

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ  
СІКОРСЬКОГО»  
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ  
Кафедра математичного моделювання та аналізу даних**

«На правах рукопису»  
УДК 004.93

ДО ЗАХИСТУ ДОПУЩЕНО  
Завідувач кафедри

\_\_\_\_\_ Наталія Куссуль  
«\_\_» \_\_\_\_\_ 2022 р.

**Магістерська дисертація  
на здобуття ступеня магістра  
за освітньо-професійною програмою «Математичні методи  
моделювання, розпізнавання образів та комп'ютерного зору»  
зі спеціальності 113 «Прикладна математика»  
на тему «Нейромережева модель розпізнавання вирв від  
бомбардування за супутниковими даними»**

Виконав:

Студент 2 курсу, групи ФІ-11мп  
Поприго Ярослав Леонідович

\_\_\_\_\_

Науковий керівник:

Завідувач кафедри, д.т.н., професор  
Куссуль Наталія Миколаївна

\_\_\_\_\_

Рецензент

к.т.н, старший науковий співробітник  
Яйлимов Богдан Ялкапович

\_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент \_\_\_\_\_

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ  
УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Навчально-Науковий Фізико-Технічний Інститут

Кафедра математичного моделювання та аналізу даних

Рівень вищої освіти другий (магістерський)

Спеціальність 113 Прикладна Математика

Освітньо-професійна програма Математичні методи моделювання,  
розпізнавання образів та комп'ютерного зору

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Наталія Куссуль  
«\_\_» \_\_\_\_\_ 2022 р.

## ЗАВДАННЯ

на магістерську дисертацію роботу студенту

Поприго Ярославу Леонідовичу

(прізвище, ім'я, по батькові)

1. Тема дисертації Нейромережева модель розпізнавання вирв від бомбардування за супутниковими даними

науковий керівник дисертації д.т.н., професор Куссуль Наталія Миколаївна

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені наказом по університету від «\_\_» 2022 р. № \_\_\_\_\_

2. Строк подання студентом дисертації грудня 2022 р
3. Об'єкт дослідження: поверхня верхнього шару земної кори, яка була пошкоджена внаслідок військових дій.
4. Вихідні дані роботи: теоретичні та аналітичні матеріали, знімки супутника Sentinel-2.
5. Перелік завдань, які потрібно розробити:
  1. Проаналізувати супутникові знімки в районах ведення бойових дій на

предмет наявності пошкоджених територій

2. Підготувати датасет зі супутникових знімків взятих з пошкоджених територій для подальшої можливості їх аналізу.
  3. Розробити нейромережеву модель розпізнавання вирв від бомбардувань за супутниковими даними.
  4. Проаналізувати результати роботи моделі та оцінити її ефективність.
6. Перелік графічного (ілюстративного) матеріалу: таблиці, рисунки.
7. Консультанти розділів дисертації відсутні.
8. Дата видачі завдання 01.09.2022

Календарний план

№ з/п	Назва етапів виконання роботи	Термін виконання етапів роботи	Підпис керівника
1	Визначення з темою роботи	01.08.2022	
2	Актуальність, область застосування, постановка задачі	01.09.2022	
3	Огляд відомих підходів та існуючих рішень	10.09.2022-10.10.2022	
4	Розробка методу розпізнавання вирв	10.10.2022-31.10.2022	
5	Програмна реалізація проекту	01.11.2022-01.12.2022	
6	Представлення до захисту	19.12.2022	

Студент \_\_\_\_\_

Ярослав ПОПРИГО

Науковий керівник дисертації \_\_\_\_\_

Наталія КУССУЛЬ

## РЕФЕРАТ

Магістерська дисертація містить 109 сторінок, 20 ілюстрацію, і 181 джерел літератури.

Наразі задача розпізнавання вирв від бомбардувань стає все гострішою. Після повномасштабної військової агресії російської федерації, чимало фондів намагаються оцінити збитки, які були нанесені об'єктам інфраструктури, цивільним будівлям, тощо.

Нейромережева модель розпізнавання вирв від бомбардувань за супутниковими даними дасть змогу комплексно та всеціло оцінити масштаб руйнувань, який в подальшому може бути використаний для підрахунку збитків.

Для досягнення мети було використано:

- Нейромережеву модель U-Net
- Google Collaboratory
- Бібліотеки pytorch, torchvision, matplotlib, Pillow, imutils, scikit-learn, tqdm, gdal, numpy

Ключові слова: НЕЙРОМЕРЕЖЕВА МОДЕЛЬ, МАШИННЕ НАВЧАННЯ, U-NET, RESNET-34, СУПУТНИКОВІ ЗНІМКИ, SENTINEL-2, ФУНКЦІЯ ВТРАТ, ДАТАСЕТ, ВИРВИ ВІД БОМБАРДУВАНЬ.

## ABSTRACT

The master's thesis contains 109 pages, 20 illustrations, and 181 references.

Nowadays the task of recognizing explosions from bombings is becoming more and more acute. After the full-scale military aggression of the Russian Federation, many funds are trying to assess the damage that was caused to infrastructure objects, civilian buildings, etc.

A neural network model for recognizing bombardment eruptions based on satellite data will enable a comprehensive and comprehensive assessment of the scale of destruction, which can later be used to estimate damages.

To achieve the goal, was used the following:

- U-Net neural network model
- Google Collaboratory
- Libraries pytorch, torchvision, matplotlib, Pillow, imutils, scikit-learn, tqdm, gdal, numpy

Keywords: NEURAL NETWORK MODEL, MACHINE LEARNING, U-NET, RESNET-34, SATELLITE IMAGES, SENTINEL-2, LOSS FUNCTION, DATASET, ESCAPES FROM BOMBING.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ ДО РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ЗА СУПУТНИКОВИМИ ДАНИМИ	11
1.1 Глибоке навчання для землекористування та класифікації земельного покриву на основі даних гіперспектральних і мультиспектральних спостережень Землі	11
1.2 Класифікація землекористування та земельного покриву	12
1.3 Мультиспектральні та гіперспектральні дані дистанційного зондування	17
1.3.1 Джерела даних і набори даних	19
1.4. Машинне навчання для LULC	23
1.4.1. Наскрізне глибоке навчання	26
1.4.2. Розробка функцій	28
1.4.2.1. Вибір і трансформація функції	31
1.4.2.2. Вилучення функцій	32
1.4.3. Класифікатор	33
1.4.4. Виклик обмеженої ґрунтової правди	42
1.4.5. Мультимодальне злиття даних	48
1.4.6. Попередня і постобробка	52
1.5. Висновки до розділу 1	53
РОЗДІЛ 2. НЕЙРОННІ МЕРЕЖІ ДЛЯ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ ТА ПІДГОТОВКА ДАНИХ	56
2.1 Місце згорткових мереж у сегментації зображень	56
2.2 Типи згорткових нейронних мереж	57
2.2.1 Повністю згорткові нейронні мережі	57
2.2.2 Мережі кодера - декодера	59
2.3 Підготовка даних	59
2.4 Висновки до розділу 2	61
РОЗДІЛ 3. РЕАЛІЗАЦІЯ НЕЙРОМЕРЕЖЕВОЇ МОДЕЛІ РОЗПІЗНАВАННЯ ВИРВ ВІД БОМБАРДУВАННЯ ЗА СУПУТНИКОВИМИ ДАНИМИ	62
3.1 Попередня обробка даних	62
3.2 Підготовка датасету	63

3.3 Функції втрат	7
3.4 Архітектура моделі	66
3.5 Огляд результатів	69
3.5.1 Експеримент 1	70
3.5.2 Експеримент 2	70
3.6 Висновки до розділу 3	73
ВИСНОВКИ	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	76
ДОДАТОК 1. ТЕКСТИ ПРОГРАМ	94

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

**PoC** – Proof of Concept – демонстрація спроможності практичної реалізації рішення.

**GAN** – генеративна змагальна мережа.

**NDVI** – нормалізований індекс різниці рослинності.

**U-Net** – тип нейронної мережі, архітектура якої розроблена для сегментації зображень.

**FCN** – (Fully Connected Network) повністю згорточна мережа.

**QGIS** – (Quantum GIS) середовище для роботи з геопросторовими даними.

**IoU** – (Intersection over Union) функція втрати, яка вимірює збіг між двома наборами даних.

**Dice Loss** – функція втрат кубиків.

**Focal Loss** – фокусна функція втрат.

**Combo Loss** – комбінована функція втрат, що містить в собі поєднання Dice Loss та IoU.



## ВСТУП

### **Актуальність роботи.**

З початком повномасштабної агресії збройних сил росії на територію України питання відновлення пошкоджених територій стало одним з найважливіших питань післявоєнного відновлення. За даними національної ради з відновлення України від наслідків війни ключовою задачею наразі є розроблення системи для фіксації руйнувань та пошкоджень, в тому числі із використанням єдиної картографічної основи, а також застосуванням технологій дистанційного зондування Землі для моніторингу масових ушкоджень та руйнувань.

### **Зв'язок дослідження з науковими програмами, планами, темами.**

Роботу виконано в межах проекту Світового банку з супутникового моніторингу землекористування (договір №1538920 від 13 січня 2022 р.), проекту НФДУ “Методи і моделі глибинного навчання для прикладних задач супутникового моніторингу” (2020.02/0292) та міжнародної ініціативи EO4UA (<https://cloudferro.com/en/eo4ua/>).

### **Мета і завдання дослідження.**

Розробити нейромережеву модель, за допомогою якої було б можливо визначати кількість та площу вирв від бомбардувань, які були утворень в результаті ведення бойових дій.

Для досягнення поставленої мети необхідно виконати наступні завдання:

1. Проаналізувати поточний стан досліджень в сфері тематичної сегментації супутникових знімків з метою визначення типів земного покриву та землекористування.
2. Побудувати нейромережеву модель ідентифікації пошкоджень сільськогосподарських полів (вирв від бомбардування), спричинених

війною.

3. Провести обчислювальний експеримент для визначення точності моделі.

#### **Об'єкт дослідження.**

Об'єктом дослідження є території, які пошкоджені в результаті бойових дій (Київська область в передмісті Ірпеня)

#### **Предмет дослідження.**

Предметом дослідження є математичні моделі машинного навчання для визначення пошкоджень на сільськогосподарських полях в результаті бойових дій (Київська область в передмісті Ірпеня)

#### **Методи дослідження.**

Для розв'язання поставленої задачі використано методи машинного навчання, математичної статистики, геопросторового аналізу за допомогою програми QGIS з використанням супутникових знімків Sentinel-2.

#### **Практичне значення одержаних результатів.**

Отримані результати магістерської дисертації можуть бути використані міжнародними фондами як PoC рішення, яке дозволяє оцінити масштаб руйнувань в результаті бойових дій, а також для оцінки втрат сільськогосподарської продукції.

## **РОЗДІЛ 1 ОГЛЯД ІСНУЮЧИХ ПІДХОДІВ ДО РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ ЗА СУПУТНИКОВИМИ ДАНИМИ**

У цьому розділі буде проаналізовано вже існуючі підходи до розпізнавання зображень за супутниковими даними.

### **1.1 Глибоке навчання для землекористування та класифікації земельного покриву на основі даних гіперспектральних і мультиспектральних спостережень Землі**

Останнім часом, коли глибоке навчання випереджає інші методи машинного навчання в класифікації зображень, ми спостерігаємо зростаючий інтерес спільноти дистанційного зондування до використання цих методів для класифікації землекористування та земельного покриву на основі мультиспектральних і гіперспектральних зображень; кількість пов'язаних публікацій, яка щорічно з 2015 року зростає майже вдвічі, є тому підтвердженням. Прогрес у технологіях дистанційного зондування, а отже, швидко зростаючий обсяг своєчасних даних, доступних у глобальному масштабі, відкриває нові можливості для різноманітних застосувань. Глибоке навчання, яке значно успішно працює з великими даними, здається чудовим кандидатом для використання потенціалу таких складних масивних даних.

Однак існують деякі проблеми, пов'язані з базовою істинністю, роздільною здатністю та характером даних, які сильно впливають на ефективність класифікації. У цій статті ми розглядаємо використання глибокого навчання у землекористуванні та класифікації земельного покриву на основі мультиспектральних та гіперспектральних зображень, а також представляємо доступні джерела даних та набори даних, які використовуються в літературних дослідженнях; ми надаємо читачам структуру для інтерпретації найсучаснішого глибокого навчання в цьому контексті та пропонуємо платформу для підходу до методологій, даних і викликів галузі.

## 1.2 Класифікація землекористування та земельного покриття

Земельні карти Землі традиційно поділяють на класифікацію землекористування та класифікацію земельного покриття. Хоча в багатьох дослідженнях ці два поняття є взаємозамінними або, як зазначено в [9], плутаються одне з одним, належне визначення кожного робить їх різними. За даними Продовольчої та сільськогосподарської організації (ФАО) [10] ООН, «грунтовий покрив — це спостережуваний (біо)фізичний покрив на поверхні Землі», тоді як «землекористування характеризується механізмами, діяльністю та вкладом людей виробляти, змінювати або підтримувати певний тип ґрунтового покриття». Згідно з визначенням, землекористування та ґрунтовий покрив тісно пов'язані, і їх спільна класифікація майже неминуха. Тому в останніх дослідженнях класифікація «землекористування та земельного покриття» (Land Use/Land Cover - LULC) у цілому розглядається як більш загальна концепція, яка також охоплює цей зв'язок.

Існують різні таксономії для LULC, засновані на цільових програмах; одне з найвідоміших визначень належить ФАО і пропонує ієрархічну систему класифікації ґрунтового покриття (LCCS), яка забезпечує можливість розміщення різних рівнів інформації, починаючи зі структурованих класів широкого рівня, які дозволяють подальший систематичний підрозділ на більш детальні підгрупи. класів (рис. 1). Це визначення забезпечує високий рівень картографування, яке також охоплює визначені користувачем дескриптори землекористування.

1st level	Primarily Vegetated Areas				Primarily Non-Vegetated Areas			
2nd level	Terrestrial Primarily Vegetated Areas		Terrestrial Primarily Non-Vegetated Areas		Aquatic or Regularly Flooded Primarily Vegetated Areas		Aquatic or Regularly Flooded Primarily Non-Vegetated Areas	
3rd level	Cultivated and Managed Terrestrial Areas	Natural and Semi-Natural Vegetation	Cultivated Aquatic or Regularly Flooded Areas	Natural and Semi-Natural Aquatic or Regularly Flooded Vegetation	Artificial Surfaces and Associated Areas	Bare Areas	Artificial Waterbodies, Snow and Ice	Natural Waterbodies, Snow and Ice
...	..	..	..	..	..	..	..	..

Рисунок 1 – Три категорії верхнього рівня в ієрархії системи класифікації земельного покриття (LCCS).

Загалом дослідження, що наближаються до класифікації LULC, розглядають дуже невелику кількість категорій ґрунтового покриття або землекористування. Залежно від цільового застосування, ці категорії можуть бути на вищому рівні ієрархії, виділяючи очевидні земельні покриття або зосереджуючись на конкретних категоріях підкласів земельного покриття. Класифікація водно-болотних угідь [12,13], міського землекористування [14,15], сільського господарства [16], лісів [17] та інших карт рослинності є деякими прикладами орієнтованих на застосування підходів до класифікації LULC, які доступні в літературі.

Перше використання даних дистанційного зондування для класифікації LULC датується серединою 1940-х років, коли Френсіс Дж. Маршнер почав складати карту всіх Сполучених Штатів, пов'язуючи землекористування з поверхнею Землі за допомогою аерофотозйомки [18]. Роки потому, відразу після запуску дослідницького супутника Earth Resources Technologies, оснащеного мультиспектральним сканером (MSS) у липні 1972 року, і початку програми Landsat, дослідження з використанням даних дистанційного зондування

зображень для класифікації LULC вийшли на новий рівень. [19,20]. Насправді, із зародженням програми Landsat і (приватним) оприлюдненням даних, нові виклики мультимодального об'єднання даних, виявлення змін землі на часовій основі та екологічного застосування супутникових даних постали в галузі LULC. Деякі з ранніх робіт на ці теми обговорюються в [21,22,23,24].

Дослідження щодо класифікації LULC та її подальших викликів постійно та швидко розвиваються в результаті швидкого вдосконалення обчислювальних і запам'ятовуючих можливостей комп'ютерів та еволюції штучного інтелекту (ШІ). Крім того, будь-який прогрес у технологіях дистанційного зондування та в якості даних дає дослідникам нові можливості отримувати нову інформацію з даних дистанційного зондування [25]. Зростаюча тенденція публікацій про класифікацію LULC даних дистанційного зондування зображена на рисунку 2. Тенденцію було зафіксовано під час пошуку набору ключових термінів у заголовку, анотації та ключових словах усіх документів, доступних у Scopus, згрупованих і відфільтрованих за п'ятьма - річними інтервалами.

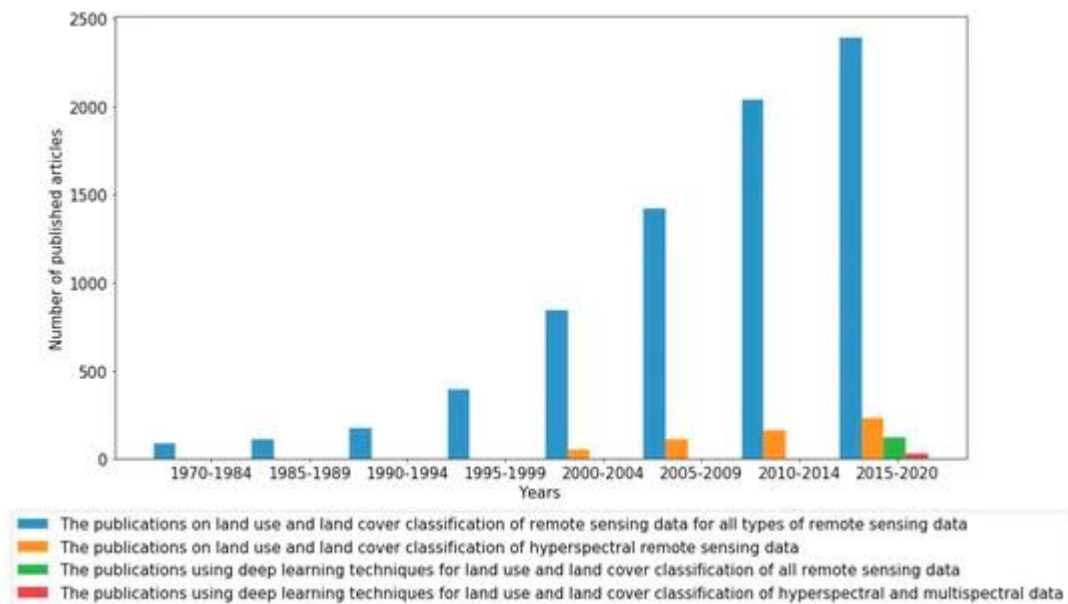


Рисунок 2 – Тенденції публікацій щодо класифікації LULC даних дистанційного зондування.

На графіку видно послідовне зростання кількості публікацій. На графіку також показано частку публікацій, присвячених класифікації гіперспектральних зображень і використанню методів глибокого навчання (дані були отримані в

травні 2020 року).

Тенденції на рисунку 2 містять чотири різні результати пошуку: перший (синій) – це кількість публікацій про класифікацію/сегментацію LULC з використанням усіх типів даних дистанційного зондування. Другий (помаранчевий) обмежує публікації про класифікацію/сегментацію LULC гіперспектральними даними: це підкреслює збільшення кількості досліджень, які працюють над такими даними за останні два десятиліття. Третій (Green) показує використання методів «глибокого навчання» в класифікації LULC з усіма типами даних дистанційного зондування, що виникли в останні роки (зацікавлені читачі можуть знайти огляд на такі публікації в [26]). Останній (червоний) обмежує останній тип, розглядаючи лише використання мультиспектральних і гіперспектральних даних дистанційного зондування, які все частіше привертають увагу через їх недавню доступність.

Гіперспектральне зображення, пов'язане з досягненнями в цифровій електроніці та обчислювальних можливостях, було пізніше прийнято спільнотою спостереження за Землею через його складність за своєю природою та обмеження обчислень того часу. Однак великий потенціал таких даних, їх доступність і швидкий розвиток обчислювальних технологій все більше приваблюють вчених, які цікавляться класифікацією LULC. Крім того, надзвичайні досягнення глибокого навчання після ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [27] спонукали вчених із дистанційного зондування також використовувати ці методи для даних дистанційного зондування, починаючи з 2015 року. Робота [28] присвячена використанню гіперспектральних зображень та огляду сучасних методологій глибокого навчання, які використовуються для класифікації гіперспектральних зображень. Посилання [29] також представляє огляд методів глибокого навчання для класифікації гіперспектральних зображень і порівнює ефективність цих методів на загальновідомих наборах даних.

У цьому розділі ми досліджуємо цю недавню зростаючу дослідницьку

тенденцію щодо використання методів глибокого навчання для класифікації LULC на основі гіперспектральних і мультиспектральних зображень, оскільки обидва типи даних мають спільні атрибути, які можна вивчати разом. Метою цього розділу є аналіз відповідних методів і наборів даних для вирішення існуючих проблем у цій галузі.

З огляду на Рисунок 2, у цьому документі розглядаються статті, виділені червоним кольором (класифікація LULC гіперспектральних і мультиспектральних зображень дистанційного зондування з використанням методів глибокого навчання), отримані за допомогою пошукового запиту: («глибоке навчання» АБО «згорточна нейронна мережа» І «land cover» АБО «landcover» АБО «землекористування» АБО «landuse» АБО «lulc» І «мультиспектральний» АБО «мультиспектральний» АБО «гіперспектральний» АБО «гіперспектральний»). Ми розглянули документи, на які цитувалися ці вибрані статті, та інші роботи, на які цитувалися ці вибрані статті. Перегляд цих джерел допоміг нам накреслити загальну схему найсучаснішого стану, яку ви знайдете нижче, зосередившись на позиції глибокого навчання в цілій картині. Як додаткову примітку, ми наголошуємо тут роз'яснення для читача щодо використання терміну «класифікація земельного покриву» в літературі, оскільки в кількох роботах він насправді відноситься до «сегментації земельного покриву». Іншими словами, термін класифікації відноситься до піксельного рівня, отже кінцевим цільовим результатом є сегментована карта. У деяких роботах метою класифікації є класифікація на основі патчів, де фрагмент зображення фіксованого розміру призначається до певного класу.

У цьому оглядовому документі для ясності ці підходи називаються «класифікацією на рівні пікселів» і «класифікацією на рівні патчів» відповідно. Заради простоти ми використовуємо термін «класифікація ґрунтового покриву» для класифікації на рівні пікселів, якщо це не зазначено явно.

**З формальної точки зору процес класифікації LULC визначається як  $f: X \rightarrow Y$ , із вхідним простором  $X \subseteq \mathbb{N}^W \times \mathbb{N}^H \times \mathbb{K}$ , де  $W, H, K$  відповідно ширина, висота та кількість спектральних**



смуг для кожного входу зображення, вихідний простір якого для класифікації земельного покриття на рівні пікселів і класифікації земельного покриття на рівні плям представлено як  $Y \subseteq CW \times H$  і  $Y \subseteq C$  відповідно, де  $C = \{\Omega_0, \Omega_1, \dots, \Omega_k\}$  є набором можливих категорії землекористування та земельного покриття.

### 1.3 Мультиспектральні та гіперспектральні дані дистанційного зондування

Зображення, отримані за допомогою дистанційного зондування, зазвичай отримують за допомогою оптичних, теплових систем або радарів із синтезованою апертурою (SAR). Оптичний датчик чутливий до діапазону спектру від видимого до середнього інфрачервоного випромінювання, що випромінюється від поверхні Землі, і створює панхроматичні, мультиспектральні або гіперспектральні зображення.

Тепловізійні датчики, які вловлюють теплове випромінювання від поверхні Землі, натомість чутливі до діапазону інфрачервоних хвиль від середніх до довгих хвиль. На відміну від теплових і оптичних датчиків, які працюють пасивно, датчик SAR є активним мікрохвильовим приладом, який освітлює землю, розсіюючи мікрохвильове випромінювання та вловлюючи відбиті хвилі від поверхні Землі.

Панхроматичний датчик — це моноспектральний каналний детектор, який вловлює випромінювання в широкому діапазоні довжин хвиль в одному каналі, тоді як мультиспектральні та гіперспектральні датчики збирають дані в кількох каналах. Таким чином, на відміну від панхроматичних продуктів, які є одношаровими 2D-зображеннями, гіперспектральні та багатоспектральні зображення мають подібну 3D-структуру з шарами зображень, кожен з яких представляє випромінювання в межах спектрального діапазону. Незважаючи на

подібність 3D-структури, головна відмінність мультиспектральних зображень від гіперспектральних полягає в кількості спектральних смуг. Зазвичай зображення з більш ніж 2 і до 13 спектральними смугами називають мультиспектральними, тоді як зображення з більшою кількістю спектральних смуг називають гіперспектральними.

Тим не менш, основна відмінність полягає в тому, що гіперспектральне отримання спектру для кожного пікселя зображення є безперервним, тоді як для мультиспектрального воно є дискретним (Рисунок 3—Ліворуч).

