

Міністерство освіти і науки України  
Київський національний університет імені Тараса Шевченка  
ННІ «Інститут геології»  
Кафедра геоінформатики

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**  
**спеціальність 103 – Науки про Землю**  
**освітня програма «Геоінформатика»**

ТЕМА: «Оцінка наслідків руйнування інфраструктури будівель внаслідок сейсмічних подій за допомогою методів машинного навчання»

Виконала



студентка 2-го курсу магістратури  
кафедри геоінформатики  
Брудько Катерина Олександрівна

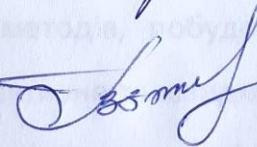
Науковий керівник



доцент кафедри геоінформатики  
кандидат фізико-математичних наук  
Демидов Всеволод Кирилович

Робота рекомендується до захисту (протокол № 8 засідання кафедри геоінформатики від 15 травня 2023 р.

Завідувач кафедри



професор  
доктор технічних наук  
Зацерковний Віталій Іванович

Київ – 2023

## РЕФЕРАТ

магістерської кваліфікаційної роботи

**Брудько Катерини Олександрівни**

на тему: «Оцінка наслідків руйнування інфраструктури будівель внаслідок сейсмічних подій за допомогою методів машинного навчання»

Спеціальність **103 Науки про Землю**

Робота на 112 аркушах складається з 4 розділів і містить 1 таблицю, 25 рисунків, 5 додатків. При підготовці роботи використовувалися матеріали з 60 джерел.

**Актуальність.** Оцінка місцеположення, площа та об'єм пошкоджень є важливою інформацією для відновлювальних робіт у зоні катастрофи. Технології дистанційного зондування Землі відіграють значну роль у отриманні інформації про контури та тип пошкодження інфраструктури будівель у сейсмічно активних зонах. Зі зростанням об'ємів доступних даних дистанційних зондувань, виникає потреба в автоматизації процесів класифікації об'єктів на знімках, зокрема виділення контурів будівель та стану їх пошкоджень в сейсмічно активних зонах. Шляхом використання автоматизованих методів, побудованих на основі згорткових нейронних мереж, можна ефективно та швидко отримувати інформацію на основі даних дистанційних зондувань, виділяючи контури об'єктів будівельної інфраструктури, оцінювати їх площу та відносити їх до певного класу об'єктів з високою точністю.

**Мета роботи.** Розробка моделей згорткових нейронних мереж для автоматичного визначення контурів житлової інфраструктури та оцінки наслідків руйнування, спричинених землетрусами, у районах з підвищеною сейсмічною активністю

**Розв'язувані в роботі задачі.** Підготовка, аналіз та обробка даних для тренування та навчання згорткових нейронних мереж, вибір архітектури та гіперпараметрів моделей, тренування моделей, оцінка точності моделей, постобробка результатів моделювання.

**Наукові результати.** Створено 2 моделі згорткових нейронних мереж для 2-х окремих задач: виділення контурів будинків та класифікації будинків. Результати оцінені на площі 0.13 km<sup>2</sup> де точність моделі для виділення контурів будинків складає 98%, точність результату класифікації пошкоджених будівель складає 89% на площі 5.98 km<sup>2</sup> міста Аматрісе (Італія). В результаті отримано 824 контури будинків з яких 406 були ідентифіковані як будинки, які мають пошкодження загальною площею 0.07 km<sup>2</sup>

**Наукова новизна** полягає в розробці моделей згорткових нейронних мереж для трьохканальних знімків (RGB) для отримання актуальної інформації про контури будинків та стан їхнього пошкодження.

**Практичне значення** роботи полягає в: розробці методики створення моделі семантичної сегментації та класифікації об'єктів житлової інфраструктури, для швидкого реагування та оцінки наслідків руйнування за даними ДЗЗ у зонах з підвищеною сейсмічною активністю.

**Апробація.** Дана робота пройшла апробацію на XVI International Scientific Conference “Monitoring of Geological Processes and Ecological Condition of the Environment” 15–18 November 2022, Kyiv, Ukraine з темою доповіді: Automatic

recognition and damage evaluation of building infrastructure in seismic active zones using machine learning.

### **Анотація**

У роботі представлено використання методів глибинного навчання для автоматичного розпізнавання пошкоджених будівель, що дозволяє швидко і просторово оцінити вплив природних небезпек, таких як землетруси, тим самим прискорюючи реагування на надзвичайні ситуації. У зв'язку зі складним характером виявлення ознак пошкоджень та зростаючою популярністю методів машинного навчання, актуальними є дослідження з використання нейронних мереж глибокого навчання, особливо згорткових нейронних мереж, для автоматичного розпізнавання та оцінки пошкоджень будівельної інфраструктури в сейсмічно активних зонах.

Метою цієї роботи є створення моделей для сегментації площі будівель та класифікації пошкоджень будівель для оцінки впливу природних небезпек. Для даного дослідження як вихідні данні були використані супутникові знімки до і після землетрусу, що стався в Центральній Італії місто Амастріче 24 серпня 2016 року о 01:36 (UTC) потужністю 6,2 МВт. Результати дослідження показали, що нейронна мережа, навчена на репрезентативному наборі зразків, може розрізняти як пошкоджені, так і неушкоджені типи будівельної інфраструктури. Результати класифікації можуть представляти набори даних для отримання статусу пошкоджень.

**Ключові слова** ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, СЕМАНТИЧНА СЕГМЕНТАЦІЯ, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, ДИСТАНЦІЙНЕ ЗОНДУВАННЯ ЗЕМЛІ.

Kateryna Brudko

Damage evaluation of building infrastructure in seismically active areas using machine learning

### **Abstract**

This work presents using of deep learning techniques to automatically recognize damaged buildings enables rapid and spatially extensive assessment of the impact of natural hazards, such as earthquakes, thereby accelerating emergency response. Due to the complex nature of indicating damage signs and increasing popularity of machine learning techniques, research has started on the use of deep learning neural networks, especially on convolutional neural network for automatic recognition and damage evaluation of building infrastructure in seismic active zones.

This work aims to create models for: buildings footprint segmentation and building damage classification to estimate impact of natural hazards. As source remote sensing data was used geoimages before and after earthquake that struck the Central Italy area. The results of the study showed that a neural network trained on a representative set of samples can distinguish both types of damaged and undamaged building infrastructure. Classification results can represent datasets to obtain damages status.

