у межах своєї області. Ключовими цілями розробки YOLOv8 були підвищення точності детектування та зниження затримки обробки, що робить модель ідеальною для реальних застосувань, таких як системи відеоспостереження, автономні транспортні засоби та інші області, де критично важливими є як швидкість, так і точність.

Архітектура цієї мережі базується на попередніх версіях YOLO, і має парамідальну структуру. Використовуючи модифіковану версію архітектури CSPDarknet53 як основу та впроваджуючи передові технології, такі як механізм самоуваги, YOLOv8 обіцяє ще кращу точність і ефективність у порівнянні зі своїми попередниками. [13]

Однією з ключових інновацій в YOLOv8 є використання механізму self-attention в головній частині мережі. self-attention є технікою, яка дозволяє моделі

фокусуватися на різних частинах зображення, виділяючи найбільш важливі області для поточного завдання. Це особливо корисно для об'єктного детектування, де важливо не лише розпізнати об'єкт, але й правильно визначити його межі та категорію.

Механізм self-attention працює шляхом обчислення ваг для кожної ознаки на зображенні. Ці ваги визначають, наскільки важливою є кожна ознака для поточного завдання. В результаті модель може коригувати важливість різних ознак, підвищуючи точність і ефективність виявлення об'єктів.

На рис. 3.3, показано основні складові цієї мережі та її детальну структуру. Цю структуру поділено на дві частини:

- Базова мережа (Backbone);
- Мережа виведення (Head);