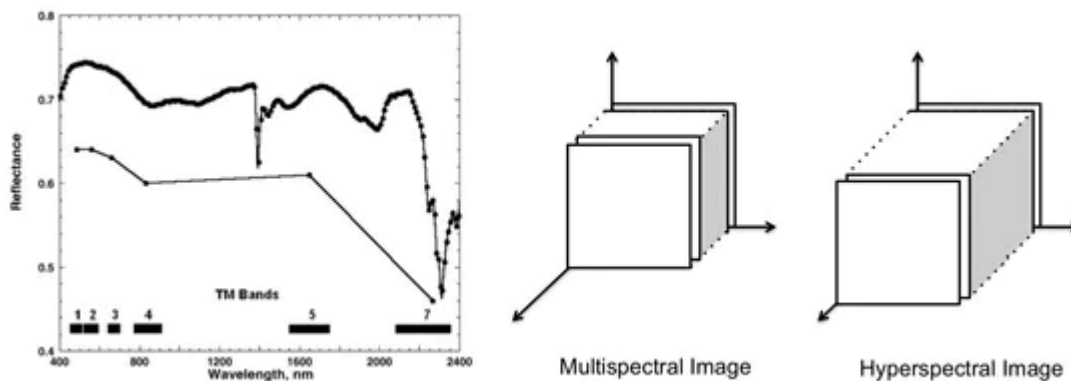


Рисунок 3 – Ліворуч: Отримання довжини хвилі спектральних смуг для мультиспектральної (внизу) та гіперспектральної вибірки (вгорі) (взято з [30]).  
Справа: схема мультиспектральних та гіперспектральних зображень у просторово-спектральній області.

Маючи сотні вузьких і безперервних спектральних смуг, гіперспектральні зображення (HSI) супроводжуються специфічними проблемами, властивими їхній природі, яких не існує з мультиспектральними (MSI) і панхроматичними зображеннями. Ці проблеми включають: (1) велику розмірність HSI, (2) різні типи шуму для кожного діапазону, (3) невизначеність спостережуваного джерела та (4) нелінійні зв'язки між отриманою спектральною інформацією [31]. Останнє пояснюється результатом розсіювання навколишніх об'єктів під час процесу отримання даних, різних атмосферних і геометричних спотворень, а також внутрішньокласової мінливості подібних об'єктів.

Незважаючи на згадані відмінності в природі MSI та HSI, обидва мають подібну тривимірну кубічну структуру (Рисунок 3—праворуч) і переважно

використовуються для подібних цілей. Дійсно, ідея класифікації/сегментації LULC спирається на морфологічні характеристики та матеріальні відмінності наземних регіонів і предметів, які відповідно можна отримати з просторової та спектральної інформації, доступної як у MSI, так і в HSI. Таким чином, на відміну від [32], де розглядаються методології, розроблені для об'єднання спектрально-просторової інформації лише для класифікацій гіперспектральних зображень, у цьому огляді ми розглядаємо обидва типи даних, які використовуються в літературі для класифікації земельного покриття з використанням методів глибокого навчання, зосереджених на спектральних та/або просторові характеристики корельованих пікселів земного покриття.

### **1.3.1 Джерела даних і набори даних**

Існує багато постачальників супутникових і повітряних знімків, які безкоштовно надають громадськості своєчасні дані дистанційного зондування з високою роздільною здатністю. USGS [33,34], NEO [35], Copernicus open access hub [36], NASA Earth data [37], NOAA [38,39] і IPMUS Terra [40] є одними з найпопулярніших постачальників даних дистанційного зондування з відкритим доступом. У літературі супутникові зображення, які використовуються для глибокого навчання, в основному отримані з Landsat-7, Landsat-8, Sentinel-1, Sentinel-2, WorldView-2, WorldView-3, QuickBird, EO-1, PROBA-1 і Супутники SPOT-6. У таблиці 1 наведено короткий огляд стану цих супутників і зображень із них. За винятком Sentinel-1, EO-1 і PROBA-1, які створюють як SAR, так і гіперспектральні зображення, продукти інших супутників, перерахованих у таблиці, є мультиспектральними зображеннями. Як пояснювалося раніше, зображення панхроматичних смуг (чорно-білі) фіксуються одноканальним детектором, чутливим до широкого діапазону довжин хвиль, що збігається з видимим діапазоном, який збирає більшу кількість сонячного випромінювання. Тому просторова роздільна здатність панхроматичних зображень зазвичай вища за MSI. Landsat, WorldView, SPOT-6 і QuickBird створюють панхроматичні

зображення разом із MSI. Серед постачальників MSI Sentinel-2, з найбільшою кількістю спектральних діапазонів (13 діапазонів) і найвищою орбітальною висотою серед цих супутників, є єдиною місією, яка може надати дані з глобальним покриттям за п'ять днів.

Таблиця 1. Короткий огляд супутників і їхніх зображень дистанційного зондування, які часто використовуються в літературі для практики глибокого навчання.

**Table 1.** A short overview of satellites and their remote sensing image products that have been frequently used in literature for deep learning practices.

Name	Launch Year	Orbital Altitude	Still Active (2019)	Image Type				Pixel Spatial Resolution
				SAR	Pan	MSI	HSI	
EO-1	2000	705 km	NO	NO	NO	NO	YES	30 m
LANDSAT 7	1999	705 km	YES	NO	YES	YES	NO	Panchromatic resolution: 15 m MSI resolution: 30 m
LANDSAT 8	2013	705 km	YES	NO	YES	YES	NO	Panchromatic resolution: 15 m MSI resolution: 30 m
QuickBird	2001	482 km	NO	NO	YES	YES	NO	2.44 m
Sentinel 1 *	2014	693 km	YES	YES	NO	NO	NO	Depends on the operational mode. The best resolution is for stripmap mode (5 m)
Sentinel 2 *	2015	785 km	YES	NO	NO	YES	NO	Depending on the band, 10 m to 60 m RGB-NIR resolution is 10 m
SPOT-6	2012	694 km	YES	NO	YES	YES	NO	Panchromatic resolution: 1.5 m MSI resolution: 6 m
WorldView-2	2009	770 km	YES	NO	YES	YES	NO	Panchromatic resolution: 0.46 m MSI resolution: 1.84 m
WorldView-3	2014	617 km	YES	NO	YES	YES	NO	Panchromatic resolution: 0.31 m MSI resolution: 1.24 m
PROBA-1	2001	615 km	YES	NO	NO	NO	YES	Visible bands resolution: 15 m Other bands resolution: 30 m

\* Each of the Sentinel-1 and Sentinel-2 missions has a couple of satellites on orbits for better global coverage (up to 2019).

Серед супутників у таблиці 1 зображення з найвищою роздільною здатністю отримують WorldView-3 та WorldView-2, за ними йдуть супутники QuickBird та SPOT-6. Усі ці супутники комерційні, тому їхні зображення дорогі та доступні у відкритому доступі з обмеженим покриттям землі. У літературі мультиспектральні та гіперспектральні зображення з дуже високою роздільною здатністю, які використовуються для виявлення об'єктів, вилучення будівель і доріг, або аналізу врожаю, є в основному повітряними зображеннями,

отриманими цифровими датчиками, такими як AVIRIS і ROSIS. Просторова роздільна здатність зображень таких датчиків може змінюватися в залежності від висоти літака.

Для використання зображень з повітря або космосу зазвичай використовуються контрольовані методи. Такі методи виводять логіку класифікації на основі позначених навчальних даних. Однак чітке позначення даних і збір фактичних даних для таких контрольованих підходів є складним і трудомістким завданням. Небагато доступних баз даних мають базову правду. Набори даних, які найчастіше використовуються в літературі, уже позначені, для класифікації земельного покриття з використанням методів глибокого навчання графічно показано на рисунку 4 і детально в таблиці 2. У деяких із цих наборів даних зображення також належним чином обрізані, виправлені та архівовані в таким чином, який машині легко отримати та обробити.

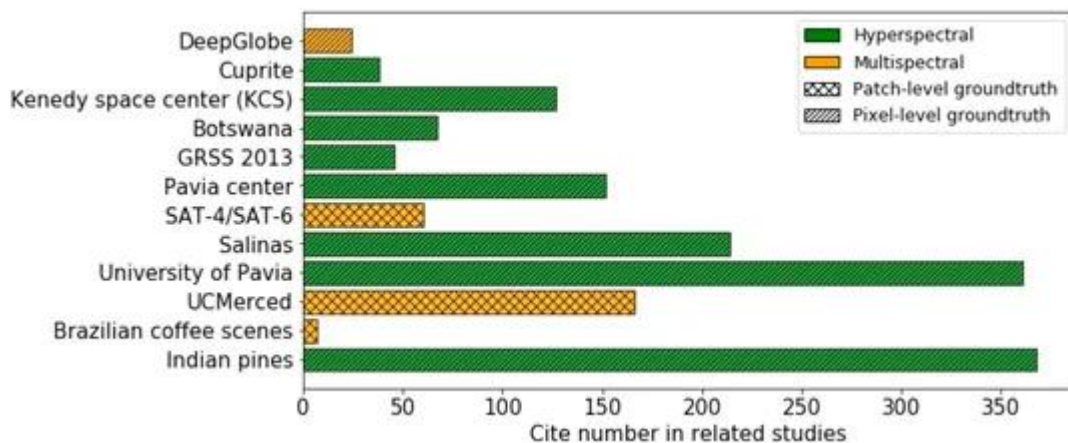


Рисунок 4 – Найпопулярніші набори даних для цілей класифікації земельного покриття з використанням методів глибокого навчання. Цей графік базується на кількості документів, які посилаються на набори даних до травня 2020 року.

Таблиця 2. Зведення найпопулярніших мультиспектральних і гіперспектральних наборів даних.

Table 2. Summary of the most popular multispectral and hyperspectral datasets.

Dataset	Source	Mapping type	Labelling	No. Samples	Image Size (pixel)	Resolution (meter/pixel)	No. Bands	No. Classes	Ref
Botswana	EO-1	Spaceborne	Pixel	377,856 pixels	1476 × 256	30	242	14	
Brazilian coffee scenes	SPOT-5	Spaceborne	Patch	50,004 images	64 × 64	10	3	3	[41]
DeepGlobe	(Mix)	Spaceborne	Pixel	5,836,893,696 pixels	2448 × 2448	0.5	3	7	[42]
Cuprite	AVIRIS	Airborne	Pixel	314,368 pixels	614 × 512	20	224	25	
GRSS 2013	CASI	Airborne	Pixel	15,029 pixels	349 × 1905	2.5	144	15	
Indian pines	AVIRIS	Airborne	Pixel	9234 pixels	145 × 145	20	224	16	
Kennedy space centre (KCS)	AVIRIS	Airborne	Pixel	5250 pixels	614 × 512	18	224	13	
Pavia centre	ROSIS	Airborne	Pixel	103,476 pixels	610 × 340	1.3	102	9	
Salinas	AVIRIS	Airborne	Pixel	54,129 pixels	512 × 217	3.7	224	16	
SAT-4	NAIP program	Airborne	Patch	500,000 images	28 × 28	1	4	4	[43]
SAT-6	NAIP program	Airborne	Patch	405,000 images	28 × 28	1	4	6	[43]
UCMerced	OPLS	Airborne	Patch	2100 images	256 × 256	0.3	4	21	[44]
University of Pavia	ROSIS	Airborne	Pixel	43,923 pixels	610 × 610	1.3	103	9	

Поодинокі зображення GRSS 2013 [46], Космічного центру Кеннеді (KSC) [45], Ботсвани [45] і Cuprite [45] — це набори даних зображень із мітками на рівні пікселів, які використовуються для класифікації земного покриття. DeepGlobe [47] (набір даних земного покриття) — це новий набір даних із мітками на рівні пікселів, представлений у 2018 році для завдання CVPR2018. Він надає величезну кількість піксельних навчальних зразків із високою роздільною здатністю, але містить лише канали RGB. Зображення набору даних DeepGlobe є результатом об'єднання різних комерційних супутникових зображень, але немає точної вказівки на те, які датчики використовуються та як об'єднуються зображення.

Для сегментації зображення використовуються навчальні зразки з мітками рівня пікселів. Таким чином, вищезазначені набори даних зазвичай застосовуються для класифікації пікселів карти та створення сегментованої карти. З іншого боку, існують також деякі набори даних, для яких фрагменти зображень позначаються одним або кількома тегами. Набори даних Sat-4 [48], Sat-6 [48], UCMerced [49] і Brazilian Coffee scenes [50] є одними з найпопулярніших наборів даних з мітками на рівні патча. На додаток до загальноновживаних наборів даних, деякі інструменти надають користувачам доступ до анотованих/напіванотованих баз даних, які зазвичай збираються

шляхом поєднання інформації з різних ресурсів, спрямованих на конкретне використання, наприклад, посіви [51,52], ліси [ 53], або моніторинг водно-болотних угідь [54].

Загалом, майже всі доступні позначені набори даних MSI та HSI мають загальні обмеження щодо застосування контрольованих методів машинного навчання. Ефективне використання контрольованих методів машинного навчання вимагає великої кількості навчальних зразків, які також повинні охоплювати різні варіації в класі. Оскільки маркування таких даних є досить повільним, дорогим і трудомістким, ці набори даних зазвичай обмежені кількістю зразків, не мають різноманітності та надто залежать від конкретного випадку. Такі обмеження в основному називають викликом обмеженої наземної правди, який буде обговорюватися далі в цій статті.

#### **1.4. Машинне навчання для LULC**

Звичайні керовані конвеєри машинного навчання LULC зазвичай включають чотири основні етапи: (1) попередня обробка, (2) розробка функцій, (3) навчання класифікаторів і (4) постобробка (Рисунок 5). Кожен із цих етапів може складатися з набору під завдань. Правильний розподіл усього процесу на підзавдання з чітким формулюванням їхніх припущень допомагає визначити окремі під проблеми, які можна вивчати незалежно та мати рішення або моделі, які можна включити в конвеєр LULC для виконання цільова класифікація/сегментація. Протягом останніх років із зростанням популярності глибокого навчання як дуже потужного інструменту для вирішення різних типів проблем штучного інтелекту ми спостерігаємо сплеск попиту на дослідження для використання методів глибокого навчання для вирішення цих під проблем.

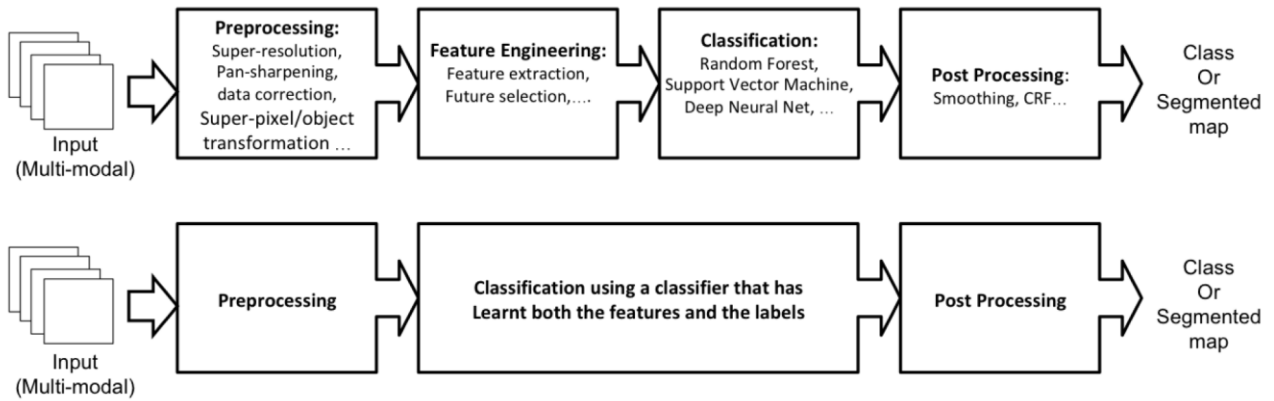


Рисунок 5 – Структури класифікації машинного навчання. Верхній показує загальні кроки звичайних підходів, а нижній показує сучасну наскрізну структуру. У наскрізній структурі глибокого навчання розробка функцій замінюється навчанням функцій як частина етапу навчання класифікатора.

Визначаються конкретні підзавдання, щоб відповідати потребам вищевказаних чотирьох етапів конвеєра машинного навчання; зазвичай етап попередньої обробки включає підзавдання, в рамках яких вхідні дані готуються для наступних етапів (тобто розробка функцій і навчання класифікатора). Підготовка може вимагати виправлення, усунення шумів, синхронізації або об'єднання даних, щоб отримати покращену версію вихідних вхідних даних і покращити продуктивність усього процесу. Фаза розробки ознак зазвичай називається набором завдань вилучення, вибору та перетворення ознак, щоб видалити надлишкову інформацію з оброблених вхідних даних, зменшити її розмірність і визначити набір хороших представлень (функцій) для вхідних даних на основі на основі яких машина може побудувати модель для прогнозування цільових класів. Серцем робочого процесу є навчання класифікатора, де машина будує математичну модель на основі навчальних зразків і розуміє кореляцію між функціями/представленням даних навчання та їх попередньо визначеними класами. Модель після навчання, тестування та перевірки використовується для прогнозування та класифікації нових даних. Нарешті, фаза постобробки в класифікації на рівні пікселів зазвичай являє собою набір методів, що застосовуються для покращення кінцевого сегментованого



зображення шляхом підкреслення морфологічних властивостей класів або об'єктів.

Зі збільшенням обчислювальної потужності в новому поколінні процесорів, протягом останнього десятиліття, наскрізний підхід глибокого навчання привернув велику увагу вчених. Наскрізний навчальний конвеєр, у якому вихідні дані використовуються як вхідні дані, а класифікована карта – як вихідні, є сучасною формою перепроєктування робочого процесу процесу, яка використовує переваги методів глибокого навчання для вирішення завдань складні проблеми. У структурі наскрізного глибокого навчання розробка функцій замінюється навчанням функцій як частина фази навчання класифікатора. У цьому випадку, замість визначення внутрішніх етапів фази розробки функцій, наскрізна архітектура узагальнює генерацію моделі, що включає вивчення функцій як її частину. Така вдосконалена здатність глибокого навчання сприяла його застосуванню в багатьох дослідницьких роботах, де добре відомі, готові наскрізні моделі безпосередньо застосовуються до нових даних, таких як дистанційне зондування. Однак існують деякі відкриті проблеми, складності та питання ефективності в наскрізному використанні глибокого навчання в класифікації LULC, що спонукає прийняти новий підхід для дослідження найсучаснішого глибокого навчання. навчання для класифікації LULC.

У наступних розділах зібрано сучасні відомості про використання методів глибокого навчання для класифікації LULC HSI та MSI, розглядаючи їх використання в наскрізному підході або на одній із фаз традиційного підходу, включаючи навчання класифікатора ґрунтового покритву, генерації наземної правди, об'єднання даних, етапи попередньої обробки даних і вихідних даних після обробки. Зокрема:

У Розділі 1.4.1 і Розділі 1.4.2 пояснена властивість наскрізного підходу до вивчення функцій та його обмеження, які спонукають нас розглядати звичайну

модель машинного навчання, включаючи етапи розробки функцій. Потім ми пояснюємо концепцію розробки функцій, її компоненти та загальні методології, а також методи глибокого навчання, які використовуються в літературі для їх досягнення. Ми також обговорюємо важливість визначення простору функцій і його прямий вплив на формування конвеєра процесу.

У розділі 1.4.3 досліджений вибір класифікаторів MSI та HSI для класифікацій LULC та обговорюємо ефективність методів глибокого навчання для цього завдання. Ми також пояснюємо різні типи підходів до глибокого навчання в класифікації MSI та HSI, які використовуються в сучасному світі.

У Розділі 1.4.4 описано, як це впливає на продуктивність моделей глибокого навчання для HSI та MSI, зосереджуючись на загальновідомій проблемі обмеженої реальної інформації. Потім ми повідомляємо про дослідницькі роботи, спрямовані на цю проблему.

У розділі 1.4.5 описано проблему об'єднання даних, з якою стикаються багато сучасних досліджень. Ми пояснюємо основні проблеми, пов'язані з об'єднанням даних, і пояснюємо, як глибоке навчання сприяє їх досягненням.

Нарешті, у розділі 1.4.6 буде описано інші потенційні методи попередньої та постобробки в літературі, які можуть покращити продуктивність класифікації LULC.

### **1.4.1. Наскрізне глибоке навчання**

Як пояснювалося раніше, збільшення обчислювальної потужності популяризувало підходи наскрізного глибокого навчання, за яких замість інженерних функцій функції автоматично вивчаються класифікатором. Іншими словами, у таких підходах градієнтне навчання застосовується до системи в цілому. Наскрізне використання моделей глибокого навчання було дуже популярним у спільноті дистанційного зондування протягом останніх років. Більшість робіт порівнюють продуктивність такої архітектури з класичними техніками, такими як, наприклад, класифікатори Support Vector Machine (SVM) і

Random Forest (RF) [16]. Однак використання глибокого навчання як наскрізного підходу супроводжується деякими складнощами та неефективністю часу обробки.

Одна з ідей ґрунтується на теоремі Волперта «Без безкоштовного обіду» (NFL) [55] (теорему пізніше було розроблено у співпраці з Macready [56]), яка стверджує, що «будь-які два алгоритми оптимізації є еквівалентними, якщо їх продуктивність усереднена за усі можливі проблеми» [56]. Це означає, що не існує єдиного контрольованого алгоритму навчання з набору рівномірно розподілених можливих функцій, який найкраще працює для всіх видів проблем. Ця теорема спростовує ідею узагальненого єдиного алгоритму машинного навчання для всіх типів проблем і даних і підкреслює необхідність перевірки всіх припущень і їх відповідності в нашій конкретній задачі. На практиці такі моделі глибокого навчання показали велику здатність добре узагальнювати, що теоретично незрозуміло і все ще викликає сумніви [57,58,59,60].

Друга відкрита проблема полягає в тому, що для автоматичного генерування ієрархії абстракцій моделі глибокого навчання потребують величезної кількості навчальних зразків, анотованих цільовими класами. У випадку наскрізних підходів для класифікації ґрунтового покриття за HSI та MSI величезна кількість навчальних вибірок повинна добре охоплювати розподіл класів вихідного кінця. Однак, як зазначено в попередньому розділі, через труднощі зі збором наземної правдивості LULC, це піддається проблемі обмеженої кількості навчальних зразків.

Навіть якщо ми зможемо знайти ефективне рішення для збільшення розміру наборів навчальних даних, наприклад, за допомогою неконтрольованого або напівконтрольованого навчання, проблема ефективності обробки залишається. Як обговорювалося в Розділі 1, складність у природі даних дистанційного зондування, як-от мультимодальність, роздільна здатність, висока розмірність, надмірність і шум у даних, роблять моделювання наскрізного робочого процесу ще більш складним і складним для класифікація LULC MSI та

HSI. Чим складнішою стає архітектура моделі, тим складнішою стає проблема навчання. Іншими словами, збільшення складності архітектур глибокого навчання призводить до більш складних проблем оптимізації та різко знижує ефективність обчислень.

Таким чином, незважаючи на значні спроби застосувати наскрізне глибоке навчання в проблемах класифікації LULC, проблеми такої структури відкривають поле для альтернативних підходів і роблять колишню чотириступеневу конвеєрну структуру машинного навчання дискусійним кандидатом. Дійсно, визначення процесу відповідно до звичайного формату робочого процесу полегшує формування, налаштування та адаптацію системи для задоволення цільових потреб і, в той же час, це зменшує складність оптимізації моделі та обчислювальний час процесу навчання. Розбиваючи припущення, потреби та цілі на набір підзавдань, емпіричний процес вибору ефективного алгоритму для кожного підзавдання стає легшим і легшим для діагностики. Дійсно, ми можемо використовувати методи глибокого навчання більш ефективно та прозоро для виконання окремих підзавдань класичного конвеєра машинного навчання з меншими проблемами для вирішення. Усі рішення та навчені моделі для кожного підзавдання можна потім використовувати в паралельних потоках або в послідовному порядку на різних етапах звичайного робочого процесу. Наприклад, автори в [61] пропонують модель, яка розглядає проблему вибору ознак як проблему реконструкції ознак за допомогою мережі глибоких переконань і порівнює її ефективність у часі з глибокою наскрізною моделлю CNN. Або для вирішення проблеми дефіциту базової істини робота в [62] пропонує використання глибокого навчання в напівконтрольованій генеративній структурі, яка може мати справу з вилученням ознак із невеликої кількості зразків.

#### **1.4.2. Розробка функцій**

Розробка функцій — це один із кроків у звичайному конвеєрі (pipeline) машинного навчання LULC перед навчанням класифікатора, який має справу з визначенням функцій (або представлень), які краще відповідають вимогам класифікатора. Ознаки — це вимірювання або характеристики, на основі яких будується модель, а процес отримання нової абстрактної ознаки на основі наданих даних широко називається розробкою функцій» [63]. Розробка функцій має на меті зменшити розмір вхідних даних і перетворити їх на набір представлень, які несуть лише відповідну значущу інформацію. Побудова моделі на великих необроблених наборах даних із великою кількістю атрибутів на дані, можливо, з деякими надмірностями, обчислювально дорога та неефективна. Тому перетворення необроблених даних у керований набір значущих представлень є дуже важливим для ефективної побудови моделі. Зазвичай різні форми розробки функцій називаються вибором функцій, перетворенням функцій і вилученням функцій.