**Key words** CONVOLUTION NEURAL NETWORKS, SEMANTIC SEGMENTATION, IMAGE CLASSIFICATION, REMOTE SENSING

## ЗМІСТ

|   |    |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І<br>ТЕРМІНІВ ..... | 8  |
| ВСТУП .....   | 9  |
| 1. ОГЛЯД ПОСТАВЛЕНОЇ ПРОБЛЕМАТИ .....                                       | 13 |
| 1.1 Основні відомості про об'єкт дослідження .....                          | 13 |
| 1.2 Основні геологічні відомості територію дослідження .....                | 15 |
| 1.3 Огляд наукових матеріалів .....   | 17 |
| 1.4 Популярні методи машинного навчання .....                               | 29 |
| 1.4.1 Метод опорних векторів .....  | 30 |
| 1.4.2 Дерева рішень .....   | 32 |
| 1.4.3 Random Forest.....  | 33 |
| 1.4.4 Gradient boosting .....   | 34 |
| 1.5 Основні відомості про нейронні мережі.....                              | 35 |
| 1.5.1 Фактори популярності нейронних мереж.....                             | 35 |
| 1.5.2 Основні елементи архітектури нейронної мережі .....                   | 36 |
| 1.5.3 Алгоритм оберненого розповсюдження помилки .....                      | 38 |
| 2. ТЕХНОЛОГІЧНІ РІШЕННЯ.....  | 45 |
| 2.1 Застосування згорткових нейронних мереж .....                           | 45 |
| 2.2 Огляд основних шарів згорткової нейронної мережі.....                   | 46 |
| 2.3 Повнозв'язний шар нейронної мережі .....                                | 47 |
| 2.4 Convolution Layer .....   | 49 |
| 2.5 Pooling Layer .....   | 51 |
| 2.6 Dropout Layer .....   | 53 |
| 2.7 Batch Normalization Layer .....   | 55 |
| 2.8 Функції втрат.....  | 56 |
| 2.9 Оптимізатор.....  | 57 |
| 2.10 Метрики оцінки якості моделей. Confusion Matrix.....                   | 58 |

|   |    |
|---|----|
| 2.11 Intersection Over Union.....                             | 59 |
| 2.12 Аналіз та розширення даних (аугментація) .....           | 60 |
| 2.13 Вибір мови програмування.....                            | 61 |
| 2.14 Огляд існуючих бібліотек для глибокого навчання .....    | 63 |
| 2.14.1 TensorFlow .....                                       | 64 |
| 2.14.2 PyTorch.....   | 65 |
| 2.14.3 Keras .....  | 66 |
| 3. ОГЛЯД СУЧАСНИХ АРХІТЕКТУР ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ..... | 68 |
| 3.1 Архітектура LeNet .....                                   | 68 |
| 3.2 Архітектура AlexNet.....                                  | 69 |
| 3.3 Архітектура EfficientNet.....                             | 70 |
| 3.4 Архітектура ResNet .....                                  | 71 |
| 3.5 Архітектури класу UNet .....                              | 73 |
| 4. ПРАКТИКА РЕАЛІЗАЦІЇ ПРОЕКТУ .....                          | 76 |
| 4.1 Джерела даних для тренування.....                         | 76 |
| 4.2 Приведення растрових даних.....                           | 78 |
| 4.3 Створення навчальної вибірки .....                        | 78 |
| 4.4 Підготовка та створення моделі .....                      | 82 |
| 4.5 Навчання моделі .....                                     | 87 |
| 4.6 Реалізація алгоритму сегментації.....                     | 89 |
| 4.7 Реалізація алгоритму класифікації .....                   | 92 |
| ВИСНОВКИ .....  | 96 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ .....                | 99 |

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

|                 |                                 |
|-----------------|---------------------------------|
| ЗНМ/CNN         | Згортова нейронна мережа        |
| ДЗЗ             | Дистанційне зондування Землі    |
| км <sup>2</sup> | Кілометри квадратні             |
| VHR             | Very high resolution            |
| SAR             | Радар із синтезованою апертурою |
| GPU             | Графічний процесор              |
| SGD             | Stochastic Gradient Descent     |
| U-Net           | Архітектура ЗНМ для сегментації |
| IoU             | Intersection Over Union         |

## ВСТУП

**Актуальність.** Землетруси є одними з найбільш катастрофічних стихійних лих, що впливають на людство. Локалізація катастрофи, створення актуальних цифрових карт та оцінка збитків можуть значно прискорити реагування рятувальних, гуманітарних та урядових служб. Сейсмічно активні території мають особливо потребу в актуальних даних наслідків землетрусів, особливо на проблемно-доступних ділянках. Саме тому методи дистанційного зондування відіграють важливу роль в отриманні актуальної інформації про пошкодження будівель завдяки їх безконтактності, низькій вартості та можливості швидкого реагування.

Оцінка місцеположення, площа та об'єм пошкоджень є важливою інформацією для реконструктивних робіт у зоні катастрофи. Технології дистанційного зондування Землі відіграють значну роль у отриманні інформації про контури та тип пошкодження будівельної інфраструктури у сейсмічно активних зонах. Зі зростанням доступних розмірів даних дистанційного зондування, виникає потреба в автоматизації процесів класифікації об'єктів на знімках, зокрема виділення контурів будівель та стану їх пошкоджень в сейсмічно активних зонах.

Наприклад в той час, коли відбувся масштабний землетрус потужністю 6,2 МВт в Центральній Італії, місто Амастріче, карта пошкоджень будівель на основі інспекційного обстеження, проведеного у вересні 2016 року, була розроблена більше місяця потому як сталася сейсмічна подія (Carrucci S. and all, 2020).

Використання методів глибокого навчання для автоматичного розпізнавання пошкоджених будівель дозволяє швидко і просторово оцінити вплив природних катастроф, тим самим прискорюючи реагування на надзвичайні ситуації. Шляхом використання автоматизованих методів, побудованих на основі згорткових нейронних мереж, можна ефективно та швидко отримувати інформацію на основі даних дистанційних зондувань, виділяючи контури об'єктів будівельної інфраструктури, оцінювати їх площу та відносити їх до певного класу з високою точністю.

Автоматизована класифікація пошкоджень будівель за допомогою дистанційного зондування може бути корисною в реагуванні на катастрофи. Найбільш корисною вона може бути на етапі раннього реагування, тобто в період від 24 до 168 годин після стихійного лиха, коли оцінка збитків і потреб ще не є доступною. Автоматизована оцінка збитків може бути надана дуже швидко, протягом декількох годин після того, як стануть доступними знімки дистанційного зондування (Brudko K. and all, 2022). Автоматизована класифікація пошкоджень будівель привертає все більше уваги протягом останніх кількох років як з боку практиків, так і з боку дослідників. Моделі використовують або дані дистанційного зондування, або геопросторові дані, або їх комбінації (Valentijn, T. and al., 2022).

Методи дешифрування та візуалізації є загальними підходами до створення карт пошкоджень будівель з використанням супутникових або аерофотознімків (Tong, X. і ін., 2012). Однак ручний візуальний метод класифікації є неефективним та працезатратним при плануванні рятувальних та реконструктивних робіт (Rastiveis, H. і ін., 2013). Натомість, автоматичні

методи, засновані на згорткових нейронних мережах, дозволяють ефективно отримувати інформацію з супутникових знімків: виявляти контури об'єктів, оцінювати їх площу та класифікувати їх у відповідні категорії.