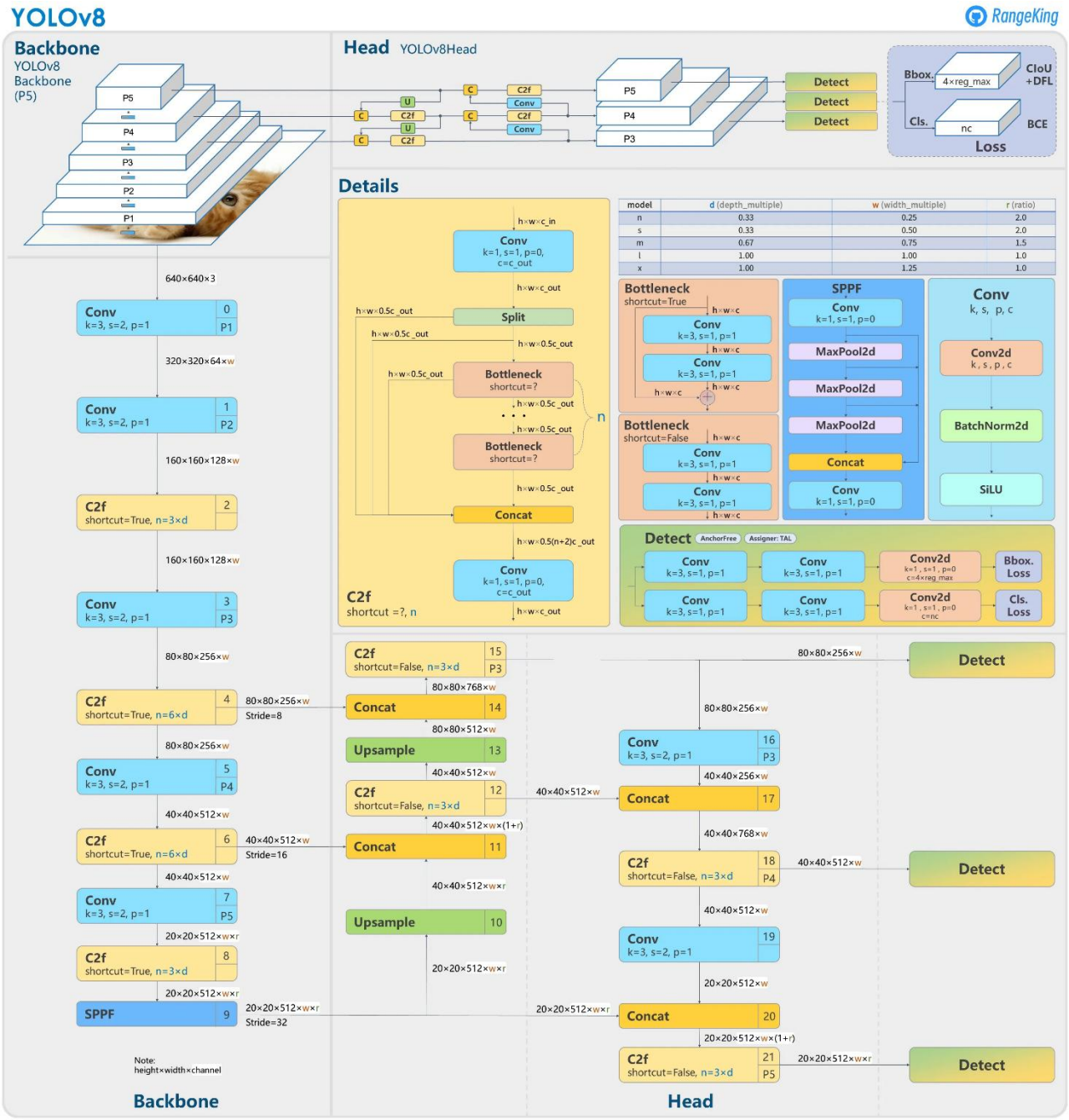

Рис. 3.3. Архітектура YOLOv8

Базова мережа у YOLOv8 відповідає за екстракцію ознак із вхідного зображення. Для цього використовуються глибокі нейронні мережі, які можуть варіюватися в залежності від конкретної реалізації. Основними критеріями вибору базової мережі є її здатність ефективно витягувати інформативні ознаки та забезпечувати високу продуктивність. У YOLOv8 використовується варіація ResNet або CSPDarknet, які добре зарекомендували себе в попередніх версіях YOLO.

Базова мережа складається з кількох рівнів (P1, P2, P3, P4, P5), які відповідають за обробку зображення на різних масштабах. Кожен рівень містить різні типи шарів, такі як конволюційні шари (Conv), шари з об'єднанням ознак (Concat), шари з використанням стрибків (Shortcut), та спеціалізовані блоки, такі як C2f та SPPF. На кожному рівні Backbone використовуються конволюційні шари з різними параметрами:

- Перший рівень (P1) починається з конволюційного шару з ядром розміром 3x3, stride 2 і 64 вихідними каналами. Цей шар застосовується для первинної обробки вхідного зображення розміром 640x640.

- Другий рівень (P2) також включає конволюційний шар з аналогічними параметрами, але зменшує розмір вхідного зображення до 320x320.

- На третьому рівні (P3) зображення ще більше зменшується до 160x160 за допомогою конволюційного шару з аналогічними параметрами.

Блоки C2f (Cross Stage Partial Networks) використовуються для ефективного об'єднання ознак з різних шарів. Наприклад, після конволюційного шару на P2 слідує блок C2f з трьома "bottleneck" шарами з короткими з'єднаннями (shortcut), вони використовуються для збереження інформації та покращення збіжності моделі. Кожен bottleneck шар включає кілька конволюційних шарів з різними параметрами. На рівні P3 також є блок C2f, але він має шість "bottleneck" модулів.

Блок SPPF використовується на останньому рівні Backbone (P5). Він складається з кількох шарів MaxPooling та конволюційних шарів, які допомагають зберегти важливі просторові ознаки на різних масштабах. Цей блок дозволяє моделі враховувати контекстну інформацію та значно покращує продуктивність на великих масштабах.

Мережа виведення (Head) у YOLOv8 складається з кількох рівнів детектування, які використовуються для передбачення об'єктів різних масштабів (Рівні P3-P5). Модуль детектування на рівні P3 включає кілька шарів, таких як Conv, C2f та Upsample, що дозволяють моделі здійснювати точні передбачення для дрібних об'єктів. Аналогічно, модуль детектування на рівні P4 використовує комбінацію шарів Conv, Concat та C2f для передбачення об'єктів середнього розміру.

Останній модуль детектування на рівні P5 обробляє великі об'єкти, використовуючи шари Conv, Concat та C2f. Concat шари використовуються для об'єднання вихідних ознак з різних рівнів або модулів, що дозволяє моделі ефективно використовувати інформацію з різних масштабів. Upsample шари застосовуються для збільшення розмірності ознак, що дозволяє моделі здійснювати точні передбачення для дрібних об'єктів. Ці шари використовуються в комбінації з Concat шарами для об'єднання інформації з різних рівнів.

Модулі детектування відповідають за передбачення меж об'єктів (bounding boxes), класів об'єктів та ймовірностей цих передбачень. Втрати (Loss) обчислюються за допомогою кількох функцій втрат, таких як BCE (Binary Cross Entropy) для класифікації та CIoU/DIoU для регресії меж об'єктів.

BatchNorm2d Використовується для нормалізації вихідних ознак, що сприяє стабільному навчанню моделі.

Активаційна функція SiLU (Swish) допомагає моделі краще узагальнювати інформацію та покращує збіжність.

Архітектура YOLOv8 є складною та багатошаровою системою, що поєднує в собі різні інноваційні підходи до обробки зображень. Її компоненти, такі як Conv, C2f, Bottleneck, Concat, Upsample та SPPF блоки, працюють разом для забезпечення високої точності та швидкості детектування. Завдяки цьому YOLOv8 є потужним інструментом для різних задач комп'ютерного зору, від відеоспостереження до автономних транспортних систем.

3.3. Інструменти для розробки програми

Важливим пунктом виконання цієї роботи є вибір інструментів для виконання завдання. Під інструментами мається на увазі мова програмування та необхідні бібліотеки також необхідно визначити середовище розробки. Мною були вибрані наступні засоби розробки:

- мова програмування Python;
- бібліотека ultralytics для інтеграції вибраної моделі YOLO;
- бібліотека PIL для роботи з зображеннями;

- бібліотека PyQt6 для створення інтерфейсу користувача;

- середовище розробки PyCharm.