Вибір ознак і перетворення ознак зазвичай називають методами зменшення розмірності. Зокрема, метою вибору функції є видалення нерелевантної або надлишкової інформації з даних, можливо, без зміни решти інформації в даних. З іншого боку, перетворення функцій відображає вхідні дані в альтернативний простір, щоб полегшити процес. Вибір і перетворення функцій може виконуватися вручну на основі попередніх знань експерта або автоматизовано за допомогою методів машинного навчання.

Вилучення ознак в основному використовується для зменшення кількості функцій даних шляхом створення нового набору функцій із існуючих. У класичних підходах до машинного навчання завдання вилучення ознак, яке також називають функціями ручного створення, обчислює набір нових представлень за допомогою попередньо визначених алгоритмів. Завдяки глибокому навчанню вилучення функцій також може проводитися автоматично, не займаючись складністю розробки та формулювання відповідних алгоритмів.

HSI і MSI є кубічними типами даних [64], які містять два просторових

виміри (ширина і висота каналів) і один спектральний вимір (кількість каналів). Просторова область містить морфологічну інформацію, а спектральна призначена для розрізнення матеріалу, який відповідає пікселю на землі. Індксація даних у часовому порядку додає ще один вимір до простору даних, і це супроводжується проблемами часових рядів. Перенесення такого складного 3- або 4-вимірного простору в простір функцій із відповідною інформацією є дуже критичним. Розмір простору ознак визначається на основі взаємозв'язку між просторовими, спектральними та часовими аспектами даних. В одних роботах усі ці аспекти вважаються самостійними, а в інших частково або повністю залежними. Попереднє припущення про залежність або незалежність таких функцій відіграє вирішальну роль у проектуванні конвеєра машинного навчання, виборі класифікатора та етапах розробки функцій.

Розробка функцій дуже складна для даних HSI. Є три проблеми, які слід розглянути: (1) велика кількість спектральних смуг призводить до проблеми високої розмірності, так званого прокляття розмірності [65]: з обмеженими навчальними вибірками це означає, що велика частина гіперспектрального простору даних є порожнім, тобто не має жодного спостереження, на основі якого можна побудувати модель; (2) кореляція між спектральними смугами не обов'язково є лінійною; і (3) подібність між деякими спектральними смугами вказує на високу спектральну міжсмугову надмірність таким чином, що зменшення деяких спектральних смуг не викликає значної втрати інформації. Тому, щоб витягти правильні представлення зі спектральної області даних HSI, перетворення ознак і вибір ознак слід розглядати разом із виділенням ознак. Таким чином можна зменшити розмірність і видалити надлишковість, що допомагає перевести дані в керовані та доступні для вивчення представлення. Робота в [66] глибоко обговорює відмінності в техніках та інструментах для реалізації вилучення ознак HSI.

Розробка функцій даних має високу цінність, коли кількість навчальних вибірок не задовольняє вимогам наскрізного навчання або коли навчання

представлень за допомогою наскрізного підходу неефективне з точки зору обчислень. Тим не менш, етап розробки функцій все ще може отримати користь від глибокого навчання та інших методів машинного навчання, щоб знайти хороші представлення даних. У наступних підрозділах ми пояснюємо вибір, перетворення та вилучення функцій для даних HSI та MSI, а також обговорюємо загальні методи машинного навчання, включаючи методи глибокого навчання, які використовуються для вирішення цих завдань.

#### **1.4.2.1. Вибір і трансформація функції**

Вибір ознак і перетворення ознак даних здебільшого називають методами зменшення розмірності. Вибір функцій має на меті усунення надмірності шляхом вибору відповідних атрибутів даних, тоді як перетворення функцій відображає дані в інший простіший простір із, можливо, меншим розміром. Хоча вибір і перетворення даних у набір релевантних керованих представлень, які сумісні з вимогами класифікатора, може значно покращити продуктивність машини, зменшити можливість переобладнання та скоротити час навчання, надмірне скорочення інформації також може піти навпаки. Тому трансформація та вибір функцій досить складні та чутливі.

Найпоширенішим алгоритмом зменшення розмірності, який використовується для даних HSI, є аналіз основних компонентів (PCA) [67,68,69]. PCA проектує дані в новий простір, у якому розміри є лінійно незалежними (ортогональними), і вони ранжуються таким чином, що головна вісь є тією, на якій дані більш розповсюджені [70]. Таким чином, PCA перетворює дані в простіший простір для аналізу, вирішуючи трансформацію ознак, щоб зменшити розмірність ознак, на основі яких будується модель.

Вибір функцій, який також називають очищенням даних або фільтрацією даних, усуває надлишки даних і зберігає найбільш релевантні атрибути для створення набору функцій для побудови моделі. Це зменшує ймовірність переобладнання та час навчання, і врешті-решт покращує точність класифікації.

Майже в усіх методах стохастичного навчання важливість ознак обчислюється автоматично на етапі навчання класифікатора. Рейтинг важливості функції показує важливість атрибутів вхідних даних, тому стає зрозумілим, які атрибути вхідних даних потенційно можна видалити. Вибір функцій для складних даних HSI та MSI з величезною кількістю атрибутів є подвійним; З класичними класифікаторами машинного навчання, такими як SVM, вибір функції є вирішальним, оскільки визначення гіперпараметрів для масивних типів вхідних даних є надто складним і непрактичним [71,72]. Однак із сучасними класифікаторами, розробленими для уникнення проблеми переобладнання, необхідність скорочення інформації з вхідних даних викликає сумнів [73,74].

Використання глибокого навчання на етапі розробки функцій в основному називають вилученням функцій. Модель глибокого навчання розробки функцій вивчає, як оптимально перетворити простір введення в менший закодований простір, який включає всю важливу інформацію. Зазвичай важливу інформацію називають закодованими ознаками, яких достатньо для реконструкції вхідних даних. У наступному підрозділі обговорюються методології вилучення ознак, засновані на методах глибокого навчання.

#### **1.4.2.2. Вилучення функцій**

Вилучення функцій визначає новий набір уявлень або абстракцій для даних на основі всіх наявних у них атрибутів, щоб полегшити процес навчання для машини. Хороший набір представлень містить всю відповідну інформацію, яка відповідає вимогам класифікації. Такі представлення можна створювати вручну, використовуючи алгоритми, які обчислюють новий набір ознак. Наприклад, добре відомий NDVI є простим прикладом функцій ручної роботи, просто поєднуючи NIR (ближній інфрачервоний діапазон) і червоні смуги зображення, і є дуже інформативним для виявлення рослинності на земному покриві. посилання [13,75] використовують маски NDVI та інші індекси, щоб



керувати моделлю на основі згорткової нейронної мережі (CNN) (техніка пояснюється в розділі 4.3) для виявлення рослинності, води та інших елементів, які підсвічуються цими масками. Функції ручної роботи також можна отримати за допомогою методів обробки зображень, таких як виявлення країв, згладжування або сегментація.

Методи машинного навчання без нагляду, напівконтрольованого та контрольованого машинного навчання також можуть витягувати відповідні функції для класифікатора. Найвідомішими методами машинного навчання без нагляду для автоматичного вилучення функцій є методи глибокого навчання Autoencoder (AE) (техніка пояснюється в розділі 4.3). За останні кілька років алгоритми AE стали дуже зручними та популярними для отримання оптимізованої абстракції даних HSI та MSI для класифікатора [76,77,78]. Хоча такі неконтрольовані алгоритми можуть знаходити представлення даних без будь-яких підказок чи міток, посилення. [79] підкреслюють переваги використання керованих алгоритмів, вказуючи на те, що в даних HSI необхідно досліджувати не лише глобальну взаємну інформацію, але й дискримінаційні прогнози в класі. Контрольовані алгоритми, які використовують мічені зразки, можуть вивчати метрики, які зберігають точки даних у класі разом і відокремлюють їх від інших класів [80]. Оскільки підготовка мічених даних для контрольованих методів досить трудомістка, звичайні контрольовані алгоритми можна розширити до напівконтрольованих варіантів [81]. Основні керовані/напівконтрольовані алгоритми зменшення розмірності, що застосовуються до даних HSI, базуються на різних типах дискримінантного аналізу, наприклад, лінійному дискримінантному аналізу (LDA), стохастичному дискримінантному аналізу (SDA) і локальному дискримінантному аналізу Фішера (LFDA), посилення . [82,83,84] і локальне дискримінантне вбудовування (LDE) і збалансоване локальне дискримінантне вбудовування (BLDE) [80,85].

### **1.4.3. Класифікатор**

Незважаючи на зростаючу популярність глибоких нейронних мереж, класичні керовані класифікатори все ще популярні в спільноті дистанційного зондування. RF і SVM є найпоширенішими класичними класифікаторами, які використовуються в літературі для класифікації земельного покриття даних дистанційного зондування. Як і інші непараметричні контрольовані класифікатори, ці алгоритми не роблять жодних припущень щодо розподілу даних, і вони показали багатообіцяючі результати в класифікації даних дистанційного зондування, випереджаючи попередні класифікатори, прийняті в галузі, такі як лінійна регресія (LG), максимальна ймовірність (MLC), K Nearest Neighbor (KNN) і Classification and Regression Tree (CART).

RF — це ансамблевий класифікатор, створений із набору деревовидних предикторів (CART), таким чином, що кожне дерево залежить від випадкового набору навчальних спостережень, які незалежно відбираються із заміною [86], а в кожному вузлі розщеплення дерев — підмножина властивостей вибирається випадковим чином для вирощування дерева [87]. Радіочастота є досить популярною для класифікації даних дистанційного зондування через її простоту та потужність у досягненні надійних моделей. Він широко використовувався для класифікації ґрунтового покриття [88,89,90] та багатьох інших застосувань, про що повідомляється в [91]. Однак, як і більшість контрольованих класифікаторів, RF вимагає достатньо великого набору довідкових даних для вивчення розподілу класів, що часто є критичною проблемою.

SVM — ще один популярний класифікатор для даних дистанційного зондування, який добре працює з відносно невеликою кількістю навчальних вибірок. Алгоритм спрямований на пошук оптимальної розділової гіперплощини, яка розділяє спостереження на цільові класи так, щоб межі між класами мінімізували частоту неправильної класифікації [92]. Параметр регуляризації в SVM відіграє вирішальну роль у його продуктивності; з добре налаштованими регуляризаціями, SVM, як правило, стійкі до переобладнання та не мають жодних внутрішніх проблем, коли кількість спостережень менша за кількість атрибутів [93,94]. Спираючись на такі характеристики, SVM був дуже

популярним для класифікації земельного покриття MSI та HSI [95,96,97].

Однак, коли справа доходить до складних проблем, таких як класифікація зображень HSI, підходи глибокого навчання з можливістю вивчення ієрархій ознак перевершують інші класифікатори. Моделі глибокого навчання складаються з кількох рівнів, так що кожен рівень обчислює нове представлення даних із представлення на попередніх рівнях штучних нейронів, створюючи ієрархію абстракцій даних [98]. CNN — це група методів глибокого навчання, які складаються з шарів згортки та об'єднання, які зазвичай завершуються повністю пов'язаним шаром нейронної мережі та належною функцією активації, тобто в моделях, які безпосередньо реконструюють прогноз вихідного зображення, наприклад U-Net і генеративних моделей (пояснено пізніше), повністю підключена мережа та функція активації не потрібні. CNN, які дуже успішно класифікують складні контекстні зображення, також широко використовуються для класифікації даних дистанційного зондування.

CNN — це нейронні мережі прямого зв'язку (штучні нейронні мережі, у яких зв'язки між вузлами/нейронами не утворюють цикл), які розроблені для обробки типів даних, що складаються з кількох масивів (наприклад, зображень, які містять шари 2D-масиву пікселів). [98]. Кожен CNN, як показано на рисунку 6, містить кілька етапів згортання та об'єднання, створюючи ієрархію залежних карт ознак. Приклад на рисунку показує згорточні нейронні мережі з двома шарами згортки та двома шарами об'єднання для (а) класифікації на рівні патча, (б) класифікації на рівні пікселя та (в) реконструктивної моделі зображення. У (а) і (б) повністю зв'язана мережа подається з плоскими картами функцій останнього шару об'єднання. У (б) центральна частина, показана червоним кольором, це піксель, якому призначено клас. У (в) модель не включає жодної повністю підключеної мережі та функції активації, але права половина моделі безпосередньо реконструює прогноз вихідного зображення.

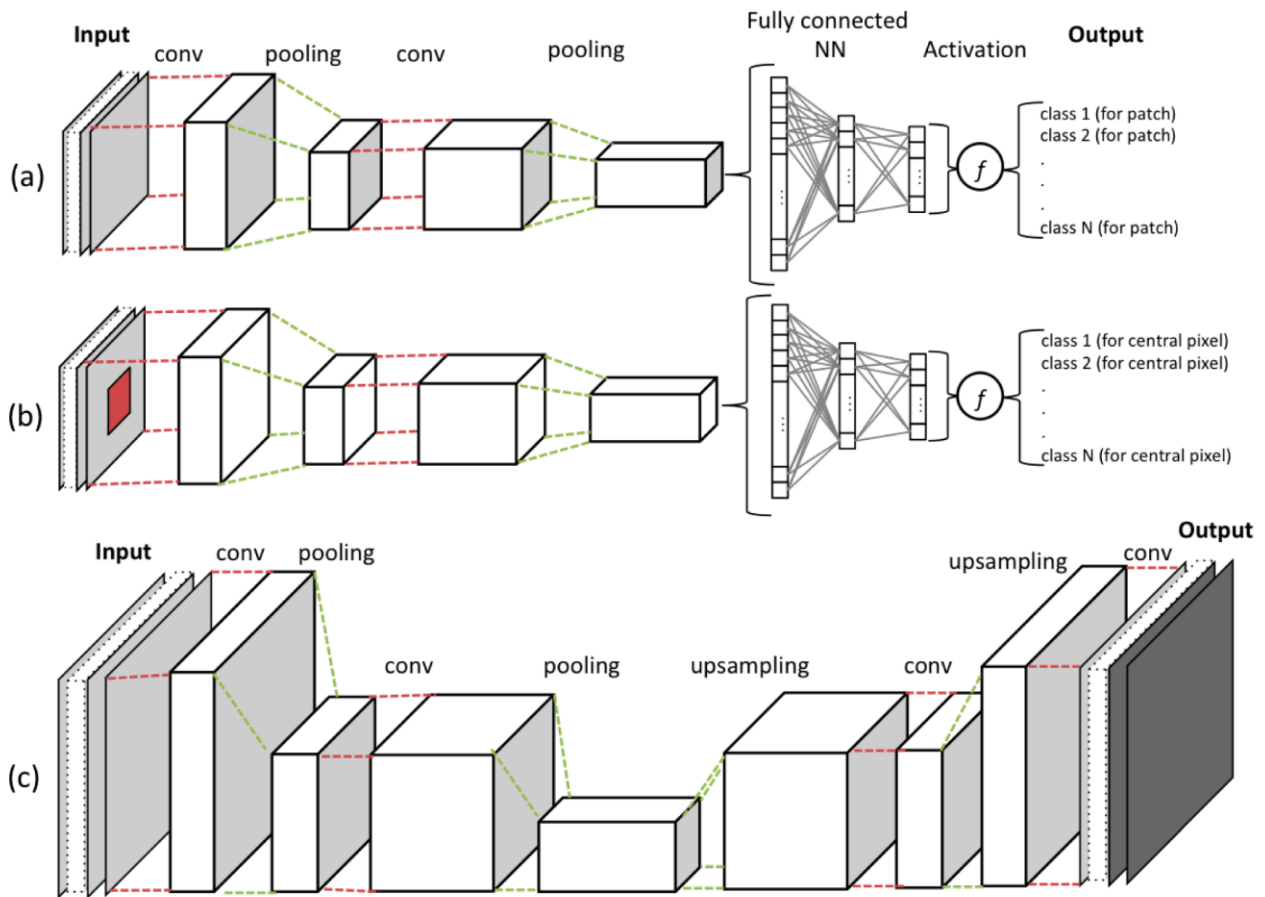


Рисунок 6 – Приклад згорткової нейронної мережі з двома шарами згортки та двома шарами об'єднання для (а) класифікації на рівні патча, (б) класифікації на рівні пікселя та (в) реконструктивної моделі зображення. Отримані куби після кожного шару згортки та об'єднання називаються картами функцій.

На кожному шарі згортки карти функцій обчислюються як зважена сума попереднього рівня патчів функцій, використовуючи фільтр зі стеком ядер фіксованого розміру, а потім результат передається в нелінійність, використовуючи функцію активації (наприклад, ReLU). Таким чином, вони виявляють локальні кореляції (встановлені в розмірі ядра), зберігаючи інваріантність до розташування в масиві вхідних даних. Рівень об'єднання використовується для зменшення розмірності отриманої карти функцій шляхом обчислення максимуму або середнього значення сусідніх одиниць для створення незмінності до масштабування, невеликих зсувів і спотворень. Зрештою, етапи

згортання та об'єднання шарів завершуються повністю пов'язаною нейронною мережею та функцією активації, які відповідають за завдання класифікації в мережі.

Процес навчання моделі CNN за допомогою набору навчальних зразків знаходить оптимізовані значення для параметрів моделі, які можна вивчати, шляхом зменшення вартості, обчисленої за допомогою функції втрат (наприклад, мінімальна квадратична помилка, крос-ентропія або шарнірні втрати). У CNN параметри, які можна вивчати, — це вагові коефіцієнти, пов'язані як з фільтрами шару згортки, так і з'єднаннями між нейронами в повноцінній нейронній мережі. Таким чином, мета оптимізатора (наприклад, Stochastic Gradient Descent, RMSprop або Adam) полягає не лише в навчанні класифікатора, але він також відповідає за вивчення функцій даних шляхом оптимізації параметрів шарів згортки.

Розмір і розмірність фільтрів для кожного шару згортки є так званими гіперпараметрами моделі. Хоча вибір розміру ядра для фільтрів зазвичай є індуктивним процесом, розмірність фільтрів може бути напряму керована попередніми знаннями про вхідний простір (наприклад, часовий ряд, одноканальне зображення, багатоканальне зображення або часовий ряд багатоканальних зображень) і над очікуваннями щодо типу ознак, які потрібно виділити (наприклад, просторові, спектральні, просторово-спектральні або просторово-спектрально-часові характеристики). CNN, які використовуються в літературі для класифікації даних дистанційного зондування, можна розділити на три підтипи: CNN з одновимірними фільтрами (1D-CNN), CNN з двовимірними фільтрами (2D-CNN) і CNN з тривимірними фільтрами. Фільтри (3D-CNN), показані на рисунку 7. Відмінності у згаданих підтипах CNN полягають у шарах згортки. Ці мережі можуть використовуватися спільно в паралельних потоках для отримання різних незалежних функцій.

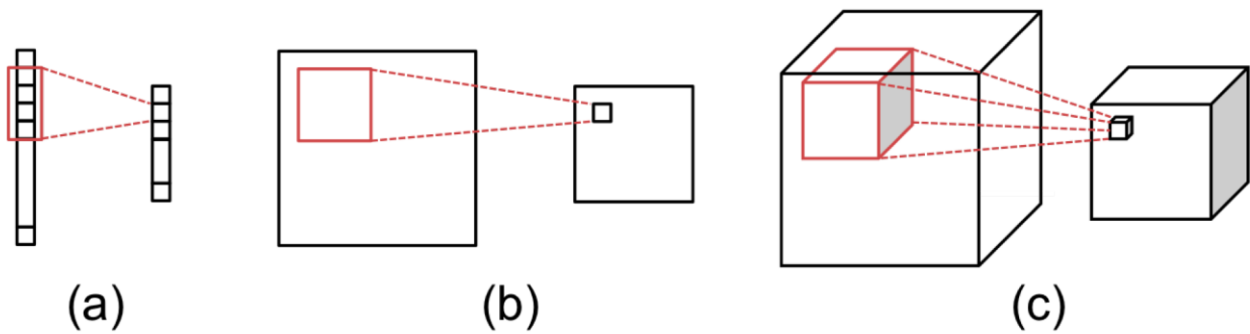


Рисунок 7 – Ілюстрація різних операцій згортки: (a) 1D згортка (з 1D фільтром) (b) 2D згортка (з 2D фільтром) і (c) 3D згортка (з 3D фільтром). Для кожного із зображень ліва частина є входом згортки, а права – виходом. Фільтр показано червоним.

Одновимірні CNN, які в основному використовуються для моделювання часових рядів, також використовувалися для вилучення спектральних ознак пікселів у даних HSI [99,100,101]. Цей метод іноді називають класифікацією спектральних кривих [101]. Накладання спектральних шарів HSI, що відповідають трьом сезонам один на одного, ref. [102] застосували 1D-CNN, щоб також розрізнити характеристику сезонної зміни в спектрально-часовій області. Посилання [103] пропонує гібридну модель 1D-CNN і RNN, яка вивчає спектральні залежності автоматично. Зокрема, він складається з шарів згортки та об'єднання (для виділення локально-інваріантних ознак), за якими йдуть повторювані шари (для отримання спектрально-контекстуальної інформації з останніх вилучених функцій), і завершується повністю пов'язаною нейронною мережею та функція активації. Посилання [104] використовує 1D-CNN у генеративній змагальній мережі (1D-GAN) для створення фальшивих спектральних даних, а також як дискримінатор для класифікації спектральних характеристик.

Двовимірні CNN є поширеним типом CNN, який використовується для класифікації зображень, де існує кореляція між морфологічними деталями та цільовими класами. У дистанційному зондуванні 2D-CNN зазвичай використовується для виділення просторових характеристик HSI та MSI,

враховуючи безперервність земного покриття в просторовій області [105]. Добре відомі попередньо визначені моделі CNN, розроблені для розуміння зображення, є іноді використовуються в літературі для класифікації земельних покриттів, включаючи моделі LeNet5 [106], AlexNet [107], VGGNet [108], CaffeNet [109], GoogLeNet [110] і ResNet [111]. У [8,112] автори порівняли згадані моделі в контексті класифікації ґрунтового покриття HSI. Загалом, оскільки зв'язок між спектральними діапазонами HSI не є лінійним, 2D-CNN зазвичай використовуються спільно з 1D-CNN для покриття спектрально-просторової області характеристик даних HSI [99]. У таких випадках моделі витягують просторові та спектральні характеристики окремо паралельно, потім їх витягнуті характеристики зазвичай збираються разом і подаються до повністю підключеного класифікатора з наступною функцією активації.

Однак, оскільки об'єднання виділених ознак у таку структуру є додатковим емпіричним процесом, точне налаштування моделі стає ще складнішим [113]. Тривимірний CNN є альтернативним підходом, який може зменшити цю складність, просто залишивши функції як тензори в 3D-просторі, враховуючи також потенційні кореляції між просторовими та спектральними аспектами даних.

Тривимірні CNN здебільшого використовуються для класифікації багатоканальних зображень, у якій часовий вимір додається до області (просторово-часова класифікація). У випадку дистанційного зондування 3D CNN використовуються для виділення спектрально-просторових [114,115] і просторово-часових [113] характеристик. У таких класифікаціях ознаки розглядаються як тензори в 3D-доменах, і кожен шар об'єднання згорток впливає на розмір об'єму ознак у глибину, ширину та довжину. Автори в [73], зосереджуючись на повному використанні спектральної та просторової інформації у вхідних даних HSI, пропонують наскрізну модель, яка містить чотири послідовні залишкові блоки з 3D CNN для вилучення спектральних і просторових характеристик відповідно. Завдяки циклу навчання-перевірки в запропонованій моделі та зміні параметрів CNN, ознаки даних HSI вивчаються.

Автори [116] представляють структуру мережі уваги для класифікації гіперспектральних зображень, яка включає просторові, спектральні та концентраційні модулі на основі 3D-CNN; останній призначений для вилучення дискримінаційних ознак із зон уваги кубів HSI.

Зазвичай рівні згортки та об'єднання застосовують лінійні операції, що передбачають множення набору вагових коефіцієнтів на вхідні дані для створення вхідних представлень. Однак хороші представлення, як правило, є дуже нелінійними функціями вхідних даних, як зазначено в [117], і моделювання такої складності за допомогою традиційної стратегії відображення функцій згортки вимагає дуже глибокого вивчення стека згортки та об'єднання, що схильне до проблеми переобладнання та обчислювальної неефективності. Щоб вирішити їх, автори [117] вводять концепцію мережі в мережевих структурах (NiN) або початкових мереж, які можуть замінити фільтри лінійної згортки на «мікромережі» для роботи з нелінійними наближеннями. Початкова мережа використовує фільтри  $1 \times 1$ , які зменшують складність 3D-CNN за рахунок зменшення обчислювальної вартості та кількості вихідних функцій. Посилання [118] використовує цю ідею для реалізації взаємодії спектральної інформації та інтеграції певних діапазонів у дані MSI. Модель GoogLeNet із дев'ятьма початковими модулями також широко використовувалася в літературі для класифікації MSI та HSI [119,120,121,122].

Однією з головних проблем глибокого навчання є проблема переобладнання. Було доведено, що залишкові блоки, введені мережею ResNet [111], є гарною заміною для звичайних блоків згортки та об'єднання, щоб уникнути цієї проблеми переобладнання.