Шаблони, отримані CNN, є інваріантними щодо передачі. Вивчивши певний шаблон в одному місці, згорткова нейронна мережа може розпізнавати його всюди. Повністю підключеній мережі доведеться заново вивчити шаблон, якщо він з'явиться в іншому місці. Це підвищує ефективність згорткових нейронних мереж у задачах обробки зображень. Такі мережі потребують меншої кількості навчальних вибірок для отримання інформації з узагальненням та можуть вивчати просторові ієрархії шаблонів. Перший згортковий шар буде вивчати невеликі локальні шаблони, такі як дуги, ребра, другий - більші шаблони, що складаються з функцій, повернутих першим шаром. Це дозволяє CNN ефективно вивчати все більш складні та абстрактні візуальні представлення (François Chollet, 2018).

**Об'єкт та предмет дослідження.** Об'єктом дослідження є дані дистанційного зондування для міста Амастріче (Центральна Італія).

**Предметом дослідження** є застосування згорткових нейронних мереж в задачах виділення контурів будівель та класифікації стану їх пошкодження.

**Мета роботи полягає у** створенні моделей для сегментації контурів будівельної інфраструктури та класифікації будівель для оцінки наслідків руйнування інфраструктури будівель внаслідок сейсмічних подій.

**Матеріали для дослідження.** Частина матеріалів була отримана у рамках написання магістерської кваліфікаційної роботи від компанії ТОВ "TVIS", а саме шейпфайл контурів будівельної інфраструктури міста Амастріче

актуальний на 01.01.2016. Як матеріали дослідження були використані, зокрема, дані дистанційного зондування Землі, а саме космознімки з роздільною здатністю 0.3 метра за 21.05.2016 (до землетрусу) та за 07.07.2017 (після землетрусу), що були надані ресурсом SASPlanet у рамках використання студентської ліцензії.

## 1. ОГЛЯД ПОСТАВЛЕНОЇ ПРОБЛЕМАТИ

### 1.1 Основні відомості про об'єкт дослідження

Системи швидкого реагування на землетруси відіграють важливу роль у швидкому і точному визначенні та оцінці просторових проявів сейсмічної небезпеки (наприклад, зсувів, зрідження) і сейсмічних наслідків (наприклад, пошкодження будівель). Ця інформація допомагає спрямовувати заходи реагування та відновлення в ті райони, які найбільше потребують критичної допомоги. Деякі з існуючих глобальних або регіональних програм швидкої оцінки сейсмічної небезпеки та наслідків включають GDACS - Глобальну систему оповіщення і координації при стихійних лихах, PAGER - Оперативну оцінку глобальних землетрусів для реагування від Геологічної служби США (USGS) та NERIES-ELER - Мережу дослідницьких інфраструктур для європейської сейсмології (Susu Xu and all, 2022).

24 серпня 2016 року в Центральній Італії стався перший землетрус з довготривалої серії. Основний поштовх стався 24 серпня в околицях міста Аматріче, Центральна Італія. Більшість будівель було пошкоджено, і одразу після землетрусу Національна служба цивільного захисту Італії (DPC) розпочала координувати надзвичайні та пост-надзвичайні заходи. Останні включали геологічні та геотехнічні дослідження для сейсмічного мікрорайонування, проведені Центром сейсмічного мікрорайонування (CMS), а також створення спеціальної робочої групи з управління завалами в місті Аматріче (Carrucci S. and all, 2020).

У 2016-17 роках Центральна Італія постраждала від складної сейсмічної послідовності на Апеннінах, що охопила значну частину адміністративних територій областей Лаціо, Марке, Умбрія та Абруццо.

24 серпня 2016 року о 01:36 (UTC) землетрус потужністю 6,2 МВт стався в Центральній Італії між муніципалітетами Амастріче і Арквата-дель-Тронто (рис. 1.1), спричинивши близько 300 жертв і понад 35 000 бездомних (Rossi, A., 2019; Amatrice, Norcia, Visso Seismic Sequence, INGV, Італійський національний інститут геофізики і вулканології, 2020).

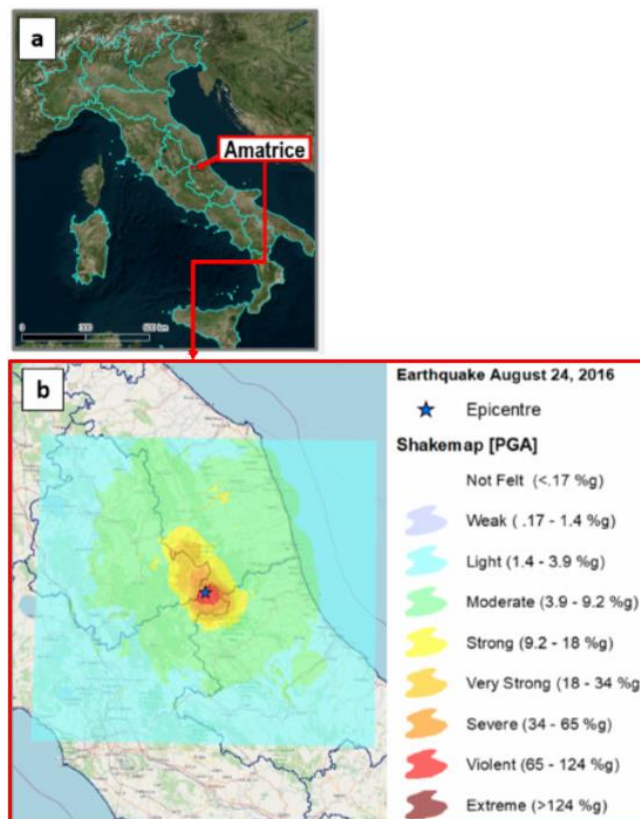


Рис.1.1 а) Розташування Амастріче (Центральна Італія) (Сарруссі S. and all, 2020); б) Shakemap землетрусу, що стався 24 серпня 2016 року ( Amatrice, Norcia, Visso Seismic Sequence, INGV, Італійський національний інститут геофізики і вулканології, 2020);

Посилаючись на італійську мережу StrongMotion (Italian Strong Motion Network, 2020) найближча акселерометрична станція АМТ розташована приблизно в 400 м на північ від історичного центру Амастріче (рис.1.2).



Рис.1.2 Розташування та фото сейсмічної станції АМТ (Italian Strong Motion Network, 2020)

Внаслідок сейсмічної катастрофи 386 людей загинули, 386 отримали поранення і близько 4800 залишилися без даху над головою. Більшість жертв були в районах Амастріче, Аккумолі та Арквата-дель Тронто. У цих муніципалітетах були зафіксовані значні пошкодження інфраструктури будинків, повідомлялося про сильні руйнування і обвалення, зокрема, житлових будинків (Fiorentino G. and all, 2018).