В якості мови програмування, було вирішено обрати Python. Python – це мова програмування високого рівня, для написання різноманітних за призначенням програм. Створена у 1991 році Гвідо ван Россумом, Python стала однією з найпопулярніших мов завдяки своїй простоті, читабельності та багатофункціональності. Python підтримує кілька парадигм програмування, включаючи об'єктно-орієнтоване, процедурне та функціональне програмування. Це дає розробникам гнучкість у виборі підходу до вирішення конкретних завдань. Python вирізняється серед інших мов своєю простотою. Код на Python часто виглядає як псевдокод, що робить його дуже легким для читання. Завдяки своєму простому та зрозумілому синтаксису, Python є однією з найкращих мов для початківців. Багато навчальних закладів обирають Python як першу мову програмування для своїх студентів. Крім того, існує безліч онлайн-курсів, підручників та відеоуроків, що допомагають швидко освоїти основи мови. [14]

Велика і активна спільнота Python — одна з його головних переваг. Безліч ресурсів, таких як форуми, групи в соціальних мережах, конференції та навчальні матеріали, доступні для розробників будь-якого рівня. Спільнота також активно працює над розвитком мови та створенням нових бібліотек і фреймворків.

В першу чергу, головним аргументом на користь вибору Python була наявність великої кількості бібліотек, які розширюють можливості цієї мови, дозволяючи виконовати безліч функції серед яких є машинне навчання та обробка зображень. Розглянемо використані в роботі бібліотеки.

Ultralytics — це відома бібліотека Python, яка спеціалізується на реалізації алгоритмів комп'ютерного зору, зокрема для об'єктного виявлення та сегментації зображень. Вона отримала широку популярність завдяки своєму флагманському проекту — YOLO (You Only Look Once), який є одним з найефективніших та найшвидших алгоритмів для об'єктного виявлення. Ultralytics підтримує різні архітектури моделей, що дозволяє користувачам обирати ту, яка найкраще

відповідає їхнім потребам. Це може бути корисно, наприклад, для балансування між точністю та швидкістю виконання [15].

Однією з головних переваг Ultralytics є її простота та інтуїтивність у використанні. Бібліотека пропонує зручний API, який дозволяє швидко інтегрувати моделі комп'ютерного зору в різноманітні проекти. Крім того, детальна документація та численні приклади допомагають розробникам швидко освоїти основи та розпочати роботу.

PIL (Python Imaging Library) — це популярна бібліотека для роботи з зображеннями в Python. Вона надає багатий набір інструментів для обробки та маніпуляції зображеннями. Хоча оригінальна бібліотека PIL більше не підтримується, її форк під назвою Pillow є сучасною альтернативою, яка активно розвивається та підтримується спільнотою. PIL (Python Imaging Library) — це популярна бібліотека для роботи з зображеннями в Python. Вона надає багатий набір інструментів для обробки та маніпуляції зображеннями. Хоча оригінальна бібліотека PIL більше не підтримується, її форк під назвою Pillow є сучасною альтернативою, яка активно розвивається та підтримується спільнотою.

PyQt6 — це бібліотека на Python для створення графічних інтерфейсів (GUI), яка поєднує Python з Qt. Вона надає широкий набір віджетів та функціональних можливостей для створення кросплатформних додатків. PyQt6 підтримує сучасні стандарти, включаючи Qt Quick, для розробки інтерактивних інтерфейсів. Бібліотека легко інтегрується з іншими фреймворками Python, що робить її придатною для складних проєктів [16].

Також використано такі бібліотеки, як os, glob, pylabel, shutil, yaml але вони не відіграють основної ролі у роботі застосунку, та являються скоріше допоміжними у процесі підготовки набору даних до навчання. У розділах 3.3.2. та 3.3.3. розкрито функціонал цих бібліотек, який було використано.

Вибираючи середовище розробки (IDE), основною вимогою, було забезпечення комфортного програмування, та налагодження написаної програми. Існує досить багато хороших IDE для мови програмування Python, але на мою думку саме PyCharm є ідеальним рішенням.