Залишкові блоки (рисунок 8) являють собою мережі, що складаються з звивин і шарів об'єднання з пропускаючими з'єднаннями. З'єднання пропуску (або ідентичності) надає процесу навчання можливість просто пропускати шари згортки та об'єднання, якщо вони не потрібні. У деяких моделях залишкові блоки використовуються в налаштованій мережі [73,74,123], а в багатьох інших добре відомі моделі ResNet безпосередньо використовуються для виконання



класифікації земельного покриття MSI та HSI [120,121,122,124,125].

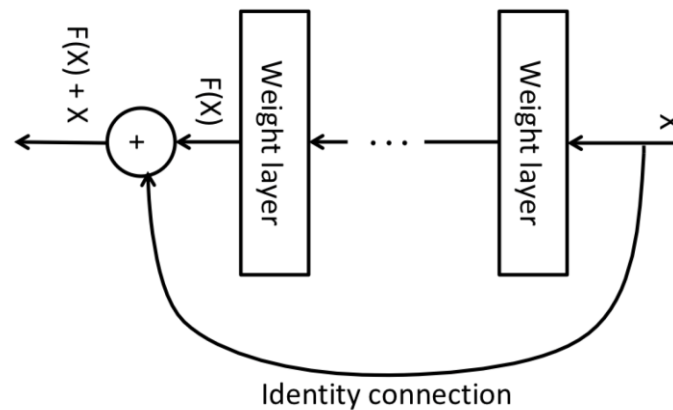


Рисунок 8 – Загальна схема залишкового блоку з підключенням пропуску або ідентичності. Пропуск з'єднання дозволяє процесу навчання обходити вивчення параметрів внутрішніх вагових шарів (згортка з/без об'єднання).

Для виведення сегментованої карти деякі роботи пропонують використовувати U-Net, який спочатку був представлений [126] для сегментації біомедичних зображень. U-Net (рис. 9) складається з трьох етапів: (1) згортання за допомогою згорткових шарів і максимального об'єднання, (2) вузьке місце з парою згорткових шарів і випадання, і (3) розширення з деякими деконволюційними (або транспонування згортки) для збільшення вибірки, згорткових шарів і конкатенацій карт функцій. Архітектура U-Net, як також зображено на рисунку 9, виглядає як «U», від якої й походить назва. Шлях скорочення поводить як кодер, намагаючись знайти приховані представлення або закодовані значення для вхідних даних. Частина розширення поводить як декодер, відновлюючи інформацію. Оскільки на шляху згортання позиційна інформація втрачається, для точного відновлення інформації на кожному кроці розширення використовуються з'єднання пропуску для передачі копії відповідної закодованої карти ознак із шляху згортання. Ці копії закодованих карт функцій з'єднуються з результатом деконволюції, щоб змусити модель вивчати більш точні результати. У контексті дистанційного зондування U-Net показав багатообіцяючі результати у вилученні будівель [127,128], доріг [129,130], хмар [131,132] і класифікації інших наземних покриттів [133,134,135]

за допомогою даних MSI високої роздільної здатності.

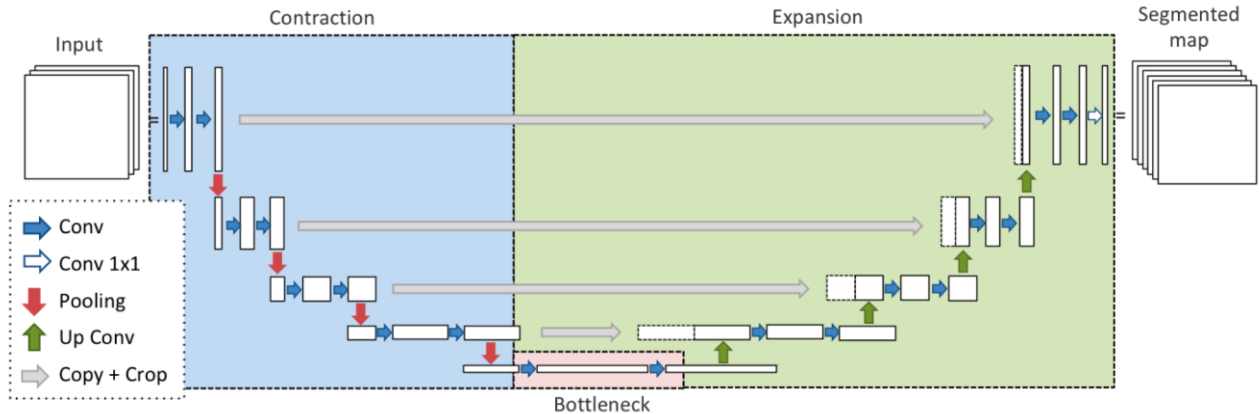


Рисунок 9 – Модель U-Net для семантичної сегментації. Модель складається з трьох кроків: скорочення із згортковими шарами та максимальним об'єднанням, вузьке місце з кількома згортковими шарами та випаданням, а також розширення з деконволюційними та згортковими шарами та конкатенаціями карт функцій.

Як останнє зауваження, процес вивчення суттєвих параметрів згорток і деконволюцій у складних архітектурах пов'язаний з важливою проблемою: вибір життєздатного оптимізатора з ефективною обчислювальною складністю та його відповідною функцією вартості, яка може це оцінити. Автори [136] детально обговорюють методи оптимізації проти втрачених функцій і пояснюють потенційні проблеми та їх обчислювальну складність.

#### 1.4.4. Виклик обмеженої ґрунтовної правди

Як пояснювалося раніше, для того, щоб глибоке навчання перевершило інші підходи, потрібна велика кількість навчальних даних із базовою правдою. Ось чому іноді класичні методи машинного навчання, такі як SVM, показують кращу або порівнянну продуктивність у класифікації LULC MSI та HSI. Як приклад, автори [137] оцінюють продуктивність Sparse Auto-Encoder (SAE) і SVM у класифікації популярних наборів даних, дійшовши висновку, що у загальній ситуації обмеженої кількості зразків SVM з меншою кількістю параметрів для вивчення, а не лише працює краще, ніж SAE, але також вимагає

розумнішого часу обчислення.

Щоб вирішити вищезазначену проблему, ref. [138] пропонують підхід до розширення даних, який використовує перетворення зображення (наприклад, перевертання, переклад і обертання) для створення додаткових і більш різноманітних зразків даних на основі вихідних даних, що покращує продуктивність його моделі CNN (рис. 10). Альтернативний підхід полягає у використанні напівконтрольованих методів навчання, які використовують немарковані дані. Одним із способів є використання методів самомаркування за допомогою попередньо навченого класифікатора маркування [139], а іншим недавнім способом є використання генеративних змагальних мереж (GAN), включаючи генеративні моделі разом із методами дискримінаційної оцінки [62] (показано на рисунку 11).

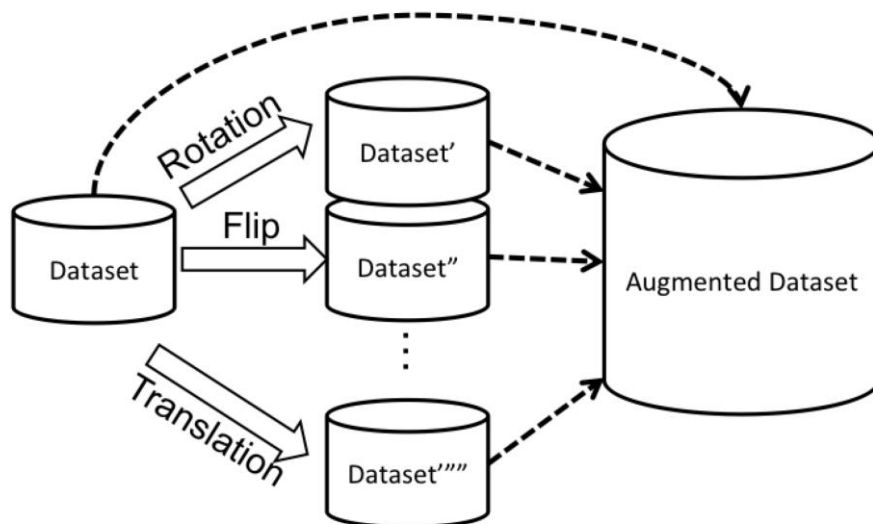


Рисунок 10 – Підхід до збільшення даних для збільшення навчального набору даних (основна правда). Доповнений набір даних складається з оригінального набору даних разом із його повернутими, перевернутими або перекладеними версіями.

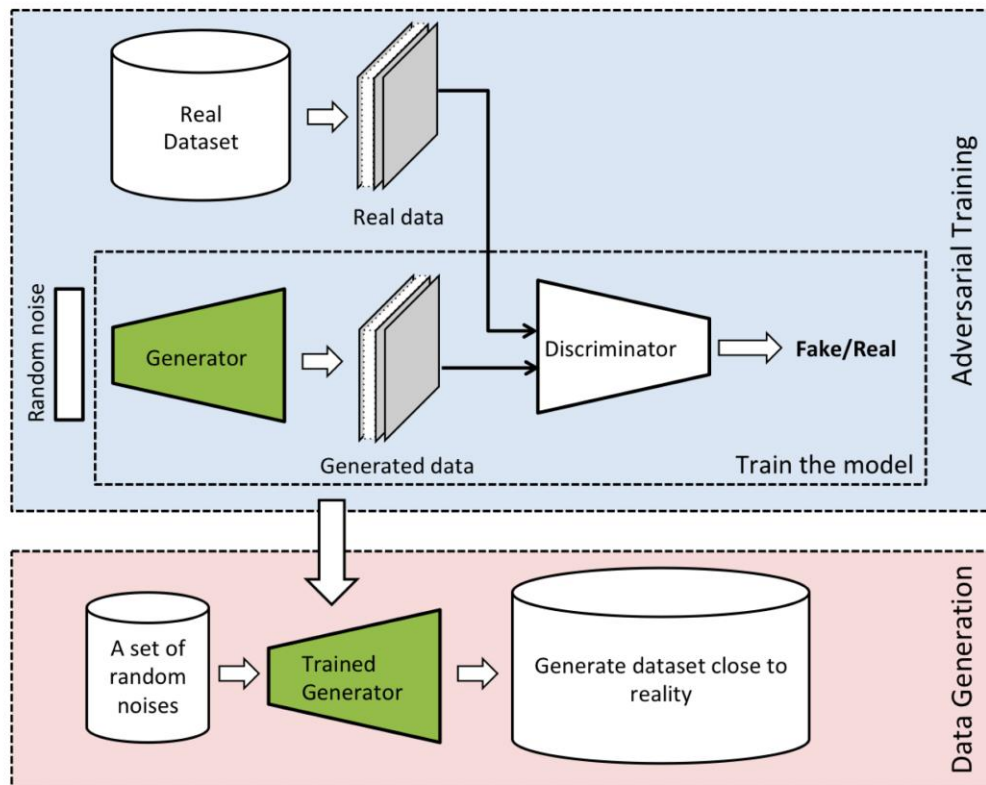


Рисунок 11 – Загальна схема генеративної змагальної мережі (GAN), що зображує, як генеруючу модель навчають і як навчений генератор використовується для створення базової правди.

Трансферне навчання є ще одним підходом, запропонованим для вирішення проблеми обмеженої базової правди. Методологія навчання передачі використовує попередньо навчений класифікатор для вилучення початкового набору уявлень для нового набору даних (рис. 12). Відповідно до [140], за допомогою трансферного навчання модель може очікувати вищий старт, вищий нахил і вищу асимптотичну продуктивність під час процесу навчання.

Посилання [141, 142] використовують класифікатор, попередньо навчений на наборі даних ImageNet, для передачі знань у проблему класифікації земельного покриття. Іншим прикладом є методологія, запропонована [143], яка попередньо навчає класифікатор на наборах даних із викликів VOC і PASCAL, який потім використовується для отримання початкових представлень зображень GoogleMap для виявлення об'єктів дистанційного зондування. Посилання [144] пропонує модель, яка базується на ідеї поєднання трансферного

навчання та напів контрольованих методів, які можуть впоратися з проблемою обмеженої реальної інформації. У цій методології попередньо навчена модель на позначеному мультимодальному наборі даних (MSI-HSI або SAR-HSI) використовується для позначення одноmodalного набору даних (тільки MSI або лише SAR).

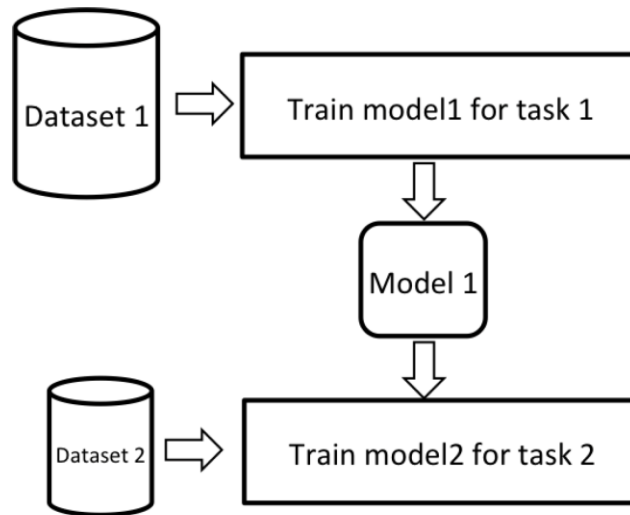


Рисунок 12 – Підхід до навчання передачі: попередньо навчена модель на іншому наборі даних використовується як відправна точка для вилучення початкових представлень з іншого (меншого) набору даних.

Іншим підходом до вирішення проблеми нестачі мічених даних є навчання без нагляду. Наприклад, без будь-яких позначених даних, робота в [145], натхненна [146], пропонує неконтрольований метод глибокого навчання для сегментації HSI, який спочатку використовує 3D згорткові автокодери (AE) (рис. 13) для вивчення вбудованих функцій і використовує вивчені представлення на рівні кластеризації для сегментації вхідного зображення. AE складається з двох етапів: шляху кодування та шляху декодування. Шлях кодування використовує згорткові шари разом із шарами об'єднання для передачі вхідних даних у простір латентного представлення або кодованих значень. Частина декодера оцінює, наскільки хороші закодовані представлення для відновлення даних, використовуючи підвищення дискретизації та згорткові шари.

Автокодувальники мають на меті отримувати значущу інформацію з даних у неконтрольований спосіб. Незважаючи на те, що ця методологія може істотно полегшити процес генерації наземної правди і може бути корисною для програм високого рівня, таких як виявлення аномалій, навчання цих моделей є дорогим з точки зору обчислень.

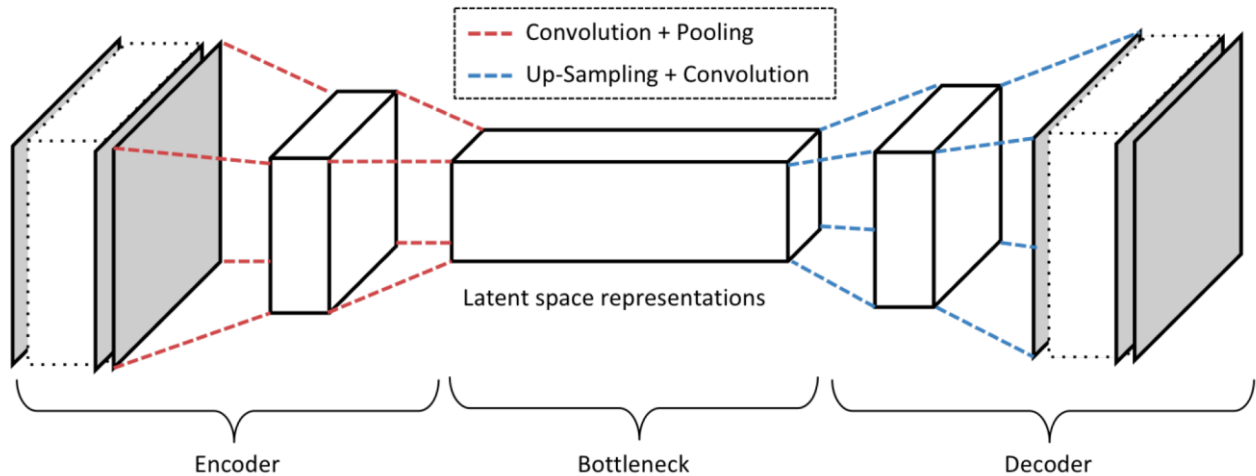


Рисунок 13 – Приклад 3D-автокодувальника з парою шарів згортки, за якими слідує пара шарів об'єднання в кодері та пара шарів підвищення дискретизації, за якими слідує згорткові шари в частині декодера, яка вивчає представлення з немаркованого набору даних. У такій стратегії неконтрольованого навчання процес навчання відбувається для кодування даних у набір уявлень, а декодер оцінює, наскільки уявлення достатньо хороші для реконструкції вихідних даних за допомогою тих самих згорток.

Мічені набори даних не тільки обмежені в кількості, але також дуже обмежені з точки зору різноманітності. Іншими словами, більшості доступних наборів даних із мітками HSI та MSI недостатньо для навчання узагальненої моделі, оскільки вони специфічні для часу та місця. Це спричиняє поширену проблему, коли класифікатор, навчений за допомогою одного набору даних, зазвичай не працює так само добре, як інші набори даних. Дійсно, сезонні зміни ґрунтового покриття, ефекти освітлення та внутрішньокласна мінливість у різних регіонах є факторами, які не враховуються в більшості наборів даних із наземною правдою. Крім того, кожен набір даних має обмежену кількість класів, які

здебільшого є специфічними для контексту, розташування та початкової цілі програми, що ускладнює їх змішування та створення більшого комплексного набору даних.

Створення позначених наборів даних вимагає ручного втручання. Проте, порівняно з наборами даних з мітками на рівні пікселів, набори даних на рівні патча відносно легше підготувати, оскільки маркування менш чутливе до дрібних деталей. EuroSAT, представлений [122] (патчі з багатьма мітками) і набори даних SAT-4 і SAT-6 [43] (патчі з однією міткою), є деякими прикладами наборів даних з мітками на рівні патчів, випущених для спільноти. З іншого боку, маркування піксельного рівня все ще є проблемою, і зазвичай це роблять польові експерти. Підходи краудсорсингу можуть значно полегшити створення наземних істинних карт; Як залучити громадян до виконання мікрозавдань через гейміфікацію та змагання, досліджується [147,148]. Потенційні проблеми та необхідні оцінки при використанні таких підходів також обговорюються в [149].

На додаток до вищезазначених обмежень, ми повинні взяти до уваги, що майже всі доступні набори даних мають фіксовану просторову роздільну здатність. Характеристики сенсора, а також вибір бортового чи космічного датчика безпосередньо впливають на роздільну здатність зображення. Просторова роздільна здатність даних може бути недостатньою або вводити в оману класифікатор залежно від цільових класів. Наприклад, у звичайних моделях зазвичай візуальна помітність [150], тобто селективна якість сприйняття людської зорової та когнітивної системи, яка дозволяє деяким елементам одразу виділятися серед інших у межах сцени, не враховується у виділенні ознак із зображення високої роздільної здатності. Одним із поширених рішень є багатомасштабне навчання: автори [151] пропонують багатомасштабну структуру CNN, у якій піраміда різномасштабованих версій зразка зображення високої роздільної здатності подається на машину для захоплення різної концептуальної інформації. З іншого боку, зображення з низькою роздільною здатністю не мають достатньо деталей для вилучення. Зазвичай, щоб вирішити таку проблему, дані з інших джерел можуть бути введені в конвеєри моделі, щоб

допомогти машині отримувати відповідні функції. Іншими словами, одним із можливих завдань, які можна вирішити шляхом злиття різних типів даних (мультимодальне злиття даних), є покращення роздільної здатності зображень. Цей аспект більш детально пояснюється в наступних підрозділах.

#### **1.4.5. Мультимодальне злиття даних**

Об'єднання даних — це процес об'єднання даних із багатьох джерел для покращення потенційних значень і продуктивності інтерпретації вихідних даних, а також для отримання високоякісного видимого представлення даних [152]. У дистанційному зондуванні злиття даних зазвичай використовується для покращення просторової та спектральної роздільної здатності даних. Хоча злиття даних має довгу історію в спільноті дистанційного зондування, поява машинного навчання і, зокрема, методів глибокого навчання, різко змінила спосіб об'єднання даних.

Початковим кроком для будь-якого об'єднання геоданих є зіставлення геокоординат. Потім, маючи спарені дані з однієї сцени, об'єднання даних може відбуватися на одному з наступних трьох етапів: (1) на етапі підготовки даних, (2) на етапі розробки функцій або (3) на етапі прийняття рішення. (все показано на рисунку 14).

Об'єднання даних на етапі підготовки даних (також називається раннім об'єднанням) (рис. 14а) зазвичай називають перетворенням із високою роздільною здатністю. У цьому процесі метою є підвищення роздільної здатності цільового набору даних за допомогою іншого, іноді тимчасового, джерела даних. Дуже традиційною формою перетворення надвисокої роздільної здатності є панорамування, коли панхроматичні дані використовуються для підвищення роздільної здатності даних MSI або HSI. Різні дослідження показують, що методи глибокого навчання перевершують звичайні підходи pan-sharpening шляхом автоматичного вилучення ознак, які вказують на кореляції між двома типами даних [153,154,155,156].



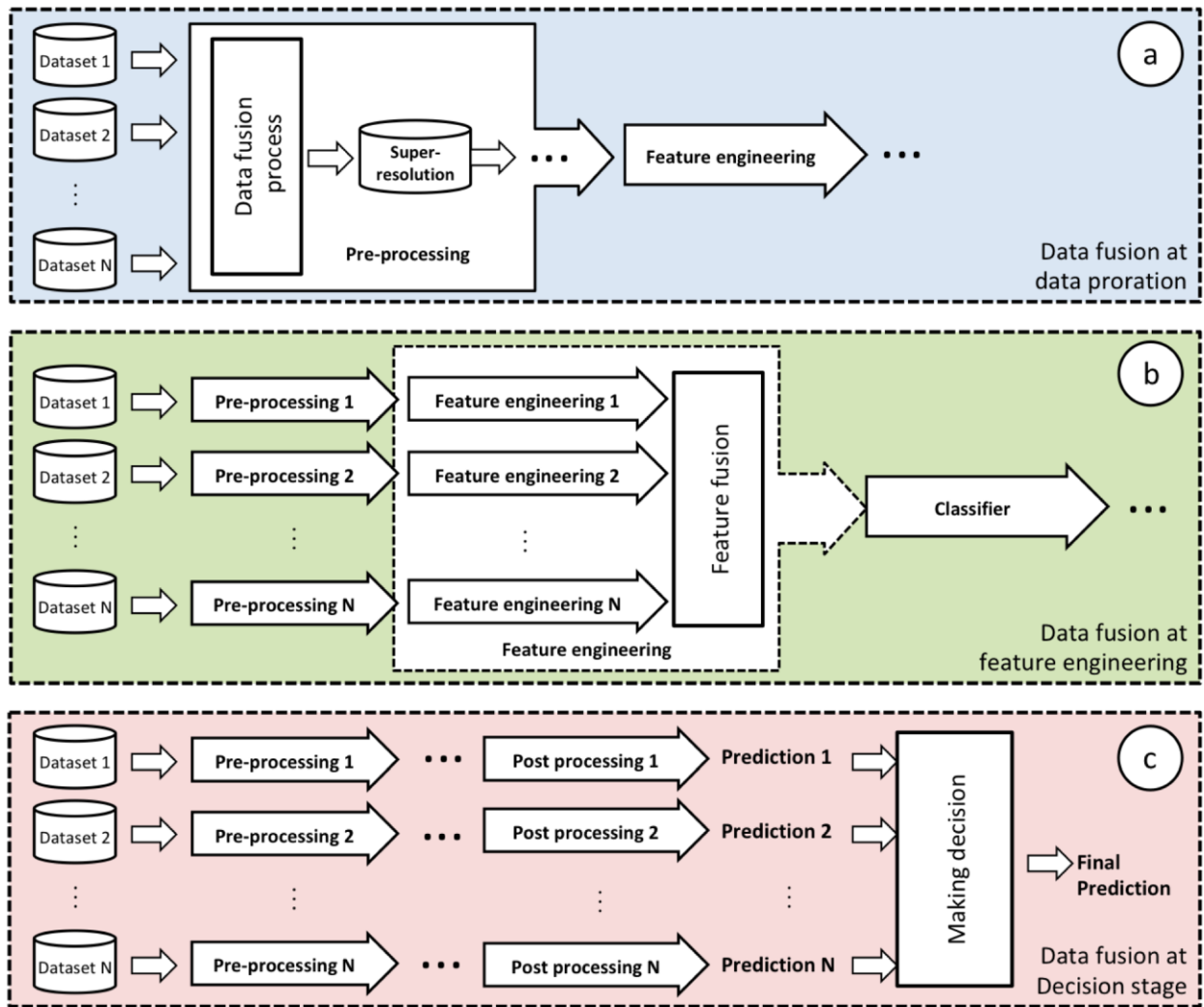


Рисунок 14 – Загальна схема мультимодального об’єднання даних на трьох основних етапах конвеєра машинного навчання: (а) Об’єднання даних на етапі підготовки даних (раннє об’єднання). (б) Об’єднання даних на етапі розробки функцій (об’єднання функцій). (с) Об’єднання даних на самому кінцевому рівні прийняття рішень (пізнє об’єднання).