## 1.2 Основні геологічні відомості територію дослідження

Амастріче знаходиться в центральній італійській частині Апеннінського ланцюга (рисунок 1.1, а). Ця територія, починаючи з верхнього пліоцену, цікава екстенсивною тектонікою з активними сейсмогенними розломами, що охоплюють весь хребет, орієнтований на північний захід-південний схід (Pizzi, A., Galadini, F., 2009).

Вивчення сейсмічних подій, що відбулися в минулому, а також нещодавні дослідження сейсмічності та сейсмотектоніки показали, що район Амастріче має високу сейсмічну небезпеку на національному рівні (PGA 0,25-0,275 g з ймовірністю перевищення 10% через 50 років) (Fiorentino G. and all, 2018).

Басейн Амастріче є морфоструктурною западиною, заповненою міоценовими силікатними відкладами лагінської свити, що перекриті четвертинними континентальними утвореннями. Лагінська свита складається з чергування шарів пісковиків і алевролітів і являє собою геологічну основу території. Місто розташоване на флювіальній терасі в межах долини річки Тронто. Ця тераса в основному складається з гравію та пісків, що безпосередньо перекривають корінну породу (Regione Lazio, 2018).

Як і в багатьох інших селах Центральної Італії, міська забудова була розташована на вершині пагорба і в безпосередній близькості до країв тераси, оскільки ця частина гори дозволяла краще захищатися від нападів і контролювати навколишні території, а також захиститись від повеней у річковій долині. Але ці райони також схильні до ефекту посилення підземних рухів, що виникають під час землетрусів. Приклад геологічного розрізу, виконаного через хребет Амастріче наведено на рисунку 1.3 (Carrucci S. and all, 2020).

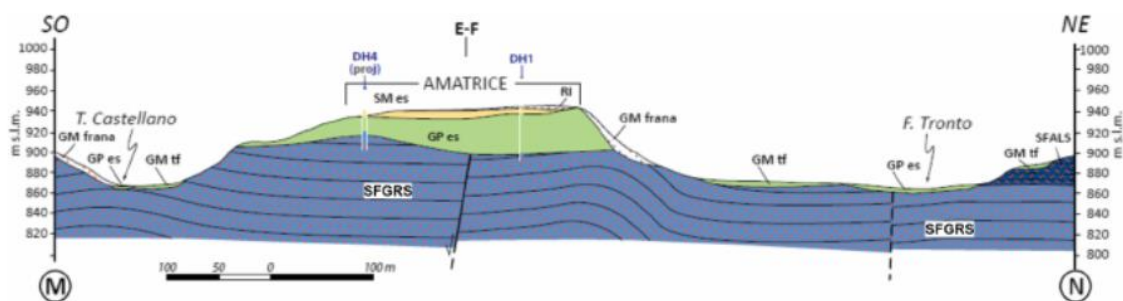


Рис.1.3 - Геологічний профіль SO-NE хребта Аматріс. Покривні ландшафти: RI - антропогенні відклади; GPes - змішані гравії та піски в алювіальному віалі; GMtf - змішані гравії, піски та мули в терасових алювіальних відкладах; SMeS - мулисті піски в алювіальних відкладах. Геологічна порода: SFALS - чергування контрастних літотипів, шаруватих і змінених або тріщинуватих; SFGRS - шаруваті, зернисто зцементовані літотипи, змінені або тріщинуваті (Regione Lazio, Microzonazione Sismica Livello 3, 2018)

### 1.3 Огляд наукових матеріалів

З розвитком методів дистанційного зондування Землі, використання супутникових та аерофотознімків стало поширеним для різних задач, включаючи отримання детальної просторово-часової інформації в цільових зонах. Це також відноситься до моніторингу наслідків природних катастроф, оскільки знімки дистанційного зондування часто не вимагають значних польових робіт, що особливо важливо для районів, постраждалих від землетрусів, де отримання актуальної інформації може бути складним.

Методи дистанційного зондування Землі (ДЗЗ) часто застосовуються у розробці ефективних планів реагування на надзвичайні ситуації при різних

масштабах катастроф. Оскільки своєчасна інформація про площу, ступінь та кількість пошкоджень є ключовою для початку планування реконструктивних робіт (Tong, X. і ін., 2012).

ДЗЗ може забезпечити широкомасштабну та швидку оцінку пошкоджень на великих територіях, дозволяючи швидше реагувати на надзвичайні ситуації та спрямовувати допомогу туди, де вона найбільше потрібна. Знімки ДЗЗ можуть також надати об'єктивну інформацію про розміри та типи пошкоджень, що допомагає в розробці ефективних планів реконструкції та відновлення після землетрусу. Однак важливо брати до уваги і обмеження, пов'язані з роздільною здатністю знімків, часовими обмеженнями і необхідністю інтеграції з іншими джерелами інформації для повноцінної оцінки ситуації після землетрусу.

Численні дослідження в даній області продемонстрували, що дистанційне зондування (ДЗ) та геоінформаційні технології (ГІС) є ефективним інструментом для швидкого моніторингу пошкоджених будівель у міських регіонах після землетрусу (Menderes, A. and all, 2015).

Якщо порівнювати сьогоднішню доступність супутникових знімків із ситуацією десятирічної давнини, то кількість, своєчасність і доступність супутникових знімків, що висвітлюють певну кризову ситуацію чи катастрофу, суттєво покращилася (Voigt, S. and all, 2007). З'явилося більше супутникових місій та комерційних постачальників послуг, що пропонують широкий вибір супутникових знімків з різною роздільною здатністю, спектральними характеристиками та частотою оновлення. Це дозволяє отримувати більше актуальної інформації про кризові ситуації та катастрофи, включаючи

землетруси. Зокрема, широкий доступ до супутникових знімків дозволяє безкоштовно отримувати зображення пошкоджених районів, оцінювати зміни в ландшафті та визначати розміри та характер пошкоджень.

Окрім того, розвиток технологій обробки даних дистанційного зондування сприяє швидкому аналізу та обробці великих обсягів гео-даних за відносно короткі проміжки часу, що дозволяє отримувати інформацію, необхідну для прийняття рішень оперативних служб та служб швидкого реагування.

Загалом, сучасні супутникові знімки стали незамінним інструментом для моніторингу, оцінки та відновлення після катастроф, забезпечуючи широкий доступ до важливої інформації та сприяючи ефективнішій реакції на надзвичайні ситуації.

Для оцінки пошкоджень використовуються різноманітні джерела даних, що включають аерофотознімки, оптичні супутникові знімки, SAR (Радар із синтезованою апертурою) та LiDAR (лазерне сканування). Кожен з цих джерел надає унікальну інформацію, яка допомагає у визначенні характеру пошкоджень та їх розміру.