PyCharm — це інтегроване середовище розробки (IDE) для мови програмування Python, розроблене компанією JetBrains. PyCharm є одним з найпопулярніших і найпотужніших інструментів для розробки на Python, надаючи безліч функцій для підвищення продуктивності розробників [17]. PyCharm надає потужний редактор коду з підтримкою підсвічування синтаксису, автозаповнення коду, рефакторингу та навігації по коду. Редактор також підтримує перевірку помилок у реальному часі та підказки, що значно полегшує написання коду. Також він включає інтегрований налагоджувач, який дозволяє розробникам встановлювати точки зупинки, виконувати код крок за кроком, аналізувати змінні та виконувати вирази на льоту. Це робить процес відладки коду значно ефективнішим та сприяє написанню якісного, надійного та безпомилкового коду. PyCharm підтримує великий набір плагінів, які можна встановлювати для розширення функціональності IDE. Це дозволяє налаштовувати PyCharm під конкретні потреби та вподобання кожного розробника. Завдяки своїм численним інструментам та функціям, таким як автозаповнення коду, рефакторинг, перевірка помилок у реальному часі та інші, значно підвищується продуктивність розробників. Це дозволяє зосередитися на написанні коду, не відволікаючись на рутинні завдання. Варто також зазначити підтримку фреймворків для тестування та вбудовану систему контролю версій, але в цій роботі цей функціонал не застосовується.

3.4. Розробка програми

Мета програми навчити готову модель розпізнавати кораблі на зображеннях зроблених SAR. Вихідний код програми надано в додатках А, Б, В, Г. Далі детально описано роботу над кожною частиною проєкту.

3.4.1 Тренувальна вибірка

Для навчання мережі було вибрано датасет “SARscope: Synthetic Aperture Radar Maritime Images” [18]. Цей датасет являє собою набір зображень 640 на 640 пікселів, у форматі “.jpg”, зроблених радаром з синтезованою апертурою, та

використовується для досліджень у сфері автоматичного розпізнавання об'єктів на морській поверхні.

Цей датасет розділений на три набори:

- тестовий набір, містить в собі 672 зображення;
- навчальний набір, містить в собі 4717 зображень;
- валідаційний набір, містить в собі 1346 зображень.

3.4.2. Процес підготовки даних для тренування моделі

Перед тим як приступити до тренування моделі та написання застосунку, необхідно провести певні маніпуляції з датасетом. Перишим кроком необхідно створити директорії для тренувального, тестового та валідаційного наборів даних. Далі необхідно скопіювати зображення з вихідних директорій до відповідних нових директорій. Кінцевим кроком є конвертування анотацій з формату COCO у формат YOLO для кожного набору даних.

За виконання цих задач відповідає код main.py наведений в додатку А. Детальніше проаналізуємо написаний код. Першочергово, за допомогою ключового слова import підключаються необхідні бібліотеки. Далі я створюю масив з назвами трьох наборів даних, “os.mkdir(path)” створює директорію “data”. Для кожного з наборів даних (train, test, valid) створюються відповідні піддиректорії. Якщо директорія вже існує, виводиться повідомлення про помилку.

Функція extract_images відповідає за копіювання забражень з завантаженого датасету до цільової директорії “data” яку було створено раніше. Використовуючи “glob.glob” знаходимо всі файли з розширенням “.jpg” у вихідній директорії та за допомогою “shutil.copy” копіюємо їх. Наступним кроком викликаємо цю функцію для кожного з трох наборів, перед цим вказавши шлях до вихідної (source_path) та цільової (dst_path) директорій

Конвертація з формату COCO у формат YOLO відбувається за допомогою функції “convert_coco2YOLO”. Функція шукає файл з анотаціями у форматі COCO з розширенням “.json” аналогічним до попередньої функції способом. Імпортується COCO-анотація за допомогою “pylabel.importer.ImportCoco”. Анотації

конвертуються у формат YOLO і експортуються до цільової директорії за допомогою методу “ExportToYoloV8”. Викликається ця функція для кожного з трьох наборів даних.

Після виконання цього коду необхідно перенести файл “dataset.yaml” з папки “train” до папки “data”, файли з такою самою назвою в інших двох папках видаляються, вони непотрібні.

3.4.3. Створення файлу з конфігураційними параметрами

Для тренування моделі “yolov8n” необхідно створити конфігураційний файл з розширенням “.yaml”, в якому буде вказано шлях до кожного з трьох наборів даних, кількість класів об’єктів та список імен класів об’єктів.

Код представлений в додатку Б відповідає за це завдання. В строковій змінній “config_file_template” записуємо шаблоблон майбутнього конфігураційного файлу, вказуючи шлях до кожного набору даних та іншу інформацію. Далі вміст змінної записується у файл “dataset.yaml”.

3.4.4. Навчання моделі

Процес навчання вибраної моделі досить простий, основна перевага вибраного методу відсутність необхідності створювати модель з нуля. В додатку В наведено код який відповідає за навчання моделі. Необхідно ініціалізувати об'єкт моделі YOLO з конфігураційного файлу “yolov8n.yaml”, та завантажити ваги з файлу “yolov8n.pt”.

Команда “model.train” Запускає процес навчання моделі, в параметрах команди вказується конфігураційний файл “dataset.yaml”, кількість епох навчання – 15, та пристрій який використовується для навчання, в моєму випадку це процесор мого комп’ютера. Навчання моделі може тривати досить довго все залежить від характеристик комп’ютера.

Після навчання можна протестувати модель за допомогою зображень з директорії “test”.Результат навчання наведено на рис. 3.4, модель успішно виконала задачу детекції та виділення об’єкта.