Генерація надвисокої роздільної здатності за допомогою глибокого навчання досягається за допомогою моделі, в якій дві версії зображення (висока роздільна здатність як ціль і низька роздільна здатність як вхідна інформація) використовуються, щоб навчитися реконструювати зображення з вищою роздільною здатністю із зображення низької роздільної здатності [157,158,159]. Ці типи моделей також стають популярними для збільшення роздільної здатності даних дистанційного зондування [160,161]. Окрім просторової роздільної

здатності, автори [162] застосовують ту саму ідею, використовуючи 3D-CNN на HSI, щоб також забезпечити вищу спектральну якість. Після запуску Sentinel 1 і Sentinel 2 виникло багато питань щодо поєднання SAR і MSI для підвищення роздільної здатності даних з точки зору заповнення прогалів, спричинених атмосферними умовами.

Наприклад, у похмурий день оптичні датчики не можуть вловити поверхню землі. Щоб підійти до цієї проблеми, автори [163] пропонують методологію, засновану на глибокому навчанні, використовуючи часові ряди Sentinel 1 і Sentinel 2 для оцінки часових рядів високої роздільної здатності NDVI для моніторингу сільськогосподарських змін. Інший підхід, спрямований на внутрішню мультимодальність даних MSI, описаний у [164], де запропонована модель забезпечує суперроздільну здатність спектральних смуг нижчої роздільної здатності даних Sentinel-2, використовуючи його спектральні смуги вищої роздільної здатності.

Об'єднання даних на етапі розробки функцій (також називається об'єднанням ознак/представлень) є дуже поширеним і ефективним способом використання мультимодальних даних. Як також зображено на рисунку 14b, замість генерації нової версії вхідних даних, джерела даних зі сцени обробляються паралельно для вилучення ознак. Потім витягнуті характеристики кожного конвеєра збираються разом і передаються в класифікатор. Глибоке навчання стало проривом у цьому процесі. Використовуючи паралельні згорткові потоки, References [165,166] об'єднують дані LiDAR і MSI/HSI на етапі розробки ознак для класифікації культур і земельного покриття. Так само вих. [167] об'єднує дані SAR і MSI, щоб отримати більше наземних деталей для класифікації. Інша цікава робота належить до [168], де автори використовують OSM (Open Street Map) карти для семантичного маркування зображень Earth Observation. Панхроматичні дані та MSI також зазвичай об'єднують у багатьох дослідженнях [169].

Інший етап, на якому може відбуватися об'єднання даних, – це рівень прийняття рішень (також називається пізнім об'єднанням). Як показано на

рисунку 14с, паралельні потоки, що ведуть до прогнозів, розглядаються для кожного джерела даних, і остаточне рішення приймається на основі всіх прогнозів потоків. Як правило, об'єднання даних на рівні прийняття рішень виділяється, коли типи та формати вхідних даних не корельовані автоматично. Дійсно, коли вхідні дані неоднорідні, мультимодальні та багатоджерельні одночасно, важко витягти кореляційну інформацію на ранніх етапах. Через велику кількість нейронів в архітектурах із об'єднанням даних на етапі прийняття рішення час, необхідний для навчання, а отже, і для тестування моделі, значно вищий, ніж в інших архітектурах об'єднання даних. Таким чином, у випадку кореляційних типів даних, як обговорювалося в [167], злиття даних на рівні прийняття рішень не є найкращим підходом. Це також зазначено в роботі в [170], яка порівнює результати пізнього термоядерного синтезу з результатами раннього термоядерного синтезу за повітряними мультиспектральними даними та даними LiDAR і оцінює їх з таким самим висновком. З іншого боку, ситуація з більш неоднорідними типами вхідних даних була досліджена [171]: вони використовують мультимодальні (MSI та SAR), багатоджерельні (аерофотознімки, Sentinel 2 та Sentinel 1) та багаточасові (для Sentinel 1 і 2) типи даних для швидкого вилучення затоплених будівель.

Іноді злиття даних також спрямоване на часову роздільну здатність. Робота в [172], опублікована в 2018 році, є оглядом сучасного стану просторово-часових мультимодальних досліджень злиття даних. У нашій оглядовій роботі не було повідомлено про жодне дослідження, яке вирішує цю проблему за допомогою методів глибокого навчання. Однак автори передбачають потенційні відкриття завдяки глибокому навчанню в цій галузі.

### 1.4.6. Попередня і постобробка

Метою попередньої обробки є в основному покращення необроблених вхідних даних для аналізу. На цьому етапі конвеєра машинного навчання можна використовувати багато методологій, а також техніки глибокого навчання, щоб створити покращений набір даних із необроблених даних. Як обговорювалося раніше, об'єднання даних також може відбуватися на етапі попередньої обробки для створення даних із високою роздільною здатністю. Крім того, коли існують попередньо навчені моделі на різних наборах даних, навчання передачі також може розглядатися на етапі попередньої обробки даних. Крім того, автори [173] використовують перехідне навчання для подолання проблеми шумів із нещодавно запущеними китайськими супутниковими гіперспектральними зображеннями. Основні завдання попередньої обробки, поставлені моделями в літературі, зосереджені на видаленні шумів, виявленні хмари та оцінці роздільної здатності. Окрім оцінки роздільної здатності даних, глибоке навчання також було успішним у видаленні шуму HSI [174,175] і виявленні хмар [176,177] для даних MSI і HSI.

Постобробка — це не обов'язковий етап, який використовується для точного налаштування результату класифікатора, зазвичай із застосуванням методів обробки зображень. Базуючись на попередніх знаннях про очікуваний результат або про потенційні помилки класифікатора та шум, постобробка застосовує набір коригувань до результату для покращення продуктивності моделі. У контексті дистанційного зондування стадія попередньої обробки дуже корисна для векторизації або створення шейп-файлів штучних об'єктів на Землі (наприклад, будівель) [170], для яких відомі морфологічні характеристики очікуваного результату. Умовні випадкові поля (CRF) є основною технікою, яка використовується для цієї мети [178], і вона успішно практикується багатьма дослідженнями спільно з моделями глибокого навчання, націленими на семантично сегментовані карти [75,179,180].

## 1.5. Висновки до розділу 1

В даний час більшість спроб застосувати методи глибокого навчання на даних дистанційного зондування пропонується фахівцями, які не займаються машинним навчанням. У цьому огляді було розглянуто критичні проблеми у використанні таких методів і підкреслено необхідність глибшого розуміння машинного навчання як складної проблеми. Зосереджуючись на класифікації землекористування та ґрунтового покриву мультиспектральних і гіперспектральних зображень, було надано огляд сучасного стану, об'єднавши широкий спектр різних підходів, про які йдеться в літературі, у загальну структуру машинного навчання, яка охоплює різні аспекти всієї проблеми. Було зазначено, як методи глибокого навчання використовувалися на різних етапах інфраструктури для вирішення різних завдань і проблем, виділяючись серед інших підходів.

Існує зростаючий інтерес до використання методів глибокого навчання для широкого спектру застосувань дистанційного зондування, що спонукає галузі інвестувати в цю сферу. Відповідно, очікується швидкий розвиток наземних знань і збільшення кількості відкритих можливостей. Переглядаючи найсучасніші технології, здавалося, що є багатообіцяючі сфери, у яких впровадження глибокого навчання може мати високий потенціал:

Для більшості комерційно життєздатних застосувань просторова роздільна здатність зображень дистанційного зондування повинна бути вищою, ніж може забезпечити будь-який супутник. Тому аерофотознімки дистанційного зондування є більш популярними завдяки вищій просторовій роздільній здатності. Проте обмежене охоплення та низька тимчасова роздільна здатність таких аерофотознімків пов'язані з певними труднощами для багатьох програм, які також залишають простір для використання супутникових зображень. Таким чином, компроміс між тимчасовим і просторовим дозволом закладає основу для подальшого обговорення цього питання.

Дефіцит реальної інформації все ще є проблемою. Точний анотований

набір даних може відкрити нові можливості для дослідників. Більшість доступних рішень страждають від браку фінансування та труднощів з оцінкою їх точності. Дійсно, використання IoT і відкритої наукової структури, яка підтримує інтеграцію громадянської науки, гейміфікації, стимулів і конкурсів, ще належить дослідити.

Незважаючи на постійне збільшення кількості постачальників геопросторових даних, протягом багатьох років не було стандартизованого способу оприлюднення та отримання даних. Зазвичай обробка та аналіз даних виконуються на локальних машинах, на локально реплікованому екземплярі даних. Зі швидким зростанням об'єму даних і обмеженням пам'яті покладатися на звичайні інфраструктури видається неможливим і ефективним. Нещодавно постачальники даних представили хмарну платформу для прямого доступу та аналізу даних, яка пропонує можливість інтеграції даних з різних джерел у найближчому майбутньому. Звичайно, узгодження з досягненнями в інфраструктурі відкриває нові можливості для дослідження.

Нещодавня ідея бортової обробки даних може створити нові виклики: як оголосили NASA та ESA, майбутні супутники планують нести більш потужні процесори, які можуть обробляти дані перед передачею їх на Землю. Однак масштаби потужності та управління енергією є важливою проблемою для процесів на борту. Таким чином, зменшення складності моделей є важливим питанням, яке слід розглянути для майбутніх робіт. Нещодавне дослідження [181], яке пропонує алгоритм налаштування Firefly Harmony Search (FHS) для моделі Deep Belief Network, також доводить, що спрощення моделей також може підвищити точність класифікацій.

Нарешті, глибоке навчання має величезну здатність діяти як незамінний інструмент для вирішення деяких із найсерйозніших і нагальних екологічних проблем нашого часу. Існує відчуття терміновості спрямування дослідницької діяльності на вирішення таких питань, і існують значні потенційні можливості для подальшого розвитку в цій галузі. Крім того, ефективне використання глибокого навчання вимагає нових прикладів і рішучих зусиль для вирішення

технічних проблем, пов'язаних із даними дистанційного зондування та проблемою машин з обмеженими ресурсами. Керування пам'яттю, підготовка даних і завантаження даних є одними з цих технічних проблем, які вимагають подальших зусиль у застосуваннях глибокого навчання. Крім того, дослідженням глибокого навчання в галузі дистанційного зондування не вистачає встановленої структури, яка могла б класифікувати та оптимально групувати моделі. У майбутніх дослідженнях може бути розглянута необхідність створення такої структури, яка може створити основу для раціональної та належної оцінки ефективності моделей.

## РОЗДІЛ 2. НЕЙРОННІ МЕРЕЖІ ДЛЯ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ ТА ПІДГОТОВКА ДАНИХ

У цьому розділі буде розглянуто конволюційні нейронні мережі, які використовуються для сегментації зображень.

### 2.1 Місце згорткових мереж у сегментації зображень

Згорткові мережі — це тип нейронних мереж, які особливо добре підходять для завдань обробки зображень, таких як класифікація зображень і виявлення об'єктів. Їх також можна використовувати для сегментації зображення, яка передбачає поділ зображення на різні сегменти або області, кожна з яких відповідає різному об'єкту чи фону.

Згорткові мережі досягають цього за допомогою серії згорткових шарів, які застосовують набір фільтрів, які можна вивчати, до вхідного зображення. Кожен фільтр невеликий і охоплює лише локальну область зображення, але, застосовуючи багато різних фільтрів у різних положеннях і масштабах, мережа може навчитися виявляти широкий спектр шаблонів на зображенні.

У сегментації зображення ці згорткові фільтри використовуються для ідентифікації різних об'єктів або регіонів на зображенні, а результатом мережі є карта сегментації, яка вказує межі кожного сегмента. Ця карта зазвичай створюється шляхом застосування останнього шару до результату згорткових шарів, який призначає кожному пікселю зображення одному з попередньо визначених сегментів.

Загалом, згорткові мережі є потужним інструментом для сегментації зображень, оскільки вони здатні виділяти інформативні ознаки безпосередньо з вхідних даних і можуть обробляти складні зображення реального світу з високою точністю.



## 2.2 Типи згорткових нейронних мереж

Існує кілька різних типів згорткових мереж, які можна використовувати для сегментації зображення, включаючи повністю згорточні мережі (FCN), мережі кодувальника-декодера та U-Net.

Повністю згорточні мережі (FCN) — це тип згорткових мереж, які замінюють повністю зв'язані рівні стандартної згорткової мережі згортковими шарами, що дозволяє мережі приймати вхідні дані будь-якого розміру та виробляти вихідні дані того самого розміру. Це робить FCN добре придатними для завдань сегментації зображення, оскільки вони можуть обробляти все вхідне зображення та створювати карту сегментації для всього зображення.

Мережі кодера-декодера є іншим типом згорткової мережі, яка часто використовується для сегментації зображення. Вони складаються з двох частин: кодера, який є стандартною згортковою мережею, яка зменшує дискретизацію вхідного зображення для створення компактного представлення, і декодера, який підвищує дискретизацію компактного представлення для створення остаточної карти сегментації. Це дозволяє мережі вивчати компактне представлення вхідного зображення, яке підходить для сегментації.

U-Net — це особливий тип мережі кодера-декодера, розроблений спеціально для завдань сегментації зображення. Він використовує «U-подібну» архітектуру з рядом шарів згортки та об'єднання в частині мережі кодера та серією транспонованих шарів згортки та рівня дискретизації в частині мережі декодера. Це дозволяє U-Net вивчати ієрархічне представлення вхідного зображення, яке добре підходить для сегментації.

## 2.2.1 Повністю згорткові нейронні мережі

Повністю згорткові мережі (FCN) є типом згорткових мереж, які особливо добре підходять для завдань сегментації зображень. Їх називають «повністю згортковими», тому що вони повністю складаються з згорткових шарів без повноцінних зв'язаних шарів.

Основна перевага FCN полягає в тому, що вони можуть приймати вхідні зображення будь-якого розміру та створювати вихідні карти сегментації того самого розміру. Це пояснюється тим, що згорткові шари в FCN можуть працювати з вхідними зображеннями будь-якого розміру, оскільки вони охоплюють лише локальні області вхідних даних і не потребують обробки всього зображення одночасно. Навпаки, стандартні згорткові мережі включають повністю пов'язані рівні, які вимагають вхідних даних фіксованого розміру та створюють вихідні дані фіксованого розміру.

Щоб використовувати FCN для сегментації зображення, мережа зазвичай має частину кодера, яка зменшує дискретизацію вхідного зображення за допомогою серії згорткових шарів і шарів об'єднання, і частину декодера, яка підвищує дискретизацію представлення зі зниженою дискретизацією, використовуючи серію транспонованих згорток і підвищення дискретизації. шарів. Останнім шаром мережі є згортковий шар, який створює карту сегментації для вхідного зображення.

Загалом FCN є потужним інструментом для сегментації зображення, оскільки вони здатні обробити все вхідне зображення та створити карту сегментації високої роздільної здатності. Їх також відносно просто навчити, і їх можна навчити від кінця до кінця з необробленого вхідного зображення.

### 2.2.2 Мережі кодера - декодера

Основна перевага мережевого кодера-декодера для сегментації зображення виникає в тому, що він може вивчити компактне ієрархічне представлення вхідного зображення, яке добре підходить для сегментації. Частина мережі, яка кодує, вчиться витягувати з вхідного зображення характеристики високого рівня, такі як форми та межі об'єктів, а частина мережі, яка використовує ці функції, остаточно відновила карту сегментації.

Щоб навчити мережу кодера-декодера для сегментації зображення, мережу зазвичай навчають наскрізним способом, використовуючи навчальний набір зображень і відповідні базові карти сегментації правдивості. Зараз мережа може отримати нове вхідне зображення та створити для нього сегментацію карти.

Загалом, мережевий кодер-декодер є корисним інструментом для сегментації зображення, оскільки вони можуть навчитися компактному представленню вхідного зображення, яке добре підходить для завдання. Їх також просто навчити, і їх можна навчити від кінця до кінця з необробленого вхідного зображення.

### 2.3 Підготовка даних

Попередня обробка зображень є важливим етапом аналізу супутникових даних, і для цієї мети можна використовувати багато різних методів. Нижче розглянуто деякі поширені методи попередньої обробки зображень для супутникових даних:

Реєстрація зображень: включає в себе вирівнювання зображень від різних датчиків або різного часу до загальної еталонної системи координат, щоб їх

можна було комбінувати або порівнювати. Реєстрацію зображень можна виконати за допомогою різноманітних методів, наприклад реєстрації на основі ознак, яка відповідає характерним ознакам зображень, або реєстрації на основі інтенсивності, яка вирівнює шаблони інтенсивності на зображеннях.

Мозаїка зображень: це передбачає об'єднання кількох зображень однієї області для створення більшого та детальнішого зображення. Мозаїку зображень можна виконати за допомогою різноманітних методів, наприклад розтушовування, яке плавно поєднує зображення, або панорамування, яке поєднує високу просторову роздільну здатність одного зображення з високою спектральною роздільною здатністю іншого зображення.

Виправлення зображень: це передбачає виправлення спотворень зображень, спричинених положенням супутника та орієнтацією, а також кривизною поверхні Землі. Виправлення зображення можна виконати за допомогою різноманітних методів, таких як поліноміальне або сплайнове викривлення, які використовують математичні функції для відображення спотворених пікселів у їхніх неспотворених положеннях, або геометричне виправлення, яке використовує контрольні точки землі та модель рельєфу для виправлення викривлення.

Покращення зображення: це передбачає застосування алгоритмів для покращення візуальної якості зображень, наприклад підвищення контрастності чи різкості або видалення шуму чи туману. Поліпшення зображення можна виконати за допомогою різноманітних методів, таких як вирівнювання гістограми, яке розширює діапазон інтенсивності зображення для покращення контрастності, або просторові фільтри, які згладжують або роблять зображення різкішим, щоб усунути шум або покращити функції.

Сегментація зображення: передбачає поділ зображення на різні області або

сегменти, кожен з яких відповідає іншому об'єкту або фону. Сегментацію зображення можна виконати за допомогою різноманітних методів, таких як кластеризація, яка групує пікселі разом на основі їх подібності, або виявлення країв, що визначає межі між різними об'єктами на зображенні.

Загалом існує багато різних методів, які можна використовувати для попередньої обробки зображень супутникових даних, і відповідний метод залежатиме від конкретних цілей і характеристик даних.

## **2.4 Висновки до розділу 2**

У цьому розділі було розглянуто конволюційні нейронні мережі, які використовуються для сегментації зображень. Перш за все, було описано місце згорткових мереж у сегментації зображень, де було визначено основні поняття та задачі, які вони розв'язують. Розглянуто типи згорткових нейронних мереж і детально описані основні переваги кожного з них та задачі, для яких вони можуть бути використані. І врешті решт, розглянуто підходи до підготовки даних з вказанням різних методів передобробок і спектр задач, в яких кожен з них використовується.

## **РОЗДІЛ 3. РЕАЛІЗАЦІЯ НЕЙРОМЕРЕЖЕВОЇ МОДЕЛІ РОЗПІЗНАВАННЯ ВИРВ ВІД БОМБАРДУВАННЯ ЗА СУПУТНИКОВИМИ ДАНИМИ**

### **3.1 Попередня обробка даних**

Попередня обробка зображень є важливим кроком у підготовці даних для сегментації зображень, і QGIS надає ряд інструментів, які можна використовувати для цієї мети.

Деякі поширені етапи попередньої обробки для сегментації зображень у QGIS включають:

Реєстрація зображень: це передбачає вирівнювання зображень від різних датчиків або різного часу до загальної еталонної системи координат, щоб їх можна було комбінувати або порівнювати. У QGIS було використано алгоритм «Деформація» у меню «Растр», щоб зареєструвати зображення в загальній системі координат.

Мозаїка зображень: це передбачає об'єднання кількох зображень однієї області для створення більшого та детальнішого зображення. У QGIS було використано алгоритм Merge у меню Raster, щоб створити мозаїку з кількох зображень.

Виправлення зображень: це передбачає виправлення спотворень зображень, спричинених положенням супутника та орієнтацією, а також кривизною поверхні Землі. У QGIS було використано алгоритм Warp (Reproject) у меню Raster для виправлення зображень за допомогою контрольних точок землі та моделі висоти.

Покращення зображення: це передбачає застосування алгоритмів для

покращення візуальної якості зображень, наприклад підвищення контрастності чи різкості або видалення шуму чи туману. У QGIS ви можете використовувати Raster Calculator у меню Raster, щоб застосувати різноманітні математичні операції до даних зображення, або ви можете скористатися алгоритмом Contrast Enhancement у меню Raster, щоб покращити контрастність і видимість зображення.

Загалом ці етапи попередньої обробки допомогли підвищити точність і ефективність алгоритмів сегментації зображень у застосуванні до супутникових даних у QGIS. Вони також можуть допомогти зробити результати сегментації зображення більш зрозумілими та корисними для подальших завдань.

На рисунку нижче наведено приклад оригінальної маски зображення.

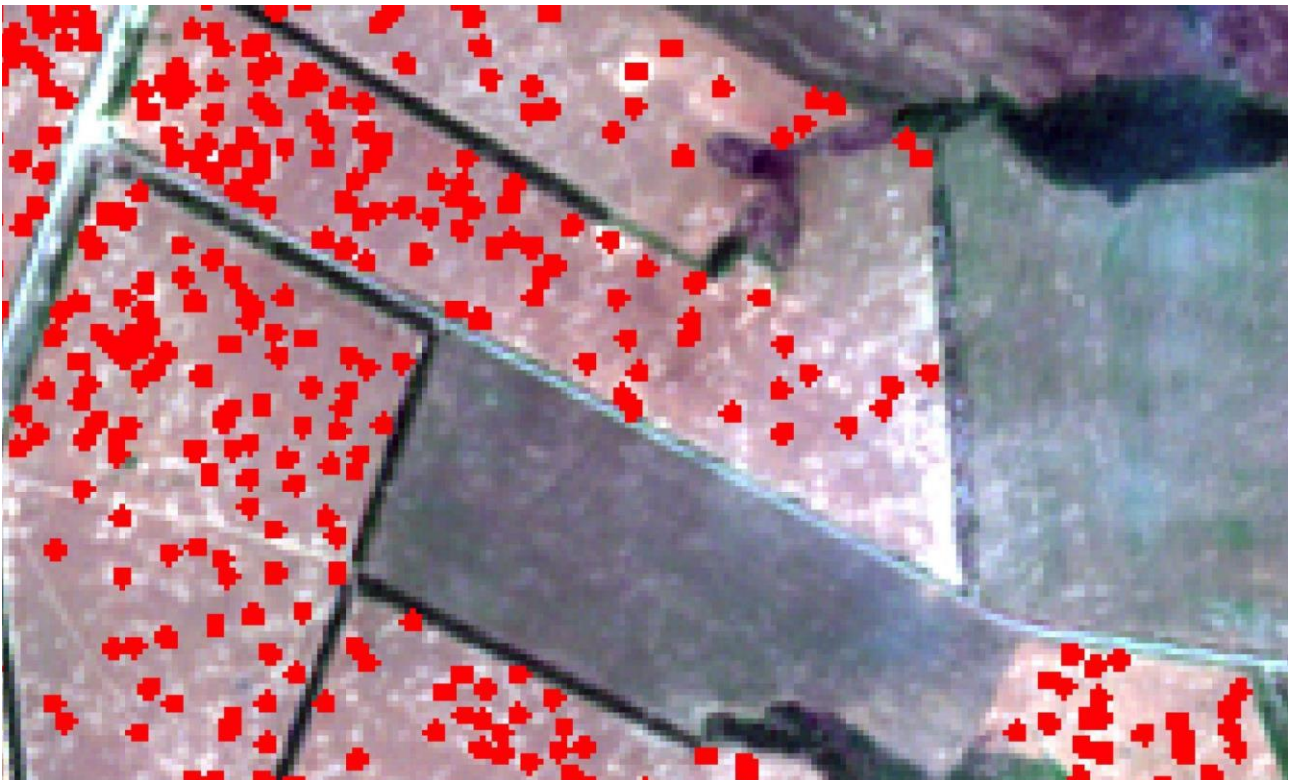


Рисунок 15 – Приклад оригінальної маски.

### 3.2 Підготовка датасету

Під час підготовки набору даних для завдання сегментації зображення було виконано п'ять кроків. Ці кроки допомогли підвищити точність і ефективність алгоритму сегментації зображення, а також зробили результати більш зручними для інтерпретації та корисними для подальших завдань.

1. Перший етап передбачає вирівнювання зображень від різних датчиків або різного часу до загальної еталонної системи координат, щоб їх можна було комбінувати або порівнювати. Це важливо, оскільки дозволяє алгоритму точно ідентифікувати об'єкти та межі на зображеннях, навіть якщо вони трохи зміщені або повернуті.

2. Другий етап полягає у об'єднанні кількох зображень однієї області для створення більшого та детальнішого зображення. Це потрібно, оскільки дозволяє алгоритму бачити більше деталей на зображеннях, що може покращити його здатність ідентифікувати та сегментувати об'єкти.

3. Також критично важливо виправити всі спотворення зображень, спричинених положенням супутника та орієнтацією, а також кривизною поверхні Землі. Це важливо, оскільки дозволяє алгоритму точно вимірювати розмір і форму об'єктів на зображеннях, навіть якщо вони спотворені перспективою супутника.