Аерофотознімки здійснюються з використанням літаків або дронів і надають високу роздільну здатність та деталізацію зображень. Вони часто використовуються для оцінки пошкоджень будівель та інфраструктури на місцевому рівні. Оптичні супутникові знімки надають широкий обсяг інформації про зміни в ландшафті та пошкодження будівель. Вони корисні для оцінки великих територій та забезпечують можливість вивчення змін у часі. SAR знімки отримуються за допомогою радіосигналів, що відбиваються від

поверхні Землі. Вони дозволяють проникати через хмари та забезпечують інформацію про пошкодження під поверхнею, такі як зсуви ґрунту. LiDAR сканування здійснюється за допомогою лазерних променів, що вимірюють відстань до поверхні. Це дозволяє отримувати детальні дані про висоту об'єктів, включаючи будівлі та рельєф місцевості (Світличний О.О., Плотницький С.В. , 2006).

Комбінація цих джерел даних дозволяє отримати більш повне та точне уявлення про пошкодження внаслідок катастрофи. Дані дистанційного зондування здатні отримати відносно точну інформацію про пошкодження будівлі. Зображення дистанційного зондування з високою роздільною здатністю можуть генерувати карти пошкодження будівлі за будівлею шляхом інтерпретації стану пошкодження (Min Ji, Lanfa Liu, 2018).

Пошкодження будівель можна виявити, використовуючи лише дані після події за допомогою появи зображень дистанційного зондування з дуже високою роздільною здатністю (VHR), які можуть надати детальні текстурні та просторові зображення особливостей пошкоджених цілей. Для оцінки збитку після землетрусу можна застосувати широкий спектр методів дистанційного зондування, включаючи оптичні супутникові зображення, SAR та LiDAR. Зі швидким покращенням просторової розрізненості супутникових оптичних датчиків (таких як WorldView-4, який має 0,31 м/піксель у панхроматичному діапазоні), використання оптичних даних є багатообіцяючим підходом для виявлення пошкоджень внаслідок землетрусів (Min Ji, Lanfa Liu, 2018).

Дослідження (Saito, K.; Spence, R, 2004) розглядали методи розподілу пошкоджених будівель завдяки візуальному окресленню за допомогою оптичних зображень після події для підтримки раннього планування надзвичайних ситуацій та порятунку. Проте даний підхід звичайно має ряд недоліків, оскільки ручне дешифрування може займати порівняно більше часу ніж автоматичне, таким чином втрачаються важливі для створення карт швидкого реагування години.

Напівавтоматичний підхід був застосований для ідентифікації пошкодження регіону Бам, Іран після землетрусу 2003 року. Використовуючи спектральну (колірну) і текстурну інформацію із зображення після землетрусу, за допомогою напівавтоматичного комп'ютерного алгоритму було ідентифіковано області пошкоджень (Rathje, E.M. and all, 2005).

Пошкодження будівлі було виявлено за допомогою сегментації вододілу аерофотознімків після події, припускаючи, що інформація про їхню форму доступна як збережений рівень географічної інформаційної системи (ГІС). Даний підхід базується на виявленні пошкоджених будівель внаслідок землетрусу за допомогою сегментації вододілу аерофотознімків, що датуються після події. Метод ґрунтується на зв'язку між будівлями та їхніми тінями. Затемнені тіні від будівель виявляються за допомогою сегментації вододілу на основі занурення (Turker, M.; Sumer, E., 2008).

Досить популярні піксельно-орієнтовані підходи, що застосовуються для VHR зображень дистанційного зондування, мають свої обмеження, такі як вилучення інформації та точність класифікація наземних об'єктів. У дослідженні (Li, X. and all., 2011) пропонується об'єктно-орієнтований метод

отримання інформації про пошкодження будівельної інфраструктури, спричинені землетрусом магнетудою 5.12, що стався у Веньчуані, Китайська Народна Республіка, за допомогою VHR зображень дистанційного зондування. Цей метод базується на сегментації зображень на об'єкти, що не перетинаються. Потім за допомогою методів класифікації виділяються пошкоджені та непошкоджені будівлі за допомогою спектральних характеристик, текстур, форм та контексту.

Для оцінки пошкоджень окремих будівель у міській зоні Порт-о-Пренса, Гаїті, внаслідок землетрусу 2010 року використовувалися VHR SAR знімки. Порівняння знімків до та після події та інвентаризація пошкоджених будівель показали, що інтенсивність розсіяння між зображеннями значно більше змінювалася в обвалених будівлях, ніж у менш пошкоджених. Була розроблена лінійна дискримінантна функція, яка ґрунтується на різниці та коефіцієнті кореляції зображень, для виявлення обвалених будівель. Результат показав, що майже 75% будівель були правильно виявлені за допомогою дискримінантного аналізу. Оцінка точності виявила складність виявлення малих і затінених будівель через недостатню кількість пікселів на зображеннях та затемнення будівельної інфраструктури іншими деталями (Miura, Hiroyuki et al, 2016).

Шляхом аналізу зображень VHR SAR можна виявити оригінальні контури будівель, опираючись на карту площ будівельної інфраструктури місцевості. Виявлені особливості контуру будівель на SAR-зображенні дозволяють ідентифікувати їх як зруйновані. У ході дослідження використовувалися три класифікатори машинного навчання: random forest (RF), support vector machine

(SVM) і K-nearest neighbor (K-NN). Результати показують, що запропонований метод досягає загальної точності понад 80% з використанням трьох класифікаторів (Wu, F. and all, 2016).

Дослідження, проведене (Zhai, W. and Huang, C, 2016), використовувало PolSAR методи ДЗЗ, які використовують радарні системи для отримання і обробки поляризованих радіохвильових сигналів. Знімки PolSAR, використовувались для виявлення пошкоджених житлових об'єктів і датуються після землетрусу в провінції Цінхай, Китай, у 2010 році. Однією з переваг радарного знімання є можливість зйомки вночі та незначний вплив погодних умов, таких як туман чи хмарність. У дослідженні був використаний метод кластеризації Wishart, і його максимальна точність на наданих авторами даних становила 79,2%.

У дослідженнях з використанням дистанційного зондування часто використовуються LiDAR знімки для отримання детальних даних про висоту, створення цифрових моделей рельєфу та формування точкових хмар високої роздільності поверхні Землі. Лазерні імпульси, випромінювані сенсором LiDAR, можуть проникати через рослинність та інші перешкоди, дозволяючи отримувати інформацію про поверхню Землі навіть в густолісних областях. Це робить LiDAR особливо корисним для картографування ландшафтів, вивчення топографії та дослідження забудованого середовища. Дане дослідження (Rastiveis, H. and all., 2015) зосереджене на автоматичному створенні карти пошкоджень після землетрусу з використанням даних LiDAR, що датуються після події та векторної карти будівельної інфраструктури, що датується до події землетрусу в Порт-о-Пренс, 2010-го року. Загальна точність 91,59%

доводить здатність класифікації SVM розрізняти пошкоджені та непошкоджені будівлі. Нарешті, 91 будівля в тестовій зоні була правильно позначена як пошкоджена/неушкоджена. Хоча результати були багатообіцяючими, необхідні подальші дослідження для покращення остаточного рішення (Rastiveis, H. and all., 2015).