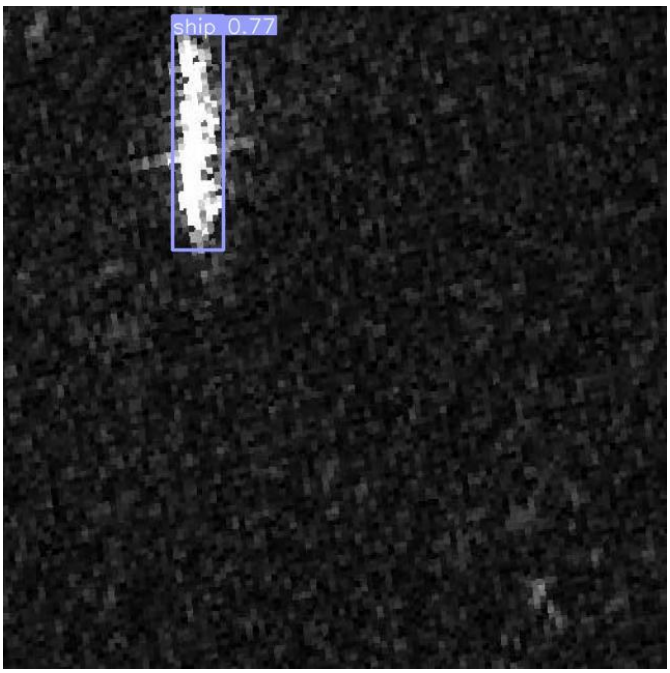

Рис. 3.4. Результат тестування навченої моделі

Після навчання моделі, в корені проєкту автоматично створюється папка “runs”, в ній зберігаються результати навчання моделі, графіки точності, та ваги, найкращі, та останні.

3.4.5. Створення інтерфейсу користувача

Щоб зручно використовувати навчену модель необхідно створити зручний інтерфейс користувача, з цією задачею допоможе впоратись бібліотека PyQt6. Користувач повинен самостійно мати можливість завантажувати зображення або набір зображень для подальшої обробки. Код написаного мною застосунку наведено в додатку Г. Програма складається з таких частин:

- Клас YOLOApp, це основний клас додатку, який успадковується від QWidget.

- У конструкторі (__init__) ініціалізується модель YOLO та інтерфейс користувача.

- initUI, метод для створення користувацького інтерфейсу. Цей метод задає назву вікна програми, розмір вікна програми, створює дві кнопки, одну для зображення, іншу для папки, і прогрес бар для відслідковування процесу обробки групи зображень.

- openFileDialog – метод, який відкриває діалогове вікно для вибору файлу, а саме зображення.

- openFolderDialog – метод, який відкриває діалогове вибору папки з зображеннями

- processImage – метод для обробки обраного зображення за допомогою моделі YOLO, збереження результату та відображення його в інтерфейсі.

- processFolder – метод для обробки набору зображень, та сповіщення користувача про успішну обробку.

3.5. Огляд роботи програми

Інтерфейс програми досить простий, і містить всього дві кнопки (Див. рис. 3.5).

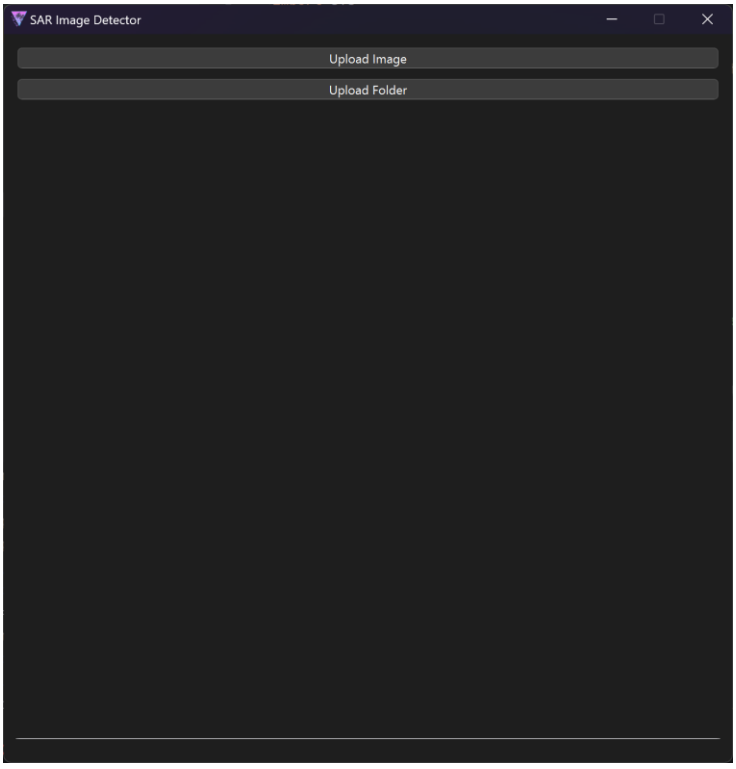

Рис. 3.5. Інтерфейс застосунку

Кнопка “Upload image” завантажує лише одне зображення для обробки, і після обробки виводить результат на у вікні програми (Рис. 3.6). Оброблене зображення зберігається в папці проєкту.