4. Четвертим етапом необхідно застосувати алгоритми для покращення візуальної якості зображень, наприклад підвищення контрастності чи різкості або видалення шуму чи туману. Цей крок необхідний, оскільки він може полегшити обробку зображень алгоритмом і може покращити його здатність ідентифікувати та сегментувати об'єкти.

5. Заключний етап включає ручне маркування зображень, щоб вказати межі різних об'єктів або регіонів на зображеннях. Це дозволяє надати правдиві



дані, які алгоритм може використовувати, щоб навчитися точно сегментувати зображення.

Загалом, підготовка набору даних для завдання сегментації зображення була важливим кроком у забезпеченні успішної роботи алгоритму та допомогло підвищити його точність і ефективність.

В роботі використано супутникові знімки Sentinel-2. Знімки з супутника було нарізано на зображення у кількості 1355 од.

На рисунках нижче зображено приклад оригінального зображення, яке нейронна мережа отримує на вхід а також сегментоване зображення оригінальної маски, яка містить інформацію по кожному пікселю: чи піксель є фоном або ж об'єктом (бомбардуванням).

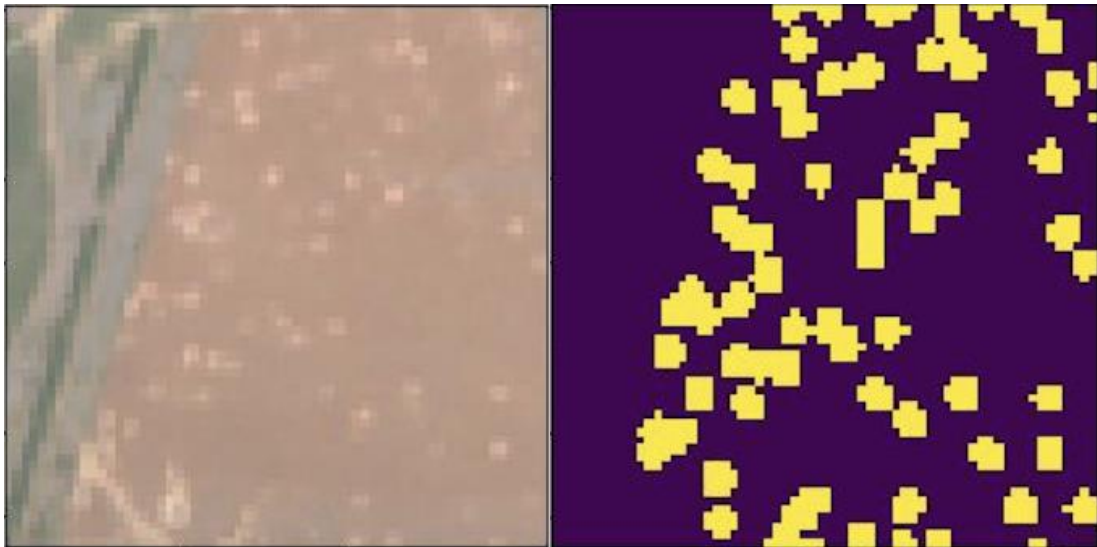


Рисунок 16 – Оригінальне зображення (зліва) та оригінальна маска (праворуч).

### 3.3 Функції витрат

У задачах сегментації зображень функція витрат є показником того, наскільки добре алгоритм виконує сегментацію зображень. Він використовується для оцінки передбачень алгоритму та керування процесом оптимізації, щоб алгоритм міг покращувати свою продуктивність з часом.

У роботі було проведено експеримент з кожною з нижче вказаних функцій витрат з метою ефективно навчатися та отримувати точні та корисні результати.

Деякі поширені функції втрати для сегментації зображення включають:

Dice Loss: це часто використовувана функція витрат для задач сегментації бінарних і багатокласових зображень. Вона вимірює перекриття між прогнозованою картою сегментації та наземною картою сегментації правдивості, обчислюючи відношення кількості пікселів, які правильно класифіковані на обох картах, до загальної кількості пікселів на обох картах.

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

IoU функція витрат: це ще одна часто використовувана функція втрати для завдань сегментації зображення. Вона вимірює перекриття між прогнозованою картою сегментації та базовою картою сегментації правдивості, обчислюючи відношення кількості пікселів, які правильно класифіковані на обох картах, до загальної кількості пікселів, які класифікуються на кожній карті.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Focal Loss: це варіант втрати перехресної ентропії, який часто використовується для завдань сегментації зображення з незбалансованими класами, де одні класи зустрічаються набагато частіше, ніж інші. Вона призначає вищі ваги пікселям, які належать до рідкісних класів, щоб алгоритм міг зосередитись на навчанні точно класифікувати ці пікселі.

Формально фокусні втрати додають фактор  $(1 - p_t)^\gamma$  до стандартного критерію перехресної ентропії. Налаштування  $\gamma < 0$  зменшує відносні втрати для добре класифікованих прикладів  $p_t > 0.5$ , приділяючи більше уваги складним, неправильно класифікованим прикладам. Тут є настроюваний параметр фокусування  $\gamma \geq 0$ .

$$FL(p_t) = -(1 - p_t)^\gamma \log(p_t)$$

Загалом відповідна функція втрат для завдання сегментації зображення залежатиме від конкретних цілей і характеристик завдання, а також від можливостей алгоритму. Вибір відповідної функції втрат може допомогти алгоритму ефективно навчатися та отримувати точні та корисні результати.

Нижче представлено графік функції витрат, на якому видно, що з ростом кількості епох покращились результати.

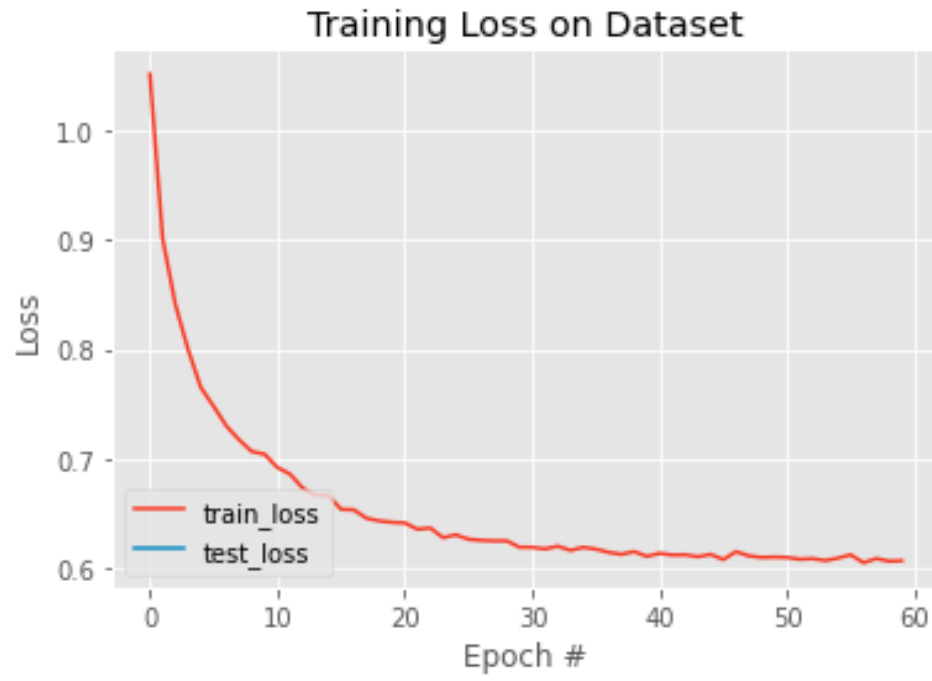


Рисунок 17 – Графік зміни функції втрат на тренувальній вибірці.

### 3.4 Архітектура моделі

Загалом наша модель U-Net (Рисунок 18) складатиметься з класу Encoder і класу Decoder. Кодер поступово зменшуватиме просторовий вимір для стиснення інформації. Крім того, це збільшить кількість каналів, тобто кількість карт функцій на кожному етапі, що дозволить нашій моделі захоплювати різні деталі чи особливості нашого зображення. З іншого боку, декодер візьме остаточне представлення кодера та поступово збільшуватиме просторовий розмір і зменшуватиме кількість каналів, щоб остаточно вивести маску сегментації того самого просторового розміру, що й вхідне зображення.

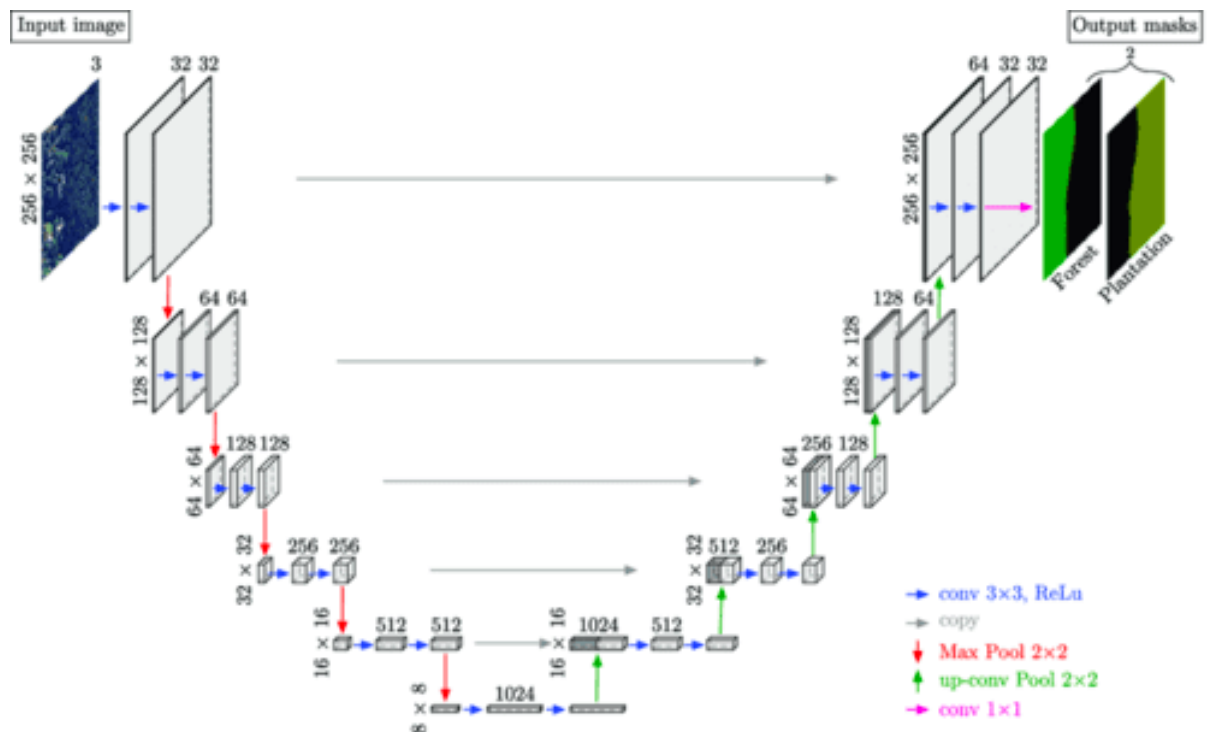


Рисунок 18 – Архітектура моделі сегментації зображень U-Net.

Далі ми визначаємо модуль Block як будівельну одиницю нашої архітектури кодера та декодера. Варто зазначити, що всі моделі або підчастини моделі, які ми визначаємо, повинні успадковувати клас модуля PyTorch, який є батьківським класом у PyTorch для всіх модулів нейронної мережі.

Тепер, коли визначено підмодулі, які складають нашу модель U-Net, все готово для побудови класу моделі U-Net.

Модель U-Net приймає такі параметри як вхідні дані:

*encChannels*: кортеж визначає поступове збільшення розмірності каналу, коли наш вхідний сигнал проходить через кодувальник. Ми починаємо з 3 каналів (тобто RGB), а потім подвоюємо кількість каналів.

*decChannels*: кортеж визначає поступове зменшення розмірності каналу, коли наш вхідний сигнал проходить через декодер. Ми зменшуємо канали в 2 рази на кожному кроці.

*nbClasses*: це визначає кількість класів сегментації, за якими ми маємо класифікувати кожен піксель. Зазвичай це відповідає кількості каналів у нашій вихідній карті сегментації, де ми маємо один канал для кожного класу.

Оскільки ми працюємо з двома класами (тобто класифікація на два класи), ми зберігаємо один канал і використовуємо порогове значення для класифікації.

*retainDim*: вказує, чи хочемо ми зберегти вихідний вихідний розмір.

*outSize*: визначає просторові розміри вихідної карти сегментації. Ми встановлюємо для нього той самий розмір, що й наше вхідне зображення.

### 3.5 Огляд результатів

Перейдемо до результатів. Нижче проілюстровано результат роботи нейронної мережі для розпізнавання вирв від бомбардувань для двох проведених експериментів.

Для обох експериментів використовувалися дані отримані супутником Sentinel-2 з передмістя Ірпеня станом на 31 березня 2022 року.

#### 3.5.1 Експеримент 1

В першому експерименті було використано конволюційну нейронну мережу U-Net. Архітектура нейронної мережі складається з трьох блоків.

В результаті проходження 100 епох отримано такі результати:

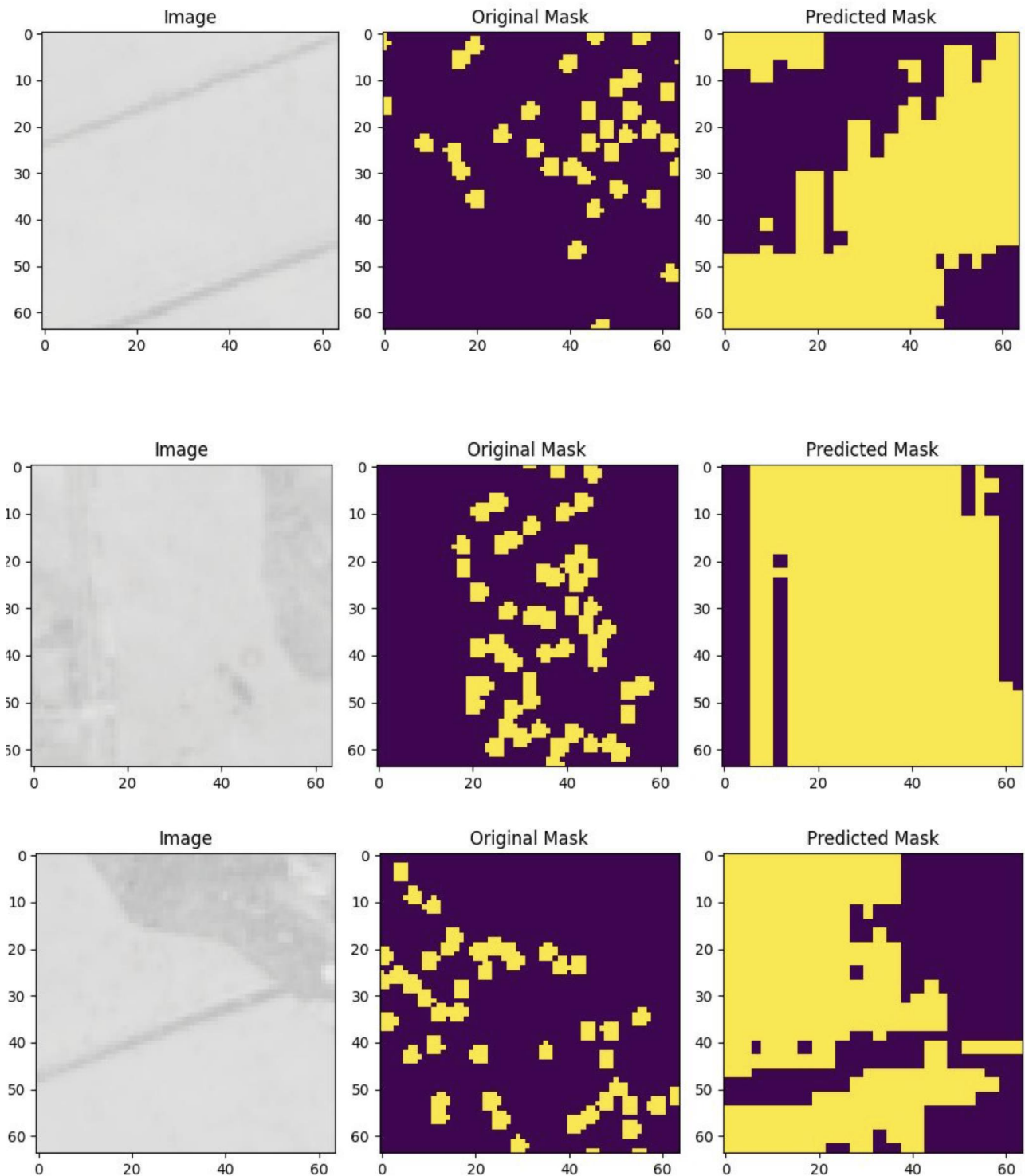


Рисунок 19 – Оригінальне зображення, оригінальна маска, прогнозована маска в результаті тестування трьохшарової нейронної мережі U-Net.

З наведених вище зображень видно, що нейронна мережа досить невдало

визначає вирви від бомбардувань. Схоже, що крок ініціалізації та вибрана архітектура моделі потребує вдосконалення.



### 3.5.2 Експеримент 2

В другому експерименті було взято за основу U-Net, який складається з п'яти блоків. Для ініціалізації ваг використано imagenet та resnet-34.

В результаті проходження 100 епох було отримано відповідні результати:

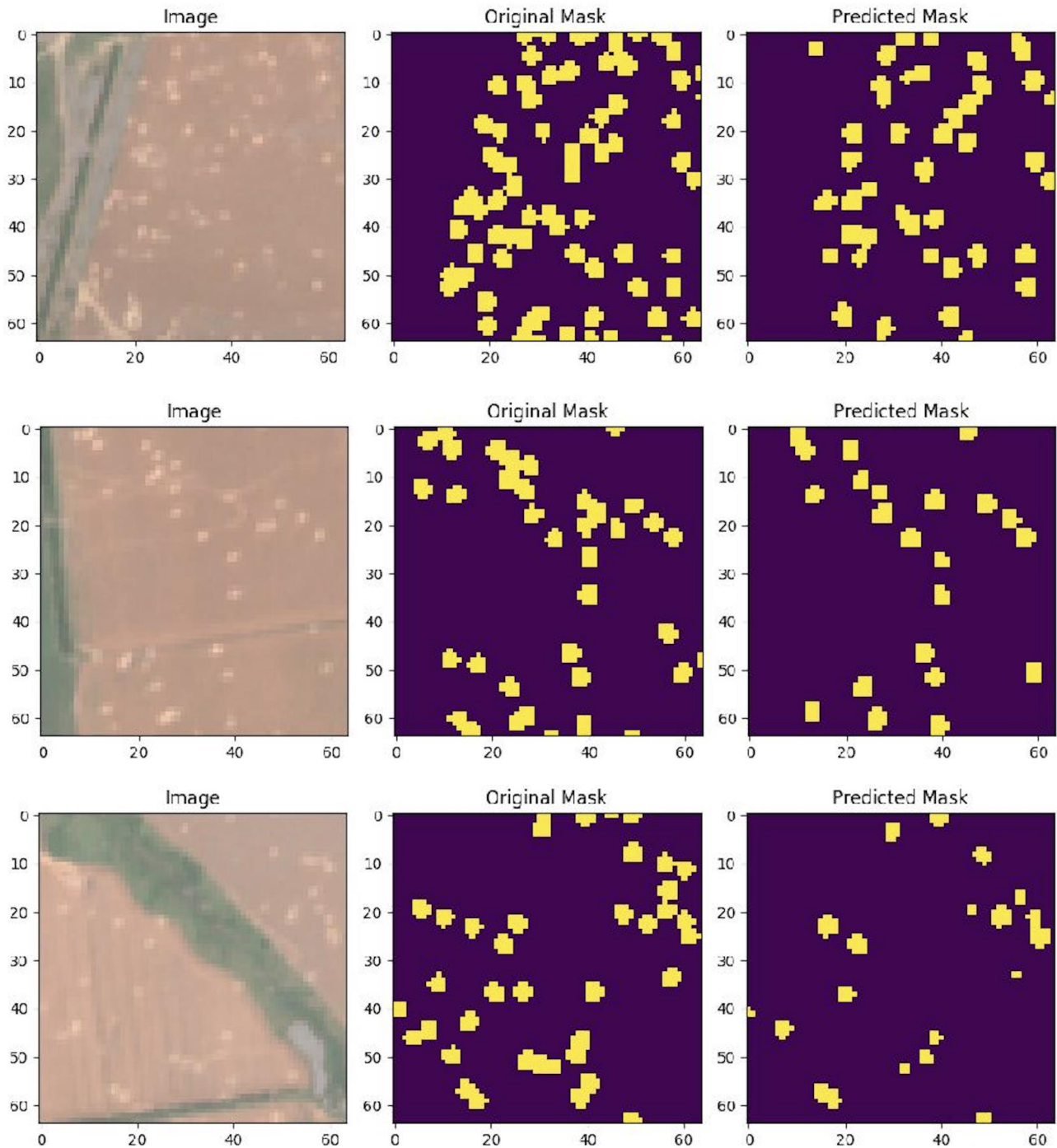


Рисунок 20 – Оригінальне зображення, оригінальна маска, прогнозована маска в результаті тестування п'ятишарової нейронної мережі U-Net.

### 3.5.3 Оцінка точності запропонованої моделі

З наведених вище зображень видно, що нейронна мережа досить вдало визначає вирви від бомбардувань. Для того, щоб надати оцінку точності отриманих результатів було використано функцію втрат IoU.

Формули підрахунку точності розпізнавання наведено нижче:

$$Intersection = \sum (inputs * targets)$$

$$Total = \sum (inputs + targets)$$

$$Union = Total - Intersection$$

$$IoU = \frac{Intersection + Smooth}{Union + Smooth}$$

де *inputs* – вхідний тензор, *targets* – прогнозований тензор, *smooth* – константа, яка дорівнює 1.

В результаті підрахунку точності отримано результат близько  $\frac{9}{10}$  вірно розпізнаних вирв від бомбардувань.

Цей результат є досить гарним, враховуючи, що резолюція супутникових знімків Sentinel-2 є порівняно низькою.

### 3.6 Висновки до розділу 3

У цьому розділі була запропонована нейромережева модель розпізнавання вирв від бомбардувань за супутниковими даними. Перш за все, було

продемонстровано, яким чином відбулася попередня обробка даних та був підготовлений датасет. Далі, було описано функції витрат, які були реалізовані у роботі, порівняно їх ключові недоліки та переваги, продемонстровано графік зміни функції витрат на тренувальній вибірці.

Було проведено два експерименти: один з трьохшаровою архітектурою U-Net, інший з п'ятишаровою архітектурою U-Net, в якому для ініціалізації вагів було використано imagenet та resnet-34.

В результаті двох експериментів було виявлено, що в другому випадку нейромережева модель визначає вирви від бомбардувань з точністю 90%.

## ВИСНОВКИ

В результаті роботи було:

1. Розглянуто існуючі підходи до розпізнавання зображень за супутниковими даними. Зокрема було розглянуто критичні проблеми у використанні різних методів побудови карти сегментації і підкреслено необхідність глибшого розуміння машинного навчання як складної проблеми.
2. Визначено вплив глибокого навчання як такий, що має величезну здатність діяти як незамінний інструмент для аналізу супутникових знімків. Існує відчуття терміновості спрямування дослідницької діяльності на вирішення таких питань, і існують значні потенційні можливості для подальшого розвитку в цій галузі.
3. Проаналізовано найсучасніші підходи до використання нейронних мереж для сегментації зображень. Розглянуто типи згорткових нейронних мереж і детально описані основні переваги кожного з них та задачі, для яких вони можуть бути використані.
4. Розглянуто підходи до підготовки даних з вказанням різних методів передобробок і спектр задач, в яких кожен з них використовується.
5. Запропонована модель нейронної мережі для ідентифікації вирв від бомбардувань на основі супутникових даних. Спочатку було показано, як був підготовлений набір даних і як дані були попередньо оброблені. Потім були окреслені функції витрат, які використовувалися в роботі, їхні основні переваги та недоліки, а також був представлений графік, що показує, як змінювалась функція витрат на навчальній вибірці.
6. Проведено два експерименти: перший використовуючи imagenet і resnet-34 для ініціалізації вагових коефіцієнтів у п'ятирівневій архітектурі U-Net, другий з трьохшаровою архітектурою U-Net. Згідно з результатами двох досліджень, було виявлено, що запропонована модель нейронної мережі визначає справжні вирви від бомбардувань з точністю 90% у всіх випадках.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Азарсков, В. Н., Блохин, Л. Н., Житецкий, Л. С., Куссуль, Н. Н. Робастные методы оценивания, идентификации и адаптивного управления. К.: НАУ, 2004, 498 с.
2. Куссуль Н. Н., Шелестов А. Ю. Grid-системы для задач исследования Земли.: Архитектура, модели и технологии. – Наукова думка, 2008, 452 с.
3. Куссуль Н. Н. и др. Регрессионные модели оценки урожайности сельскохозяйственных культур по данным MODIS //Современные проблемы дистанционного зондирования Земли из космоса. – 2012. – Т. 9. – №. 1. – С. 95.
4. Shelestov A. et al. Cloud approach to automated crop classification using Sentinel-1 imagery //IEEE Transactions on Big Data. – 2019. – Т. 6. – №. 3. – С. 572-582.
5. Kolotii, A., Kussul, N., Shelestov, A., Skakun, S., Yailymov, B., Basarab, R., . . . Ostapenko, V. (2015). Comparison of biophysical and satellite predictors for wheat yield forecasting in Ukraine. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, , 40(7W3) 39-44. doi:10.5194/isprsarchives-XL-7-W3-39-2015
6. Kussul, N., Mykola, L., Shelestov, A., & Skakun, S. (2018). Crop inventory at regional scale in Ukraine: Developing in season and end of season crop maps with multi-temporal optical and SAR satellite imagery. European Journal of Remote Sensing, 51(1), 627-636.
7. Kussul, N., Shelestov, A., Lavreniuk, M., Butko, I., & Skakun, S. (2016). Deep learning approach for large scale land cover mapping based on remote sensing data fusion. Paper presented at the International Geoscience and Remote Sensing Symposium (IGARSS), , 2016-November 198-201.
8. ESA. Towards a European AI4EO R&I Agenda. 2018.
9. Newbold, T.; Hudson, L.N.; Hill, S.L.; Contu, S.; Lysenko, I.; Senior, R.A.; Börger, L.; Bennett, D.J.; Choimes, A.; Collen, B.; et al. Global effects of land use on local terrestrial biodiversity. *Nature* **2015**, 520, 45.
10. Vitousek, P.M.; Mooney, H.A.; Lubchenco, J.; Melillo, J.M. Human domination of Earth's ecosystems. *Science* **1997**, 277, 494–499.