Сучасні тенденції і досягнення у галузі машинного навчання спонукають дослідників звернути увагу на його застосування у оцінці пошкоджень будівельної інфраструктури внаслідок природних катастроф. Методи машинного навчання, зокрема глибокі нейронні мережі, показали перспективні результати в автоматичній класифікації, зокрема в розпізнаванні ознак пошкоджень. Ці дослідження мають на меті використання потужності алгоритмів машинного навчання для аналізу та інтерпретації різноманітних ознак пошкоджень, що дозволяє ефективно та точно виявляти пошкоджені ділянки (Theodoridis, S., Koutroumbas, K., 2009). Шляхом навчання нейронних мереж на великих наборах даних вони можуть вилучати складні зв'язки між вхідними ознаками, поліпшуючи загальну ефективність процесу класифікації. Цей новаторський підхід має великі перспективи для покращення швидкості та точності оцінки пошкоджень для поліпшення заходів миттєвого реагування на сейсмічні та інші природні катастрофи.

Більшість досліджень використовували методи SVM для створення класифікаторів пошкодженої будівельної інфраструктури. Виявлення пошкоджених будівель у міському середовищі на основі різночасових даних VHR-зображень за допомогою однокласового класифікатора One-Class Support Vector Machine (OCSVM) вимагає навчальної вибірки лише з пошкоджень

будівель. Вилучення ознак проводиться на рівні пікселів та об'єктів. Різні вхідні вектори для класифікатора OCSVM були протестовані з метою оцінки дискримінаційної здатності спектральних і просторових ознак: спектральних ознак на рівні пікселів і текстурних ознак, а також ознак на рівні об'єктів. Результати показали, що OCSVM показав кращі результати на рівні об'єктів із загальною точністю 82,33% (Peijun Li and all., 2010).

Досліджено багатоетапні методи виявлення зруйнованої інфраструктури з використанням двочасових (до і після землетрусу) VHR даних LiDAR. Наземні об'єкти, які не постраждали і значно відрізняються від зруйнованих будівель, такі як неушкоджені будівлі, тротуари, тіні та рослинність, спочатку були вилучені за допомогою LiDAR знімків, а потім замасковані. Зруйновані будівлі були виділені шляхом класифікації комбінованих до і після землетрусу знімків і текстурних зображень решти території за допомогою однокласового класифікатора (OCSVM) (Xue Wang, Peijun Li, 2015).

Дослідження (James B. and all, 2016) використали системний підхід для оцінки об'єктної сегментації зображень і алгоритмів машинного навчання для класифікації пошкоджень будівель на знімках дистанційного зондування. Дослідження мало на меті протестувати різні алгоритми і параметри моделей, а результати порівнювали з відібраними вручну тестовими прикладами, що представляють різні класи. Таким чином, оцінюючи ефективність сегментації та класифікації різних класів і порівняти різні рівні багатокрокової сегментації знімків ДЗЗ.

Якщо кількість повністю зруйнованих будівель набагато менша, ніж неушкоджених або менш постраждалих (наприклад, землетрус на Гаїті 2010 року), класифікатору складно вивчити вибірки меншості класів через проблему навчання з дисбалансом класів. У такому випадку згорткова нейронна мережа (CNN) використовується для ідентифікації зруйнованих будівель на супутникових знімках після події. Архітектура SqueezeNet добре показала себе в класифікації зруйнованих і не зруйнованих будівель, досягнувши середнього показника загальної точності (OA) 78,6% для двох тестових регіонів (Ji M. and all, 2018). У цьому дослідженні було використано гібридний підхід для збалансування класів, що включав методи збільшення даних через аугментацію та внесення поправочних коефіцієнтів до функції втрат.

Convolution Neural Networks (CNN) або згорткові нейронні мережі (ЗНМ) - це тип нейронної мережі з глибоким навчанням, спеціально розроблений для аналізу візуальних даних, таких як зображення. В контексті оцінки пошкоджень будівель, ЗНМ можна навчати на маркованих наборах даних зображень, які зображують як пошкоджені, так і непошкоджені будівлі. Під час навчання мережа вчиться розпізнавати візуальні патерни і особливості, пов'язані з різними типами і ступенями пошкоджень, тому ЗНМ широко використовуються в оцінці пошкоджень будівель. CNN продемонстрували високу продуктивність у різних завданнях обробки зображень, в тому числі в оцінці пошкоджень будівель. Вони можуть ефективно фіксувати локальні та глобальні особливості, дозволяючи точно ідентифікувати та класифікувати пошкоджені ділянки (Zheng, Z. and all, 2021).

CNN здатні ефективно обробляти великі обсяги даних зображень, що робить їх придатними для застосування в режимі реального або близького до реального часу при ліквідації наслідків стихійних лих і відновлювальних роботах.

Об'єктно-орієнтовані методи розпізнавання будівельної інфраструктури на зображеннях дистанційного зондування з високою роздільною здатністю мають ряд проблем, таких як незадовільна сегментація об'єктів і складний вибір ознак, що ускладнює швидку оцінку пошкоджень, у випадках розташованих поруч груп будівель. Вдосконалена архітектура згорткової нейронної мережі (CNN) Inception V3 поєднує зображення дистанційного зондування та блокові векторні дані для оцінки ступеня пошкодження груп будівель на зображеннях дистанційного зондування після землетрусу. Використання CNN, дозволяє обирати найкращі ознаки автоматично, вирішуючи проблему складного вибору ознак. Крім того, межі блоків можуть формувати значущі межі для груп будівель, що може ефективно замінити сегментацію зображень і уникнути її фрагментарних і незадовільних результатів. Додавши окремі та комбіновані шари, метод покращує мережу. Данна методика була використана при класифікації пошкоджених груп будівель на аерофотознімках з роздільною здатністю 0,5 м після землетрусу Юшу. Точність тесту склала 90,07% (Ma, H. and all, 2020).

Нещодавня серія експериментів, що де були використані дані з 13 різних природних катастроф, які відрізнялися за типом небезпеки, географічним розташуванням та параметрами зображень як тестовий набір. Та навчальний датасет з 175 289 будівель з набору даних, який містить анотовані людиною

різнокласові мітки пошкоджень на супутникових знімках високої просторової розрізненості RGB діапазонів. Результат показав, що модель все одно досягла високого рівня продуктивності без використання будь-яких мічених даних тестової катастрофи під час навчання. Данна модель може успішно застосовуватись в оперативних умовах, коли інформація про пошкодження інфраструктури внаслідок катастрофи не доступна у реальному часі, а отже, повторне навчання моделі не є можливим (Valentijn, T. and all, 2022).