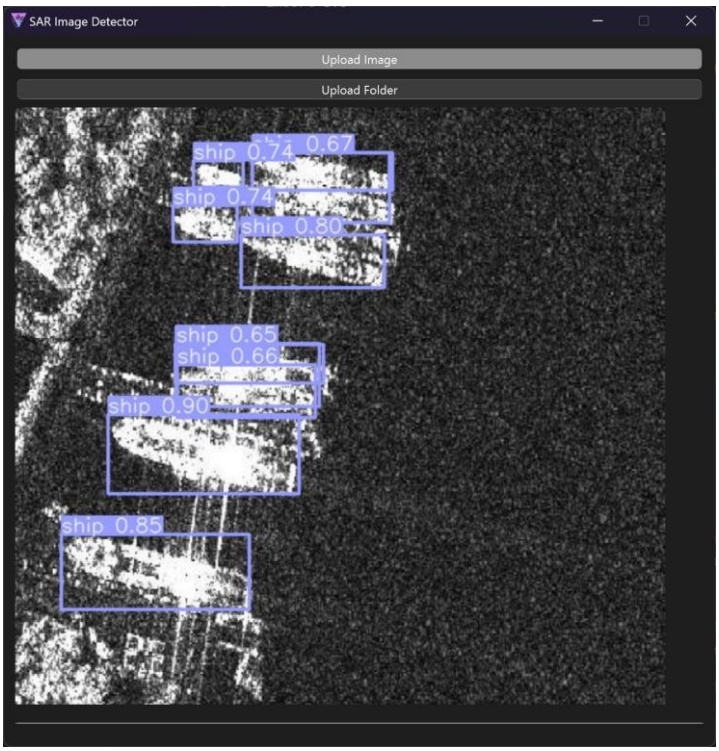

Рис. 3.6. Результат роботи програми при завантаженні одного зображення

Кнопка “Upload folder”, дозволяє користувачу завантажити папку з зображеннями які необхідно обробити. Під час обробки, внизу вікна програми, знаходиться шкала прогресу, яка показує скільки відсотків вже оброблено. Також у вибраній користувачем папці, програма створює окрему папку з назвою “processed_images”, для зображень які вже пройшли обробку (Рис. 3.5). Назва файлу, кожного обробленого зображення зберігається наступним чином - “processed_image_№_ + назва оригінального файлу”

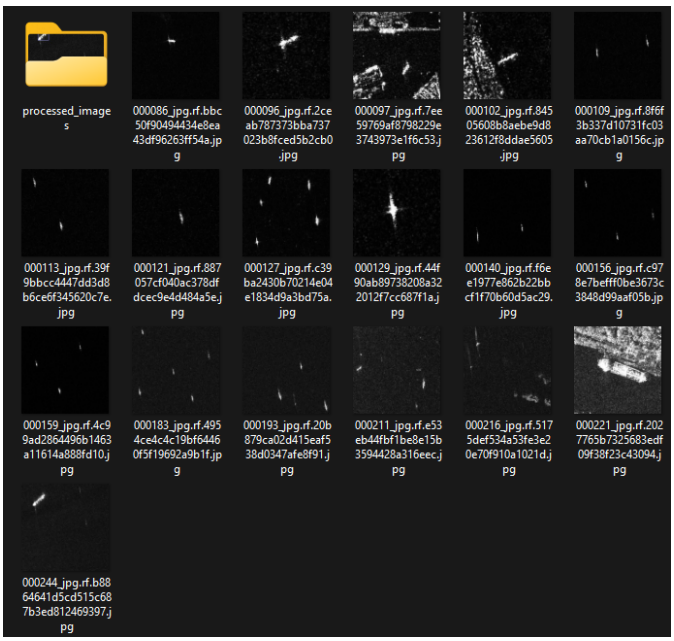

Рис. 3.5. Результат роботи застосунку при обробці групи зображень

ВИСНОВКИ

1. Актуальність SAR систем, доводить необхідність розробок програмних комплексі для обробки отриманих радіолокаційних зображень для використання в різних галузях, сільському господарстві, екології, моніторингу, військовій сфері. Система обробки таких зображень, полегшує роботу, яка раніше займала багато часу та ресурсів.

2. В роботі було проаналізовано можливості SAR, його параметри, медоти збирання даних. Описано системи які використовують радар з синтезованою апертурою для виявлення мін. Розглянуто принципи роботи та структуру згорткових нейронних мереж. Ахітектуру такої мережі детально розглянуто на прикладі YOLOv8

3. Використання систем інтелектуальної обробки радіолокаційних зображень, дозволяє швидко і якісно, а головне точно, проводити аналіз цих даних. Завдяки таким архітектурам згорткових нейронних мереж, як YOLO цей процес значно спрощується, що робить можливим інтеграцію таких систем для різних потреб моніторингу.