11. Feddema, J.J.; Oleson, K.W.; Bonan, G.B.; Mearns, L.O.; Buja, L.E.; Meehl, G.A.; Washington, W.M. The importance of land-cover change in simulating future climates. *Science* **2005**, *310*, 1674–1678.
12. Turner, B.L.; Moss, R.H.; Skole, D. *Relating Land Use and Global Land-Cover Change*; IGBP Report 24, HDP Report 5; IGDP Report No. 24; HDP Report No. 5; International Geosphere-Biosphere Programme: Stockholm, Sweden, 1993.
13. United Nations Office for Disaster Risk Reduction. Sendai framework for disaster risk reduction 2015–2030. Proceedings of the 3rd United Nations World Conference on Disaster Risk Reduction (WCDRR), Sendai, Japan, 14–18 March 2015; pp. 14–18.
14. Zikopoulos, P.; Eaton, C. *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*; McGraw-Hill Osborne Media: New York, NY, USA, 2011.
15. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36.
16. Fisher, P.; Comber, A.J.; Wadsworth, R. Land use and land cover: Contradiction or complement. In *Re-Presenting GIS*; Wiley: New York, NY, USA, 2005; pp. 85–98.
17. Food and Agriculture Organization of the United Nations. 2019.
18. Di Gregorio, A. *Land Cover Classification System: Classification Concepts and User Manual: LCCS*; Food & Agriculture Org.: Rome, Italy, 2005; Volume 2.
19. Isikdogan, F.; Bovik, A.C.; Passalacqua, P. Surface water mapping by deep learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 4909–4918.
20. Rezaee, M.; Mahdianpari, M.; Zhang, Y.; Salehi, B. Deep convolutional neural network for complex wetland classification using optical remote sensing imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 3030–3039.
21. Huang, B.; Zhao, B.; Song, Y. Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote

sensing imagery. *Remote Sens. Environ.* **2018**, *214*, 73–86.

22. Hu, J.; Mou, L.; Schmitt, A.; Zhu, X.X. FusioNet: A two-stream convolutional neural network for urban scene classification using PolSAR and hyperspectral data. In Proceedings of the 2017 Joint Urban Remote Sensing Event (JURSE), Dubai, UAE, 6–8 March 2017; pp. 1–4.

23. Kussul, N.; Lavreniuk, M.; Skakun, S.; Shelestov, A. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 778–782.

24. Awad, M.; Jomaa, I.; Arab, F. Improved capability in stone pine forest mapping and management in Lebanon using hyperspectral CHRIS-Proba data relative to Landsat ETM+. *Photogramm. Eng. Remote Sens.* **2014**, *80*, 725–731.

25. Marschner, F. *Major Land Uses in the United States (Map Scale 1:5,000,000)*; USDA Agricultural Research Service: Washington, DC, USA, 1950; Volume 252.

26. Anderson, J.R. *A Land Use and Land Cover Classification System for Use with Remote Sensor Data*; US Government Printing Office: Washington, DC, USA, 1976; Volume 964.

27. Cowardin, L.M.; Carter, V.; Golet, F.C.; LaRoe, E.T. *Classification of Wetlands and Deepwater Habitats of the United States*; Technical Report; US Department of the Interior, US Fish and Wildlife Service: Washington, DC, USA, 1979.

28. Pohl, C.; Van Genderen, J.L. Review article multisensor image fusion in remote sensing: Concepts, methods and applications. *Int. J. Remote Sens.* **1998**, *19*, 823–854.

29. Congalton, R.G. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sens. Environ.* **1991**, *37*, 35–46.

30. Singh, A. Review article digital change detection techniques using remotely-sensed data. *Int. J. Remote Sens.* **1989**, *10*, 989–1003.

31. Kasischke, E.S.; Melack, J.M.; Dobson, M.C. The use of imaging radars

for ecological applications—A review. *Remote Sens. Environ.* **1997**, *59*, 141–156.

32. Li, S.; Kang, X.; Fang, L.; Hu, J.; Yin, H. Pixel-level image fusion: A survey of the state of the art. *Inf. Fusion* **2017**, *33*, 100–112.

33. Ma, L.; Liu, Y.; Zhang, X.; Ye, Y.; Yin, G.; Johnson, B.A. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *152*, 166–177.

34. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.

35. Paoletti, M.; Haut, J.; Plaza, J.; Plaza, A. Deep learning classifiers for hyperspectral imaging: A review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *158*, 279–317.

36. Li, S.; Song, W.; Fang, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Deep learning for hyperspectral image classification: An overview. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6690–6709.

37. Goetz, A.F. Three decades of hyperspectral remote sensing of the Earth: A personal view. *Remote Sens. Environ.* **2009**, *113*, S5–S16.

38. Ghamisi, P.; Maggiori, E.; Li, S.; Souza, R.; Tarablaka, Y.; Moser, G.; De Giorgi, A.; Fang, L.; Chen, Y.; Chi, M.; et al. New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning. *IEEE Geosci. Remote Sens. Mag.* **2018**, *6*, 10–43.

39. Imani, M.; Ghassemian, H. An overview on spectral and spatial information fusion for hyperspectral image classification: Current trends and challenges. *Inf. Fusion* **2020**, *59*, 59–83.

40. USGS. USGS Earth Explorer. 2019.

41. USGS. USGS Global Visualization Viewer. 2019.

42. NASA. NASA Earth Observation—NEO. 2019.



43. ESA. The Copernicus Open Access Hub. 2019.
44. NASA. NASA Earth Data Search. 2019.
45. NOAA. NOAA Data Access. 2019.
46. NOAA. NOAA Digital Coast. 2019.
47. IPUMS. IPUMS Terra Integrates Population and Environmental Data. 2018.
48. Penatti, O.A.; Nogueira, K.; Dos Santos, J.A. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 44–51.
49. Demir, I.; Koperski, K.; Lindenbaum, D.; Pang, G.; Huang, J.; Basu, S.; Hughes, F.; Tuia, D.; Raska, R. Deep Globe 2018: A challenge to parse the earth through satellite images. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 172–17209.
50. Basu, S.; Ganguly, S.; Mukhopadhyay, S.; DiBiano, R.; Karki, M.; Nemani, R. Deepsat: A learning framework for satellite imagery. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Washington, DC, USA, 3–6 November 2015; p. 37.
51. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
52. GIC. Hyperspectral Remote Sensing Scenes. 2020.
53. Geoscience. *2013 IEEE GRSS Data Fusion Contest*; GRSS: Piscataway, NJ, USA, 2013.
54. Codalab. *DeepGlobe Land Cover Classification Challenge*; DeepGlobe: Salt Lake City, UT, USA, 2018.
55. Basu, S. *SAT-4 and SAT-6 Airborne Datasets*; Louisiana State University:

Baton Rouge, LA, USA, 2015.

56. University of California, Merced. *UC Merced Land Use Dataset*; University of California, Merced: Merced, CA, USA, 2010.

57. Patrero. *Brazilian Coffee Scenes Dataset*; Patrero: San Francisco, CA, USA, 2015.

58. System(EOS). Crop Monitoring. 2020.

59. Awad, M.M.; Alawar, B.; Jbeily, R. A new crop spectral signatures database interactive tool (CSSIT). *Data* **2019**, *4*, 77.

60. Global Forest Watch. Developer Tools. 2020.

61. SERVIR-Mekong. Surface Water Mapping Tool. 2020.

62. Wolpert, D.H. The lack of a priori distinctions between learning algorithms. *Neural Comput.* **1996**, *8*, 1341–1390.

63. Wolpert, D.H.; Macready, W.G. Coevolutionary free lunches. *IEEE Trans. Evol. Comput.* **2005**, *9*, 721–735.

64. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv* **2016**, arXiv:1611.03530.

65. Kawaguchi, K.; Kaelbling, L.P.; Bengio, Y. Generalization in deep learning. *arXiv* **2017**, arXiv:1710.05468.

66. Saxe, A.M.; Bansal, Y.; Dapello, J.; Advani, M.; Kolchinsky, A.; Tracey, B.D.; Cox, D.D. On the information bottleneck theory of deep learning. *J. Stat. Mech. Theory Exp.* **2019**, *2019*, 124020.

67. Dinh, L.; Pascanu, R.; Bengio, S.; Bengio, Y. Sharp minima can generalize for deep nets. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 10–15 July 2017; pp. 1019–1028.

68. Zou, Q.; Ni, L.; Zhang, T.; Wang, Q. Deep learning based feature selection for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2321–2325.

69. Han, W.; Feng, R.; Wang, L.; Cheng, Y. A semi-supervised generative framework with deep learning features for high-resolution remote sensing image

scene classification. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 23–43.

70. IBM. Removing the Hunch in Data Science with AI-Based Automated Feature Engineering. 2017.

71. Zhang, L.; Zhang, L.; Tao, D.; Huang, X. Tensor discriminative locality alignment for hyperspectral image spectral–spatial feature extraction. *IEEE Trans. Geosci. Remote Sens.* **2012**, *51*, 242–256.

72. Hughes, G. On the mean accuracy of statistical pattern recognizers. *IEEE Trans. Inf. Theory* **1968**, *14*, 55–63.

73. Rasti, B.; Hong, D.; Hang, R.; Ghamisi, P.; Kang, X.; Chanussot, J.; Benediktsson, J.A. Feature extraction for hyperspectral imagery: The evolution from shallow to deep. *arXiv* **2020**, arXiv:2003.02822.

74. Yu, S.; Jia, S.; Xu, C. Convolutional neural networks for hyperspectral image classification. *Neurocomputing* **2017**, *219*, 88–98.

75. Sun, W.; Du, Q. Graph-regularized fast and robust principal component analysis for hyperspectral band selection. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 3185–3195.

76. Zabalza, J.; Ren, J.; Ren, J.; Liu, Z.; Marshall, S. Structured covariance principal component analysis for real-time onsite feature extraction and dimensionality reduction in hyperspectral imaging. *Appl. Opt.* **2014**, *53*, 4440–4449. **[PubMed]**

77. Chen, S.; Zhang, D. Semisupervised dimensionality reduction with pairwise constraints for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2010**, *8*, 369–373.

78. Archibald, R.; Fann, G. Feature selection and classification of hyperspectral images with support vector machines. *IEEE Geosci. Remote Sens. Lett.* **2007**, *4*, 674–677.

79. Kuo, B.C.; Ho, H.H.; Li, C.H.; Hung, C.C.; Taur, J.S. A kernel-based feature selection method for SVM with RBF kernel for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *7*, 317–326.

80. Zhong, Z.; Li, J.; Luo, Z.; Chapman, M. Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 847–858.
81. Mou, L.; Ghamisi, P.; Zhu, X.X. Unsupervised spectral–spatial feature learning via deep residual Conv–Deconv network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 391–406.
82. Audebert, N.; Le Saux, B.; Lefèvre, S. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In Proceedings of the Asian Conference on Computer Vision (ACCV), Taipei, Taiwan, 20–24 November 2016; pp. 180–196.
83. Tao, C.; Pan, H.; Li, Y.; Zou, Z. Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2438–2442.
84. Ma, X.; Wang, H.; Geng, J. Spectral–spatial classification of hyperspectral image based on deep auto-encoder. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 4073–4085.
85. Zabalza, J.; Ren, J.; Zheng, J.; Zhao, H.; Qing, C.; Yang, Z.; Du, P.; Marshall, S. Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging. *Neurocomputing* **2016**, *185*, 1–10.
86. Lunga, D.; Prasad, S.; Crawford, M.M.; Ersoy, O. Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning. *IEEE Signal Process. Mag.* **2013**, *31*, 55–66.
87. Zhao, W.; Du, S. Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4544–4554.
88. Shi, Q.; Zhang, L.; Du, B. Semisupervised discriminative locally enhanced alignment for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 4800–4815.
89. Li, W.; Prasad, S.; Fowler, J.E.; Bruce, L.M. Locality-preserving

dimensionality reduction and classification for hyperspectral image analysis. *IEEE Trans. Geosci. Remote Sens.* **2011**, *50*, 1185–1198.

90. Prasad, S.; Bruce, L.M. Limitations of principal components analysis for hyperspectral target recognition. *IEEE Geosci. Remote Sens. Lett.* **2008**, *5*, 625–629.

91. Wang, Q.; Meng, Z.; Li, X. Locality adaptive discriminant analysis for spectral–spatial classification of hyperspectral images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2077–2081.

92. Zhou, Y.; Peng, J.; Chen, C.P. Dimension reduction using spatial and spectral regularized local discriminant embedding for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 1082–1095.

93. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140.

94. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32.

95. Pal, M. Random forest classifier for remote sensing classification. *Int. J. Remote Sens.* **2005**, *26*, 217–222.

96. Colditz, R. An evaluation of different training sample allocation schemes for discrete and continuous land cover classification using decision tree-based algorithms. *Remote Sens.* **2015**, *7*, 9655–9681.

97. Stefanski, J.; Mack, B.; Waske, B. Optimization of object-based image analysis with random forests for land cover mapping. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 2492–2504.

98. Belgiu, M.; Drăguț, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 24–31.

99. Mountrakis, G.; Im, J.; Ogole, C. Support vector machines in remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 247–259.

100. Cawley, G.C.; Talbot, N.L. Preventing over-fitting during model selection via Bayesian regularisation of the hyper-parameters. *J. Mach. Learn. Res.* **2007**, *8*, 841–861.

101. Cawley, G.C.; Talbot, N.L. On over-fitting in model selection and

subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.* **2010**, *11*, 2079–2107.

102. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790.

103. Fauvel, M.; Chanussot, J.; Benediktsson, J.A.; Sveinsson, J.R. Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles. In Proceedings of the 2007 IEEE International Geoscience and Remote Sensing Symposium (IGARSS2007), Barcelona, Spain, 23–27 July 2007; pp. 4834–4837.

104. Mitra, P.; Shankar, B.U.; Pal, S.K. Segmentation of multispectral remote sensing images using active support vector machines. *Pattern Recognit. Lett.* **2004**, *25*, 1067–1074.

105. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [PubMed]

106. Zhang, H.; Li, Y.; Zhang, Y.; Shen, Q. Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network. *Remote Sens. Lett.* **2017**, *8*, 438–447.

107. Mou, L.; Ghamisi, P.; Zhu, X.X. Fully conv-deconv network for unsupervised spectral-spatial feature extraction of hyperspectral imagery via residual learning. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017; pp. 5181–5184.

108. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*, 12.

109. Guidici, D.; Clark, M. One-Dimensional convolutional neural network land-cover classification of multi-seasonal hyperspectral imagery in the San Francisco Bay Area, California. *Remote Sens.* **2017**, *9*, 629.

110. Wu, H.; Prasad, S. Convolutional recurrent neural networks for hyperspectral data classification. *Remote Sens.* **2017**, *9*, 298.

111. Zhu, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Generative adversarial networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5046–5063.
112. Zhang, L.; Zhang, L.; Du, B. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 22–40.
113. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324.
114. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*; ACM: New York, NY, USA, 2012; pp. 1097–1105.
115. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
116. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.
117. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
118. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
119. Nogueira, K.; Penatti, O.A.; dos Santos, J.A. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognit.* **2017**, *61*, 539–556.
120. Ji, S.; Zhang, C.; Xu, A.; Shi, Y.; Duan, Y. 3D convolutional neural networks for crop classification with multi-temporal remote sensing images. *Remote*

*Sens.* **2018**, *10*, 75.

121. Li, Y.; Zhang, H.; Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67.

122. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251.

123. Sun, H.; Zheng, X.; Lu, X.; Wu, S. Spectral-Spatial Attention Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**.

124. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* **2013**, arXiv:1312.4400.

125. Hu, Y.; Zhang, Q.; Zhang, Y.; Yan, H. A Deep Convolution Neural Network Method for Land Cover Mapping: A Case Study of Qinhuangdao, China. *Remote Sens.* **2018**, *10*, 2053.

126. Castelluccio, M.; Poggi, G.; Sansone, C.; Verdoliva, L. Land use classification in remote sensing images by convolutional neural networks. *arXiv* **2015**, arXiv:1508.00092.

127. Scott, G.J.; England, M.R.; Starms, W.A.; Marcum, R.A.; Davis, C.H. Training deep convolutional neural networks for land–cover classification of high-resolution imagery. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 549–553.

128. Cheng, G.; Han, J.; Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proc. IEEE* **2017**, *105*, 1865–1883.

129. Helber, P.; Bischke, B.; Dengel, A.; Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**.

130. Lee, H.; Kwon, H. Going deeper with contextual CNN for hyperspectral image classification. *IEEE Trans. Image Process.* **2017**, *26*, 4843–4855.

131. Mahdianpari, M.; Salehi, B.; Rezaee, M.; Mohammadimanesh, F.; Zhang, Y. Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery. *Remote Sens.* **2018**, *10*, 1119.



132. Wang, Q.; Liu, S.; Chanussot, J.; Li, X. Scene classification with recurrent attention of VHR remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 1155–1167.
133. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical image computing and computer-assisted intervention (MICCAI), Munich, Germany, 5–9 October 2015; pp. 234–241.
134. Xu, Y.; Wu, L.; Xie, Z.; Chen, Z. Building extraction in very high resolution remote sensing imagery using deep learning and guided filters. *Remote Sens.* **2018**, *10*, 144.
135. Hamaguchi, R.; Fujita, A.; Nemoto, K.; Imaizumi, T.; Hikosaka, S. Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1442–1450.
136. Zhang, Z.; Liu, Q.; Wang, Y. Road extraction by deep residual u-net. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 749–753.
137. Shi, Q.; Liu, X.; Li, X. Road detection from remote sensing images by generative adversarial networks. *IEEE Access* **2017**, *6*, 25486–25494.
138. Mohajerani, S.; Krammer, T.A.; Saeedi, P. Cloud Detection Algorithm for Remote Sensing Images Using Fully Convolutional Neural Networks. *arXiv* **2018**, arXiv:1810.05782.
139. Zhang, Z.; Iwasaki, A.; Xu, G.; Song, J. Cloud detection on small satellites based on lightweight U-net and image compression. *J. Appl. Remote Sens.* **2019**, *13*, 026502.
140. Li, R.; Liu, W.; Yang, L.; Sun, S.; Hu, W.; Zhang, F.; Li, W. DeepUNet: A deep fully convolutional network for pixel-level sea-land segmentation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 3954–3962.
141. Papadomanolaki, M.; Vakalopoulou, M.; Karantzalos, K. A Novel Object-Based Deep Learning Framework for Semantic Segmentation of Very High-

Resolution Remote Sensing Data: Comparison with Convolutional and Fully Convolutional Networks. *Remote Sens.* **2019**, *11*, 684.

142. Rakhlin, A.; Davydow, A.; Nikolenko, S.I. Land Cover Classification From Satellite Imagery With U-Net and Lovasz-Softmax Loss. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 262–266.

143. Shrestha, A.; Mahmood, A. Review of deep learning algorithms and architectures. *IEEE Access* **2019**, *7*, 53040–53065.

144. Liu, P.; Choo, K.K.R.; Wang, L.; Huang, F. SVM or deep learning? A comparative study on remote sensing image classification. *Soft Comput.* **2017**, *21*, 7053–7065.

145. Yu, X.; Wu, X.; Luo, C.; Ren, P. Deep learning in remote sensing scene classification: A data augmentation enhanced convolutional neural network framework. *GIScience Remote Sens.* **2017**, *54*, 741–758.

146. Triguero, I.; García, S.; Herrera, F. Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowl. Inf. Syst.* **2015**, *42*, 245–284.

147. Torrey, L.; Shavlik, J. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*; IGI Global: Hershey, PA, USA, 2010; pp. 242–264.

148. Marmanis, D.; Datcu, M.; Esch, T.; Stilla, U. Deep learning earth observation classification using ImageNet pretrained networks. *IEEE Geosci. Remote Sens. Lett.* **2015**, *13*, 105–109.

149. Zhou, W.; Newsam, S.; Li, C.; Shao, Z. Learning low dimensional convolutional neural networks for high-resolution remote sensing image retrieval. *Remote Sens.* **2017**, *9*, 489.

150. Chen, Z.; Zhang, T.; Ouyang, C. End-to-end airplane detection using transfer learning in remote sensing images. *Remote Sens.* **2018**, *10*, 139.

151. Hong, D.; Yokoya, N.; Xia, G.S.; Chanussot, J.; Zhu, X.X. X-ModalNet: A

- semi-supervised deep cross-modal network for classification of remote sensing data. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 12–23.
152. Nalepa, J.; Myller, M.; Imai, Y.; Honda, K.i.; Takeda, T.; Antoniak, M. Unsupervised Segmentation of Hyperspectral Images Using 3D Convolutional Autoencoders. *arXiv* **2019**, arXiv:1907.08870.
153. Guo, X.; Liu, X.; Zhu, E.; Yin, J. Deep clustering with convolutional autoencoders. In Proceedings of the International Conference on Neural Information Processing (ICONIP), Guangzhou, China, 14–18 November 2017; pp. 373–382.
154. Laso Bayas, J.; See, L.; Fritz, S.; Sturn, T.; Perger, C.; Dürauer, M.; Karner, M.; Moorthy, I.; Schepaschenko, D.; Domian, D.; et al. Crowdsourcing in-situ data on land cover and land use using gamification and mobile technology. *Remote Sens.* **2016**, *8*, 905.
155. Fritz, S.; Fonte, C.; See, L. The role of citizen science in earth observation. *Remote Sens.* **2017**, *9*, 357.
156. Basiri, A.; Haklay, M.; Foody, G.; Mooney, P. Crowdsourced geospatial data quality: Challenges and future directions. *Int. J. Geogr. Inf. Sci.* **2019**, *33*, 1588–1593.
157. Li, G.; Yu, Y. Visual saliency based on multiscale deep features. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5455–5463.
158. Zhao, W.; Du, S. Learning multiscale and deep representations for classifying remotely sensed imagery. *ISPRS J. Photogramm. Remote Sens.* **2016**, *113*, 155–165.
159. Zhang, J. Multi-source remote sensing data fusion: Status and trends. *Int. J. Image Data Fusion* **2010**, *1*, 5–24.
160. Huang, W.; Xiao, L.; Wei, Z.; Liu, H.; Tang, S. A new pan-sharpening method with deep neural networks. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1037–1041.
161. Yuan, Q.; Wei, Y.; Meng, X.; Shen, H.; Zhang, L. A multiscale and

- multidepth convolutional neural network for remote sensing imagery pan-sharpening. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 978–989.
162. Wei, Y.; Yuan, Q.; Shen, H.; Zhang, L. Boosting the accuracy of multispectral image pansharpening by learning a deep residual network. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1795–1799.
163. Vitale, S.; Scarpa, G. A detail-preserving cross-scale learning strategy for CNN-based pansharpening. *Remote Sens.* **2020**, *12*, 348.
164. Ma, X.; Hong, Y.; Song, Y. Super resolution land cover mapping of hyperspectral images using the deep image prior-based approach. *Int. J. Remote Sens.* **2020**, *41*, 2818–2834.
165. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 295–307.
166. Kim, J.; Kwon Lee, J.; Mu Lee, K. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, 27 June–30 June 2016; pp. 1646–1654.
167. Lei, S.; Shi, Z.; Zou, Z. Super-resolution for remote sensing images via local–global combined network. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1243–1247.
168. Liebel, L.; Körner, M. Single-image super resolution for multispectral remote sensing data using convolutional neural networks. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *41*, 883–890.
169. Mei, S.; Yuan, X.; Ji, J.; Zhang, Y.; Wan, S.; Du, Q. Hyperspectral image spatial super-resolution via 3D full convolutional neural network. *Remote Sens.* **2017**, *9*, 1139.
170. Scarpa, G.; Gargiulo, M.; Mazza, A.; Gaetano, R. A CNN-based fusion method for feature extraction from Sentinel data. *Remote Sens.* **2018**, *10*, 236.
171. Lanaras, C.; Bioucas-Dias, J.; Galliani, S.; Baltsavias, E.; Schindler, K. Super-resolution of Sentinel-2 images: Learning a globally applicable deep neural

network. *ISPRS J. Photogramm. Remote Sens.* **2018**, *146*, 305–319.