Дослідження багатьох науковців у цій області дають підстави стверджувати, що більшість з них зосереджена на розв'язанні задачі класифікації житлової інфраструктури на пошкоджені та непошкоджені об'єкти, використовуючи як основу вектор попередньо оконтурених будівель. Проте, першочергова задача виділення контурів будівель не менш важлива, оскільки ця інформація потрібна для створення карт пошкоджень та планування реконструктивних робіт.

Дослідження (Iglouikov V. and all, 2017) акцентують свою увагу на сегментації сегментації мультиспектральних супутникових знімків високої просторової розрізненості на 10 різних класів: будівлі, магістралі, дороги, стежки, дерева, трава, річки, стояча вода, великі та малі машини, а також інші створені людиною споруди. Для цього була використана згортова нейронна мережа UNet, яка складається з енкодера та декодера. Загальна точність (Intersection over Union (IoU)) виділення контурів будівель склала 75%, а для деяких інших простіших класів точність сягала до 96%, що підтверджує ефективність використання архітектури UNet для семантичної сегментації супутникових знімків.

## 1.4 Популярні методи машинного навчання

Машинне навчання передбачає розробку і застосування алгоритмів моделювання що дозволяють комп'ютерам або машинам «навчатися», створюючи прогнози або приймати рішення без явного програмування. Машинне навчання включає аналіз даних, виявлення закономірностей або взаємозв'язків і розробку моделей, які можуть узагальнювати дані. Таким чином, модель може давати валідні результати на вибірках даних, яких вона до цього не бачила.

Машинне навчання використовує два основні підходи: навчання з учителем, коли модель навчається на основі маркованих пар вхід-вихід для прогнозування нових даних, і навчання без учителя, яке фокусується на виявленні прихованих шаблонів або структур у вхідних даних без явних вихідних міток.

Поширені алгоритми машинного навчання з учителем включають методи опорних векторів (SVM), дерева рішень, класифікатори на основі k-найближчих сусідів, наївний Байєсівський класифікатор, логістичну регресію та нейронні мережі. У даній роботі буде розглянуто алгоритми машинного навчання з учителем, що вже показали значні результати у завданнях сегментації та класифікації при роботі з даними ДЗЗ (Peijun Li and all., 2010; Wu, F. and all, 2016; Valentijn, T. and all, 2022).

На сьогоднішній день, згорткові нейронні мережі стали передовими алгоритмами, що практично повністю замінили стандартні методи машинного навчання. Основною недоліком стандартних методів класифікації є необхідність ручного вибору ознак. У випадку класифікації зображень, такими

ознаками можуть бути абсолютна яскравість або попередньо розраховані текстурні характеристики.

Згорткові нейронні мережі відрізняються від класичних методів машинного навчання тим, що вони автоматично вилучають ознаки у процесі навчання. Зображення подається на вхід нейронної мережі, а згорткова частина мережі виконує вилучення ознак, які потім передаються до класифікаційної частини моделі. Ці ознаки мають різний рівень детальності, починаючи від найпростіших елементарних одиниць зображення, таких як лінії або дуги, і закінчуючи складними ознаками, наприклад будівлею, дорогою, деревом тощо. Кількість рівнів ознак в нейронній мережі у теорії не має обмежень, так само як і кількість виділених ознак на кожному рівні. Широкий діапазон автоматично виділених ознак дозволяє згортковим нейронним мережам враховувати різноманітні фактори при прийнятті рішень.

Цей розділ призначений для порівняння стандартних алгоритмів класифікації згорткових нейронних мереж у контексті розпізнавання образів на зображеннях. В ньому будуть надані короткі описи класичних алгоритмів, таких як метод опорних векторів, дерева рішень, Random Forest та Gradient boosting, а також будуть розглянуті їх основні переваги та недоліки.

#### **1.4.1 Метод опорних векторів**

Сучасне ядерне формулювання методу опорних векторів було розроблено Володимиром Вапником та Корінною Кортес на початку 1990-х років. Ядерні методи охоплюють набір алгоритмів класифікації, найвідомішим з яких є машина опорних векторів (SVM) (Vapnik V. and Cortes C.,1995).

Мета SVM полягає у вирішенні проблем класифікації шляхом визначення оптимальних меж рішення між двома наборами точок, що належать до різних категорій. Ці межі розділяють навчальні дані на окремі простори для кожної категорії. Для класифікації нових точок даних враховується їхнє положення відносно границі рішення.

SVM визначають ці межі прийняття рішень за допомогою двох кроків:

- дані відображаються в нове високорозмірне представлення, де межа рішення може бути виражена у вигляді гіперплощини (у випадку двовимірних даних гіперплощина відповідає прямій лінії).
- відповідна межа рішення, відома як гіперплощина поділу, обчислюється шляхом максимізації відстані між гіперплощиною і найближчими точками даних з кожного класу. Цей крок, який називається максимізацією запасу, гарантує, що границя добре узагальнює нові вибірки за межами навчального набору даних.

Ця властивість дозволяє ефективно застосовувати границю рішення до нових точок даних, які не були частиною навчального набору даних. Проте, виявилось, що метод опорних векторів (SVM) має складності у масштабуванні до великих наборів даних і не показує задовільних результатів в задачах сприйняття, зокрема в класифікації зображень. Оскільки SVM є поверхневим методом, його застосування до задач сприйняття вимагає вручну вибирати інформативні ознаки (François Chollet, 2018).

### 1.4.2 Древа рішень

Древа рішень - це графічні структури, які нагадують блок-схеми і використовуються для класифікації вхідних даних або прогнозування вихідних значень на основі заданих вхідних даних. Вони забезпечують чітке та інтуїтивно зрозуміле представлення процесів прийняття рішень. У 2000-х роках древа рішень привернули значну увагу дослідників і до 2010 року поступово перевершили за популярністю ядерні методи. Цей зсув відбувся завдяки простоті візуалізації та інтерпретації дерев рішень, що зробило їх кращим вибором для багатьох застосувань (François Chollet, 2018).

Вони будуються у деревоподібному форматі, де кожен внутрішній вузол представляє ознаку або атрибут, кожна гілка представляє можливе значення або результат цього атрибуту, а кожен вузол листя представляє рішення або прогноз.

Процес побудови дерева рішень передбачає рекурсивне розбиття даних на основі різних атрибутів з метою максимізації однорідності або чистоти отриманих підмножин при кожному розбитті. Критерії розбиття можуть відрізнятися залежно від використовуваного алгоритму, але найчастіше використовуються такі показники, як ентропія, домішка Джині або інформаційний виграш.

Після того, як дерево рішень побудовано, його можна використовувати для класифікації нових екземплярів або прогнозування результатів, слідуючи шляхом від кореневого вузла до листового вузла на основі значень вхідних ознак. Кожен листовий вузол відповідає певному класу або прогнозу.