4. За допомогою вибраного набору даних навчено мережу YOLOv8. В даній роботі детально описано процес підготовки до навчання вибраної мережі, та процес написання програми. Розроблена програма інтегрує навчену мережу, для аналізу радіолокаційних зображень, та виявлення морських суден. Основною перевагою розробленої програми є швидкість аналізу зображень. За наявності набору радіолокаційних зображень з мінами різних типів, можна навчити обрану модель визначати тип міни, що становить велику цінність для задач гуманітарного розмінування.

5. Отже, поєднання SAR та останніх досягнень в області штучного інтелекту, відкривають перед нами широкі можливості. На жаль, такому поєднанню з метою впровадження цих програм в різні галузі, заважає брак якісних наборів даних для навчання нейронних мереж, але на мою думку в майбутньому таких наборів зображень у відкритому доступі буде з’являтися все більше, що дозволить популяризувати цей напрямок досліджень серед розробників різного рівня.

Список бібліографічних посилань використаних джерел

1. https://www.planetwatchers.com/services/synthetic-aperture-radar/

2. Gerhard Krieger, Irena Hajnsek, Konstantinos Panagiotis Papathanassiou,

Marwan Younis, Alberto Moreira «Interferometric Synthetic Aperture Radar (SAR) Missions Employing Formation Flying», Proceedings of the IEEE | Vol. 98, No. 5, May 2010

3. Gregory L. Charvat «Radar imaging in your garage: synthetic aperture radar», 2014

URL: https://hackaday.com/2014/03/17/radar-imaging-in-your-garage-synthetic-aperture-radar/

4. І. Тревого, А. Горб, О. Мелешко, «Застосування радарів із синтезованою апертурою для високоточного геопросторового моніторингу», Сучасні досягнення геодезичної науки та виробництва, випуск I (33), 2017

5.Ager, T.P. 2013. «An introduction to synthetic aperture radar imaging» Oceanography 26(2):20–33

6. Andreas Braun, «Radar satellite imagery for humanitarian response Bridging the gap between technology and application» 20ст, 2019

7. P. Berens, «Introduction to Synthetic Aperture Radar (SAR)», Advanced Radar Signal and Data Processing (pp. 3-1-3-14), 2006

8. SAR-Guidebook, 2009

9. https://www.findmine.org/gpsar

10. M. Bradley, T. Witten, M. Duncan, B. McCunmmins, «Mine detection with a forward-looking ground-penetrating synthetic aperture radar», SPIE Vol. 5089, 2003

11. M. Schartel, R. Burr, W. Mayer, C. Waldschmidt, «Airborne Tripwire Detection using a Synthetic Aperture Radar», Journal of Latex Class Files Vol. 14, No. 8, August 2015

12. https://github.com/ultralytics/ultralytics?tab=readme-ov-file

13. https://medium.com/@VK_Venkatkumar/yolov8-architecture-cow-counter-with-region-based-dragging-using-yolov8-e75b3ac71ed8

14. https://w3schoolsua.github.io/python/index.html#gsc.tab=0

15. https://docs.ultralytics.com

16. https://www.pythonguis.com/pyqt6-tutorial/

17. https://www.jetbrains.com/pycharm/

18. https://www.kaggle.com/datasets/kailaspsudheer/sarscope-unveiling-the-maritime-landscape/code

Додаток А

Лістинг main.py

```
import os
import glob
import shutil

from pylabel import importer

splits = ['train', 'test', 'valid']

path = 'data/'
os.mkdir(path)
for spl in splits:
    try:
        os.mkdir(path+spl)
    except OSError as error:
        print(error)


def extract_images(source_path, dst_path):
    if not os.path.exists(dst_path):
        os.mkdir(dst_path)
    image_path = glob.glob(source_path+"/*.jpg")
    for path in image_path:
        shutil.copy(path, dst_path)


source_path = "D:/SARscope/train"
dst_path = "D:/Programing/Py/diploma/data/train/images"
extract_images(source_path, dst_path)

source path = "D:/SARscope/test"
dst_path = "D:/Programing/Py/diploma/data/test/images"
extract_images(source_path, dst_path)

source_path = "D:/SARscope/valid"
dst_path = "D:/Programing/Py/diploma/data/valid/images"
extract_images(source_path, dst_path)


def convert_coco2YOLO(source_path, dst_path):
    if not os.path.exists(dst_path):
        os.mkdir(dst_path)
    json_file = glob.glob(source_path+"/*.json")
    dataset = importer.ImportCoco(json_file[0], path_to_images=source_path,
name="SARscope")
    dataset.export.ExportToYoloV8(dst_path)


source_path = "D:/SARscope/train"
dst_path = "D:/Programing/Py/diploma/data/train/labels"
convert_coco2YOLO(source_path, dst_path)

source_path = "D:/SARscope/test"
dst_path = "D:/Programing/Py/diploma/data/test/labels"
convert_coco2YOLO(source_path, dst_path)

source_path = "D:/SARscope/valid"
dst_path = "D:/Programing/Py/diploma/data/valid/labels"
convert_coco2YOLO(source_path, dst_path)
```