172. Xu, X.; Li, W.; Ran, Q.; Du, Q.; Gao, L.; Zhang, B. Multisource remote sensing data classification based on convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 937–949.

173. Chen, Y.; Li, C.; Ghamisi, P.; Jia, X.; Gu, Y. Deep fusion of remote sensing data for accurate classification. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1253–1257.

174. Piramanayagam, S.; Saber, E.; Schwartzkopf, W.; Koehler, F. Supervised classification of multisensor remotely sensed images using a deep learning framework. *Remote Sens.* **2018**, *10*, 1429.

175. Audebert, N.; Le Saux, B.; Lefèvre, S. Joint learning from earth observation and OpenStreetMap data to get faster better semantic maps. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 67–75.

176. Gaetano, R.; Ienco, D.; Ose, K.; Cresson, R. A two-branch CNN architecture for land cover classification of PAN and MS imagery. *Remote Sens.* **2018**, *10*, 1746.

177. Audebert, N.; Le Saux, B.; Lefèvre, S. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS J. Photogramm. Remote Sens.* **2018**, *140*, 20–32.

178. Rudner, T.G.; Rußwurm, M.; Fil, J.; Pelich, R.; Bischke, B.; Kopackova, V.; Bilinski, P. Multi3Net: Segmenting Flooded Buildings via Fusion of Multiresolution, Multisensor, and Multitemporal Satellite Imagery. *arXiv* **2018**, arXiv:1812.01756.

179. Zhu, X.; Cai, F.; Tian, J.; Williams, T. Spatiotemporal fusion of multisource remote sensing data: Literature survey, taxonomy, principles, applications, and future directions. *Remote Sens.* **2018**, *10*, 527.

180. Zhong, Y.; Li, W.; Wang, X.; Jin, S.; Zhang, L. Satellite-ground integrated destriping network: A new perspective for EO-1 Hyperion and Chinese hyperspectral satellite datasets. *Remote Sens. Environ.* **2020**, *237*, 111416.

181. Xing, C.; Ma, L.; Yang, X. Stacked denoise autoencoder based feature extraction and classification for hyperspectral images. *J. Sensors* **2016**, *2016*, 3632943.
182. Xie, W.; Li, Y. Hyperspectral imagery denoising by deep learning with trainable nonlinearity function. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1963–1967.
183. Xie, F.; Shi, M.; Shi, Z.; Yin, J.; Zhao, D. Multilevel cloud detection in remote sensing images based on deep learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 3631–3640.
184. Shi, M.; Xie, F.; Zi, Y.; Yin, J. Cloud detection of remote sensing images by deep learning. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 701–704.
185. Lin, G.; Shen, C.; Van Den Hengel, A.; Reid, I. Efficient piecewise training of deep structured models for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3194–3203.
186. Kampffmeyer, M.; Salberg, A.B.; Jenssen, R. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1–9.
187. Kemker, R.; Salvaggio, C.; Kanan, C. Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 60–77.
188. Gavade, A.B.; Rajpurohit, V.S. Sparse-FCM and deep learning for effective classification of land area in multi-spectral satellite images. *Evol. Intell.* **2020**.

## ДОДАТОК 1. ТЕКСТИ ПРОГРАМ

```
# import the necessary packages
from pathlib import Path
```

```
import torch

# base path of the dataset
DATASET_PATH = Path(".") / "train_03"

# define the path to the images and masks dataset
IMAGE_DATASET_PATH = (DATASET_PATH / "images").as_posix()
MASK_DATASET_PATH = (DATASET_PATH / "masks").as_posix()

# define the explosion-crater-detection split
TEST_SPLIT = 0.15

# determine the device to be used for training and evaluation
DEVICE = "cuda" if torch.cuda.is_available() else "cpu"

# determine if we will be pinning memory during data loading
PIN_MEMORY = True if DEVICE == "cuda" else False

# define the number of channels in the input, number of classes,
# and number of levels in the U-Net model
NUM_CHANNELS = 1
NUM_CLASSES = 1
NUM_LEVELS = 3

# initialize learning rate, number of epochs to train for, and the
# batch size
INIT_LR = 1e-3
NUM_EPOCHS = 60
BATCH_SIZE = 512

# define the input image dimensions
INPUT_IMAGE_WIDTH = 64
INPUT_IMAGE_HEIGHT = 64

# define threshold to filter weak predictions
THRESHOLD = 0.5

# define the path to the base output directory
BASE_OUTPUT = Path("output")

BASE_OUTPUT.mkdir(exist_ok=True)
# define the path to the output serialized model, model training
# plot, and testing image paths
MODEL_PATH = (BASE_OUTPUT / "unet.pt").as_posix()
PLOT_PATH = (BASE_OUTPUT / "plot.png").as_posix()
```

```

TEST_PATHS = (BASE_OUTPUT / "test_paths.txt").as_posix()

# import the necessary packages
# import cv2
import numpy as np
from osgeo import gdal
from torch.utils.data import Dataset

class SegmentationDataset(Dataset):
    def __init__(self, image_paths, mask_paths, transforms):
        # store the image and mask filepaths, and augmentation
        # transforms
        self.image_paths = image_paths
        self.mask_paths = mask_paths
        self.transforms = transforms

    def __getitem__(self, index):
        # grab the image path from the current index
        image_path = self.image_paths[index]
        mask_paths = self.mask_paths[index]

        # load the image from disk, swap its channels from BGR to RGB,
        # and read the associated mask from disk in grayscale mode
        image_dataset = gdal.Open(image_path)
        mask_dataset = gdal.Open(mask_paths)

        # image = (image_dataset.ReadAsArray().transpose(1, 2,
0)).astype(np.uint8)
        image = image_dataset.ReadAsArray().transpose(1, 2, 0)
        image = (image / 3000).astype(np.float32)
        # image = (image / 18496).astype(np.float32)
        mask = mask_dataset.ReadAsArray()
        mask = (mask * 255).astype(np.uint8)
        # check to see if we are applying any transformations
        if self.transforms is not None:
            # apply the transformations to both image and its mask
            image = self.transforms(image)
            mask = self.transforms(mask)
        # return a tuple of the image and its mask
        return image, mask

    def __len__(self):
        # return the number of total samples contained in the dataset
        return len(self.image_paths)

```



```

from torch.nn import Module
import torch
from torch.nn import functional as F
from torchvision.ops.bboxes import _box_inter_union

# https://www.kaggle.com/code/bigironsphere/loss-function-library-keras-
# pytorch/notebook

class IoULoss(Module):
    def __init__(self, weight=None, size_average=True):
        super(IoULoss, self).__init__()

    def forward(self, inputs, targets, smooth=1):
        # comment out if your model contains a sigmoid or equivalent
activation layer
        inputs = torch.sigmoid(inputs)

        # flatten label and prediction tensors
        inputs = inputs.view(-1)
        targets = targets.view(-1)

        # intersection is equivalent to True Positive count
        # union is the mutually inclusive area of all labels & predictions
        intersection = (inputs * targets).sum()
        total = (inputs + targets).sum()
        union = total - intersection

        io_u = (intersection + smooth) / (union + smooth)

        return 1 - io_u

class DiceLoss(Module):
    def __init__(self, weight=None, size_average=True):
        super(DiceLoss, self).__init__()

    def forward(self, inputs, targets, smooth=1):
        # comment out if your model contains a sigmoid or equivalent
activation layer
        inputs = torch.sigmoid(inputs)

        # flatten label and prediction tensors
        inputs = inputs.view(-1)

```

```

targets = targets.view(-1)

intersection = (inputs * targets).sum()
dice = (2. * intersection + smooth) / (inputs.sum() + targets.sum() +
smooth)

return 1 - dice

```

```

class DiceBCELoss(Module):
    def __init__(self, weight=None, size_average=True):
        super(DiceBCELoss, self).__init__()

    def forward(self, inputs, targets, smooth=1):
        # comment out if your model contains a sigmoid or equivalent
activation layer
        inputs = torch.sigmoid(inputs)

        # flatten label and prediction tensors
        inputs = inputs.view(-1)
        targets = targets.view(-1)

        intersection = (inputs * targets).sum()
        dice_loss = 1 - (2. * intersection + smooth) / (inputs.sum() +
targets.sum() + smooth)
        bce = F.binary_cross_entropy(inputs, targets, reduction='mean')
        dice_bce = bce + dice_loss

        return dice_bce

```

ALPHA = 0.8

GAMMA = 2

```

class FocalLoss(Module):
    def __init__(self, weight=None, size_average=True):
        super(FocalLoss, self).__init__()

    def forward(self, inputs, targets, alpha=ALPHA, gamma=GAMMA, smooth=1):
        # comment out if your model contains a sigmoid or equivalent
activation layer
        inputs = torch.sigmoid(inputs)

        # flatten label and prediction tensors

```

```

inputs = inputs.view(-1)
targets = targets.view(-1)

# first compute binary cross-entropy
bce = F.binary_cross_entropy(inputs, targets, reduction='mean')
bce_exp = torch.exp(-bce)
focal_loss = alpha * (1 - bce_exp) ** gamma * bce

return focal_loss

```

```
ALPHA = 0.5
```

```
BETA = 0.5
```

```

class TverskyLoss(Module):
    def __init__(self, weight=None, size_average=True):
        super(TverskyLoss, self).__init__()

    def forward(self, inputs, targets, smooth=1, alpha=ALPHA, beta=BETA):
        # comment out if your model contains a sigmoid or equivalent
activation layer
        inputs = F.sigmoid(inputs)

        # flatten label and prediction tensors
        inputs = inputs.view(-1)
        targets = targets.view(-1)

        # True Positives, False Positives & False Negatives
        tp = (inputs * targets).sum()
        fp = ((1 - targets) * inputs).sum()
        fn = (targets * (1 - inputs)).sum()

        tversky = (tp + smooth) / (tp + alpha * fp + beta * fn + smooth)

        return 1 - tversky

```

```
ALPHA = 0.5
```

```
BETA = 0.5
```

```
GAMMA = 1
```

```

class FocalTverskyLoss(Module):
    def __init__(self, weight=None, size_average=True):

```

```

    super(FocalTverskyLoss, self).__init__()

    def forward(self, inputs, targets, smooth=1, alpha=ALPHA, beta=BETA,
gamma=GAMMA):
        # comment out if your model contains a sigmoid or equivalent
activation layer
        inputs = torch.sigmoid(inputs)

        # flatten label and prediction tensors
        inputs = inputs.view(-1)
        targets = targets.view(-1)

        # True Positives, False Positives & False Negatives
        tp = (inputs * targets).sum()
        fp = ((1 - targets) * inputs).sum()
        fn = (targets * (1 - inputs)).sum()

        tversky = (tp + smooth) / (tp + alpha * fp + beta * fn + smooth)
        focal_tversky = (1 - tversky) ** gamma

    return focal_tversky

ALPHA = 0.5 # < 0.5 penalises FP more, > 0.5 penalises FN more
CE_RATIO = 0.5 # weighted contribution of modified CE loss compared to Dice
loss

class ComboLoss(Module):
    def __init__(self, weight=None, size_average=True):
        super(ComboLoss, self).__init__()

    def forward(self, inputs, targets, smooth=1, alpha=ALPHA, beta=BETA,
eps=1e-9):
        # flatten label and prediction tensors
        inputs = inputs.view(-1)
        targets = targets.view(-1)

        # True Positives, False Positives & False Negatives
        intersection = (inputs * targets).sum()
        dice = (2. * intersection + smooth) / (inputs.sum() + targets.sum() +
smooth)

        inputs = torch.clamp(inputs, eps, 1.0 - eps)
        out = - (ALPHA * ((targets * torch.log(inputs)) + ((1 - ALPHA) * (1.0

```

```

- targets) * torch.log(1.0 - inputs)))
    weighted_ce = out.mean(-1)
    combo = (CE_RATIO * weighted_ce) - ((1 - CE_RATIO) * dice)

    return combo

def dice_loss(pred, target, smooth=1.):
    pred = pred.contiguous()
    target = target.contiguous()

    intersection = (pred * target).sum(dim=2).sum(dim=2)

    loss = (1 - ((2. * intersection + smooth) / (pred.sum(dim=2).sum(dim=2) +
target.sum(dim=2).sum(dim=2) + smooth)))

    return loss.mean()
# import the necessary packages
import torch
from torch.nn import BatchNorm2d
from torch.nn import Conv2d
from torch.nn import ConvTranspose2d
from torch.nn import MaxPool2d
from torch.nn import Module
from torch.nn import ModuleList
from torch.nn import ReLU
from torch.nn import functional as F
from torchvision.transforms import CenterCrop

class Block(Module):
    def __init__(self, in_channels, out_channels):
        super().__init__()
        # store the convolution and RELU layers
        self.conv1 = Conv2d(in_channels, out_channels, 3)
        self.bn = BatchNorm2d(out_channels)
        self.relu = ReLU()
        self.conv2 = Conv2d(out_channels, out_channels, 3)

    def forward(self, x):
        # apply CONV => RELU => CONV block to the inputs and return it
        x = self.conv1(x)
        x = self.relu(self.bn(x))

        x = self.conv2(x)

```

```
x = self.relu(self.bn(x))
return x
```

```
class Encoder(Module):
    def __init__(self, channels=(4, 16, 32, 64)):
        super().__init__()
        # store the encoder blocks and maxpooling layer
        self.enc_blocks = ModuleList(
            [Block(channels[i], channels[i + 1])
             for i in range(len(channels) - 1)])
        self.pool = MaxPool2d(2)

    def forward(self, x):
        # initialize an empty list to store the intermediate outputs
        block_outputs = []

        # loop through the encoder blocks
        for block in self.enc_blocks:
            # pass the inputs through the current encoder block, store
            # the outputs, and then apply maxpooling on the output
            x = block(x)
            block_outputs.append(x)
            x = self.pool(x)

        # return the list containing the intermediate outputs
        return block_outputs
```

```
class Decoder(Module):
    def __init__(self, channels=(64, 32, 16)):
        super().__init__()
        # initialize the number of channels, upsampler blocks, and
        # decoder blocks
        self.channels = channels
        self.upconvs = ModuleList(
            [ConvTranspose2d(channels[i], channels[i + 1], 2, 2)
             for i in range(len(channels) - 1)])
        self.dec_blocks = ModuleList(
            [Block(channels[i], channels[i + 1])
             for i in range(len(channels) - 1)])

    def forward(self, x, enc_features):
        # loop through the number of channels
        for i in range(len(self.channels) - 1):
```

```

        # pass the inputs through the upsampler blocks
        x = self.upconvs[i](x)

        # crop the current features from the encoder blocks,
        # concatenate them with the current upsampled features,
        # and pass the concatenated output through the current
        # decoder block
        enc_feat = self.crop(enc_features[i], x)
        x = torch.cat([x, enc_feat], dim=1)
        x = self.dec_blocks[i](x)

    # return the final decoder output
    return x

@staticmethod
def crop(enc_features, x):
    # grab the dimensions of the inputs, and crop the encoder
    # features to match the dimensions
    (_, _, H, W) = x.shape
    enc_features = CenterCrop([H, W])(enc_features)

    # return the cropped features
    return enc_features

class UNet(Module):
    def __init__(self, enc_channels=(4, 16, 32, 64),
                 dec_channels=(64, 32, 16),
                 nb_classes=1, retain_dim=True,
                 out_size=(INPUT_IMAGE_HEIGHT, INPUT_IMAGE_WIDTH)):
        super().__init__()
        # initialize the encoder and decoder
        self.encoder = Encoder(enc_channels)
        self.decoder = Decoder(dec_channels)

        # initialize the regression head and store the class variables
        self.head = Conv2d(dec_channels[-1], nb_classes, 1)
        self.retain_dim = retain_dim
        self.out_size = out_size

    def forward(self, x):
        # grab the features from the encoder
        enc_features = self.encoder(x)
        # pass the encoder features through decoder making sure that
        # their dimensions are suited for concatenation

```

```

    dec_features = self.decoder(enc_features[::-1][0],
                               enc_features[::-1][1:])
    # pass the decoder features through the regression head to
    # obtain the segmentation mask
    map_ = self.head(dec_features)
    # check to see if we are retaining the original output
    # dimensions and if so, then resize the output to match them
    if self.retain_dim:
        map_ = F.interpolate(map_, self.out_size)
    # return the segmentation map_
    return map_

# USAGE
# python train.py
# import the necessary packages
import logging
import os
import time

import matplotlib.pyplot as plt
import torch
from imutils import paths
from sklearn.model_selection import train_test_split
from torch.nn import BCEWithLogitsLoss, CrossEntropyLoss
from torch.optim import Adam, RMSprop
from torch.utils.data import DataLoader
from torchvision import transforms
from tqdm import tqdm

logging.basicConfig(
    filename="logs.log",
    filemode='a',
    format='%(asctime)s | %(levelname)s | %(message)s',
    level=logging.INFO
)

# load the image and mask filepaths in a sorted manner
image_paths = sorted(list(paths.list_images(IMAGE_DATASET_PATH)))
mask_paths = sorted(list(paths.list_images(MASK_DATASET_PATH)))
print(image_paths)
# partition the data into training and testing splits using 85% of
# the data for training and the remaining 15% for testing
train_images, test_images, train_masks, test_masks = train_test_split(
    image_paths, mask_paths, test_size=TEST_SPLIT, random_state=42)

```



```

# write the testing image paths to disk so that we can use them
# when evaluating/testing our model
logging.info("saving testing image paths...")
f = open(TEST_PATHS, "w")
f.write("\n".join(test_images))
f.close()

# define transformations
transforms = transforms.Compose([
    # transforms.ToPILImage(),
    # transforms.Resize((INPUT_IMAGE_HEIGHT,
    #                     INPUT_IMAGE_WIDTH)),
    transforms.ToTensor()
])

# create the train and explosion-crater-detection datasets
train_ds = SegmentationDataset(image_paths=train_images,
                              mask_paths=train_masks,
                              transforms=transforms)
test_ds = SegmentationDataset(image_paths=test_images,
                              mask_paths=test_masks,
                              transforms=transforms)

logging.info(f"found {len(train_ds)} examples in the training set...")
logging.info(f"found {len(test_ds)} examples in the testing set...")

# create the training and explosion-crater-detection data loaders
train_loader = DataLoader(train_ds, shuffle=True,
                          batch_size=BATCH_SIZE, pin_memory=PIN_MEMORY,
                          num_workers=os.cpu_count())
test_loader = DataLoader(test_ds, shuffle=False,
                         batch_size=BATCH_SIZE, pin_memory=PIN_MEMORY,
                         num_workers=os.cpu_count())

# initialize our UNet model
UNET = UNet().to(DEVICE)

# initialize loss function and optimizer
loss_func = BCEWithLogitsLoss()
# loss_func = IoULoss()
opt = RMSprop(UNET.parameters(), lr=INIT_LR)

# calculate steps per epoch for training and explosion-crater-detection set
train_steps = len(train_ds) // BATCH_SIZE
test_steps = len(test_ds) // BATCH_SIZE

```

```

# initialize a dictionary to store training history
H = {"train_loss": [], "test_loss": []}

# loop over epochs
logging.info("training the network...\n\n")
startTime = time.time()

for epoch in range(NUM_EPOCHS):
    # set the model in training mode
    unet.train()

    # initialize the total training and validation loss
    total_train_loss = 0
    total_test_loss = 0

    tqdm_data = tqdm(train_loader, desc=f'Training (epoch #{epoch})',
total=int(len(train_loader)))
    # loop over the training set
    for (i, (x, y)) in enumerate(tqdm_data):
        # send the input to the device
        (x, y) = (x.to(DEVICE), y.to(DEVICE))

        # perform a forward pass and calculate the training loss
        pred = unet(x)
        loss = loss_func(pred, y)

        # first, zero out any previously accumulated gradients, then
        # perform backpropagation, and then update model parameters
        opt.zero_grad()
        loss.backward()
        opt.step()

        # add the loss to the total training loss so far
        total_train_loss += loss
        tqdm_data.set_postfix(loss=(total_train_loss / ((i + 1) *
train_loader.batch_size)))

    # switch off autograd
    with torch.no_grad():
        # set the model in evaluation mode
        unet.eval()

        tqdm_data = tqdm(test_loader, desc=f'Validation (epoch #{epoch})',
total=int(len(test_loader)))

```

```

# loop over the validation set
for i, (x, y) in enumerate(tqdm_data):
    # send the input to the device
    (x, y) = (x.to(DEVICE), y.to(DEVICE))

    # make the predictions and calculate the validation loss
    pred = unet(x)
    total_test_loss += loss_func(pred, y)
    tqdm_data.set_postfix(loss=(total_test_loss / ((i + 1) *
test_loader.batch_size)))

# calculate the average training and validation loss
avg_train_loss = total_train_loss / train_steps
avg_test_loss = total_test_loss / test_steps

# update our training history
H["train_loss"].append(avg_train_loss.cpu().detach().numpy())
H["test_loss"].append(avg_test_loss.cpu().detach().numpy())

# print the model training and validation information
logging.info(f"Train loss: {avg_train_loss:.6f}, Test loss:
{avg_test_loss:.4f}")

# display the total time needed to perform the training
end_time = time.time()
logging.info(f"total time taken to train the model: {end_time -
startTime:.2f}s")

# plot the training loss
plt.style.use("ggplot")
plt.figure()
plt.plot(H["train_loss"], label="train_loss")
plt.plot(H["test_loss"], label="test_loss")
plt.title("Training Loss on Dataset")
plt.xlabel("Epoch #")
plt.ylabel("Loss")
plt.legend(loc="lower left")
plt.savefig(PLOT_PATH)

# serialize the model to disk
torch.save(unet, MODEL_PATH)

# USAGE
# python predict.py

```

```

import logging
from pathlib import Path

import matplotlib.pyplot as plt
import numpy as np
import torch
from osgeo import gdal

# import the necessary packages

logging.basicConfig(
    filename="logs.log",
    filemode='a',
    format='%(asctime)s | %(levelname)s | %(message)s',
    level=logging.INFO
)

def prepare_plot(orig_image, orig_mask, pred_mask):
    # initialize our figure
    figure, ax = plt.subplots(nrows=1, ncols=3, figsize=(10, 10))

    # plot the original image, its mask, and the predicted mask
    ax[0].imshow(orig_image)
    ax[1].imshow(orig_mask)
    ax[2].imshow(pred_mask)

    # set the titles of the subplots
    ax[0].set_title("Image")
    ax[1].set_title("Original Mask")
    ax[2].set_title("Predicted Mask")

    # set the layout of the figure and display it
    figure.tight_layout()
    figure.show()

def make_predictions(model, image_path):
    # set model to evaluation mode
    model.eval()

    # turn off gradient tracking
    with torch.no_grad():
        # load the image from disk, swap its color channels, cast it
        # to float data type, and scale its pixel values

```

```

image_dataset = gdal.Open(image_path)
image = image_dataset.ReadAsArray().transpose(1, 2, 0)
image = (image / 3000).astype(np.float32)

# resize the image and make a copy of it for visualization
orig = image.copy()

# find the filename and generate the path to ground truth
# mask
filename = Path(image_path).name
ground_truth_path = (Path(MASK_DATASET_PATH) / filename).as_posix()

# load the ground-truth segmentation mask in grayscale mode
# and resize it
mask_dataset = gdal.Open(ground_truth_path)
gt_mask = mask_dataset.ReadAsArray()
gt_mask = (gt_mask * 255).astype(np.uint8)

# make the channel axis to be the leading one, add a batch
# dimension, create a PyTorch tensor, and flash it to the
# current device
image = np.transpose(image, (2, 0, 1))
image = np.expand_dims(image, 0)
image = torch.from_numpy(image).to(DEVICE)

# make the prediction, pass the results through the sigmoid
# function, and convert the result to a NumPy array
pred_mask = model(image).squeeze()

# pred_mask -= pred_mask.min(1, keepdim=True)[0]
# pred_mask /= pred_mask.max(1, keepdim=True)[0]
pred_mask = torch.sigmoid(pred_mask)
pred_mask = pred_mask.detach().cpu().numpy()

# filter out the weak predictions and convert them to integers
pred_mask = (pred_mask > THRESHOLD) * 255
pred_mask = pred_mask.astype(np.uint8)
# prepare a plot for visualization
prepare_plot(orig, gt_mask, pred_mask)

# load the image paths in our testing file and randomly select 10
# image paths
logging.info("loading up explosion-crater-detection image paths...")
image_paths = open(TEST_PATHS).read().strip().split("\n")

```

```
image_paths = np.random.choice(image_paths, size=20)

# load our model from disk and flash it to the current device
logging.info("load up model...")
UNET = torch.load(MODEL_PATH).to(DEVICE)

# iterate over the randomly selected explosion-crater-detection image paths
for path in image_paths:
    # make predictions and visualize the results
    make_predictions(UNET, path)
```