Додаток Б

Лістинг cfg.py

```
import yaml

config_file_template = '''
val: ../data/valid/images
train: ../data/train/images
test: ../data/test/images

nc: 2
names ['background', 'ship']
'''

with open('data/dataset.yaml', 'w') as f:
    f.write(config_file_template)


with open('data/dataset.yaml', 'r') as f:
    file_path = yaml.safe_load(f)
print(file_path)
```

Додаток В

Лістинг model.py

```
import glob

from IPython.core.display_functions import display
from ultralytics import YOLO
from PIL import Image


model = YOLO('yolov8n.yaml').load('yolov8n.pt')

model.train(data="dataset.yaml", epochs=15, imgsz=640, device='cpu')

model.export()


test_images = glob.glob("D:/SARscope/test/*.jpg")
results = model(test_images[4], stream=False)

res = results[0].plot()
Image.fromarray(res)

test_model = YOLO("runs/detect/train/weights/best.pt")
results = test_model(test_images[:4], stream=False)
for i in range(4):
    res = results[i].plot()
    display(Image.fromarray(res))
```

Додаток Г

Лістинг app.py

```
import sys
import os
from PyQt6.QtCore import QSize, Qt
from PyQt6.QtWidgets import QApplication, QWidget, QPushButton, QLabel, QFileDialog,
QVBoxLayout, QMessageBox, \
    QProgressBar
from PyQt6.QtGui import QPixmap, QIcon
from PIL import Image
from ultralytics import YOLO


class YOLOApp(QWidget):
    def __init__(self):
        super().__init__()
        self.model = YOLO("runs/detect/train/weights/best.pt")

        self.initUI()

    def initUI(self):
        self.setWindowTitle('SAR Image Detector')
        self.setFixedSize(QSize(700, 700))
        self.layout = QVBoxLayout()

        self.upload_btn = QPushButton('Upload Image', self)
        self.upload_btn.clicked.connect(self.openFileDialog)
        self.layout.addWidget(self.upload_btn)

        self.upload_folder_btn = QPushButton('Upload Folder', self)
        self.upload_folder_btn.clicked.connect(self.openFolderDialog)
        self.layout.addWidget(self.upload folder btn)

        self.image_label = QLabel(self)
        self.layout.addWidget(self.image_label)

        self.progress_bar = QProgressBar(self)
        self.progress_bar.setAlignment(Qt.AlignmentFlag.AlignCenter)
        self.layout.addWidget(self.progress_bar)

        self.setLayout(self.layout)

    def openFileDialog(self):
        file_dialog = QFileDialog(self)
        file_dialog.setNameFilter("Images (*.png *.xpm *.jpg *.jpeg)")
        if file_dialog.exec():
            file_path = file_dialog.selectedFiles()[0]
            self.processImage(file_path)

    def openFolderDialog(self):
        folder_dialog = QFileDialog(self)
        folder_dialog.setFileMode(QFileDialog.FileMode.Directory)
        if folder_dialog.exec():
            folder_path = folder_dialog.selectedFiles()[0]
            self.processFolder(folder_path)

    def processImage(self, file_path):
        results = self.model(file_path, stream=False)
        res = results[0].plot()
        img = Image.fromarray(res)
        img.save("processed_image.jpg")
```

```
        pixmap = QPixmap("processed_image.jpg")
        self.image_label.setPixmap(pixmap)

    def processFolder(self, folder_path):
        jpg_files = [f for f in os.listdir(folder_path) if
f.lower().endswith('.jpg')]

        if not jpg_files:
            QMessageBox.warning(self, 'Error', 'No .jpg files found in the selected
folder.')
            return

        output_dir = os.path.join(folder_path, 'processed_images')
        os.makedirs(output_dir, exist_ok=True)

        self.progress_bar.setMaximum(len(jpg_files))
        self.progress_bar.setValue(0)

        for i, jpg_file in enumerate(jpg_files):
            file_path = os.path.join(folder_path, jpg_file)
            results = self.model(file_path, stream=False)
            res = results[0].plot()
            img = Image.fromarray(res)

            output_file_path = os.path.join(output_dir, f'processed_image_{i +
1}_{jpg_file}')
            img.save(output_file_path)

            self.progress_bar.setValue(i + 1)

        QMessageBox.information(self, 'Success', f'Processed images are saved in
{output_dir}')
        self.progress_bar.setValue(0)  # Reset the progress bar


if __name__ == '__main__':
    app = QApplication(sys.argv)
    app.setWindowIcon(QIcon('icon.png'))
    ex = YOLOApp()
    ex.show()
    sys.exit(app.exec())
```