Дерева рішень мають ряд переваг, таких як їх інтерпретованість, оскільки структура дерева дозволяє чітко візуалізувати і зрозуміти процес прийняття рішень. Вони можуть обробляти як категоріальні, так і числові дані, а також є стійкими до викидів і пропущених значень. Дерева рішень також можна використовувати для задач регресії, прогнозуючи безперервні значення у вузлах листків.

Однак дерева рішень можуть бути схильні до перенавчання, особливо коли дерево стає занадто складним або коли навчальні дані є зашумленими. Щоб зменшити надмірну пристосованість, можна застосувати такі методи, як обрізання, встановлення максимальної глибини або використання ансамблевих методів, таких як випадкові ліси або градієнтне підсилення.

Загалом, дерева рішень є універсальними і широко використовуваними моделями машинного навчання, які пропонують простий, але потужний підхід до завдань прийняття рішень і прогнозування (François Chollet, 2018).

### **1.4.3 Random Forest**

Алгоритм випадкових лісів представив надійний практичний підхід до навчання на основі дерев рішень, який передбачає побудову великої кількості спеціалізованих дерев рішень, а потім об'єднання їхніх результатів. Випадкові ліси застосовуються до широкого кола проблем - можна сказати, що вони майже завжди є другим найкращим алгоритмом для будь-якої неглибокої задачі машинного навчання (François Chollet, 2018).

Random Forest це метод навчання ансамблю, він базується на концепції беггінгу (агрегування бутстреп-вибірок) та вводить додатковий рівень випадковості в алгоритм дерев прийняття рішень.

У Random Forest кілька дерев прийняття рішень навчаються на різних підмножинах навчальних даних, випадково вибраних з повторенням (бутстрепінг). Кожне дерево навчається незалежно і робить передбачення на основі ознак вхідних даних. Загальне передбачення Random Forest визначається шляхом агрегації передбачень всіх окремих дерев, за допомогою більшості голосів для завдань класифікації або усереднення для завдань регресії. Головна ідея Random Forest полягає в тому, що шляхом поєднання кількох дерев модель ансамблю може зменшити перенавчання і покращити загальну ефективність.

#### **1.4.4 Gradient boosting**

Gradient boosting - це метод машинного навчання, який використовується для побудови потужних прогностичних моделей. Він базується на ідеї поетапного навчання слабких моделей (наприклад, дерев рішень) і поступового покращення їх шляхом зведення помилок моделей до мінімуму (François Chollet, 2018).

Процес gradient boosting починається з побудови першої слабкої моделі, яка навчається на початкових даних. Потім обчислюється помилка між прогнозами першої моделі і справжніми значеннями цільової змінної. Наступним кроком є побудова другої моделі, яка намагається скоригувати

помилки першої моделі. Цей процес продовжується, додаючи нові моделі і коригуючи помилки попередніх моделей.

Ключовою ідеєю gradient boosting є використання градієнта функції втрати для покращення моделей. Градієнт вказує напрямок найшвидшого зменшення помилки моделі, тому нові моделі будуються таким чином, щоб коригувати помилки в напрямку, протилежному градієнту.

Gradient boosting є дуже потужним алгоритмом і здатний досягати високої точності в прогнозуванні. Він використовується в багатьох областях, включаючи рекомендації, ранжування, класифікацію та регресію. Проте, через свою складність, даний алгоритм може вимагати більшої обчислювальної потужності та більшої кількості даних для ефективної роботи особливо у випадках розпізнавання.

## **1.5 Основні відомості про нейронні мережі**

### **1.5.1 Фактори популярності нейронних мереж**

Популярність нейронних мереж обумовлена не лише кращою ефективністю у більшості завдань, одним з головних факторів є те, що глибоке навчання також суттєво спрощує процес конструювання ознак, повністю автоматизуючи його. Даний процес є ключовим етапом машинного навчання, який раніше виконувався вручну.

Попередні методи машинного навчання, тобто поверхневе навчання, передбачали лише перетворення вхідних даних в один або два послідовних простори представлення, зазвичай за допомогою простих перетворень, таких як методи опорних векторів (SVM) або дерева рішень. Однак для розв'язання

складних завдань ці методи зазвичай не забезпечували достатньо точних результатів. Тому дослідникам доводилося прикладати значні зусилля для приведення вихідних даних до форми, більш придатної для обробки такими методами, шляхом ручного покращення представлення даних. Цей процес називається конструюванням ознак.

Глибоке навчання, навпаки, повністю автоматизує цей крок: із застосуванням методів глибокого навчання всі ознаки витягуються за один прохід, без необхідності конструювати їх вручну. Це дуже спростило процес машинного навчання, тому що часто складний і багатоступінчастий конвеєр виявилось можливим замінити єдиною простий наскрізний моделлю глибокого навчання.

Починаючи з 2012 року, глибокі згорткові нейронні мережі (convnets) стали передовими алгоритмами для всіх завдань розпізнавання образів, а також успішно застосовуються у багатьох інших завданнях розпізнавання. Вони повністю замінили метод опорних векторів і дерев рішень у подібному спектрі завдань (François Chollet, 2018).

### **1.5.2 Основні елементи архітектури нейронної мережі**

Нейронні мережі - це моделі обчислювальної системи, які нагадують структуру та функціонування людського мозку. Вони складаються зі з'єднаних штучних нейронів, які працюють разом для вирішення завдань обробки інформації, таких як розпізнавання образів, класифікація даних, передбачення та управління (François Chollet, 2018).

Ключовим елементом нейронних мереж є штучні нейрони, які функціонують подібно до нейронів мозку. Кожен штучний нейрон отримує вхідні сигнали, виконує обчислення з використанням вагових коефіцієнтів і активаційної функції, і видає вихідний сигнал. З'єднання між нейронами мають вагові коефіцієнти, які визначають силу впливу одного нейрона на інший.

Кожен з нейронів характеризується своєю вагою, зсувом та функцією активації. Нейронні мережі мають структуру, яка складається з вхідного шару, прихованих шарів (якщо вони присутні) і вихідного шару. Вхідний шар отримує дані від зовнішнього середовища, приховані шари обробляють ці дані шляхом здійснення обчислень, а вихідний шар генерує відповідь або результат.

Вхідний шар приймає необроблені вхідні дані з області. На цьому рівні не виконується жодних обчислень. Вузли тут просто передають інформацію (ознаки) до прихованого шару.

Прихований шар виконує всі види обчислень над ознаками, введеними через вхідний шар, і передає результат вихідному шару. Як впливає з назви, вузли цього шару не є відкритими. Вони є абстракцією для нейронної мережі.

Це кінцевий шар мережі, який вносить інформацію, отриману через прихований шар, і видає кінцеве значення в результаті.

Нейронні мережі навчаються за допомогою алгоритму навчання, такого як зворотне поширення помилки. В процесі навчання вагові коефіцієнти нейронів оновлюються для досягнення бажаного вихідного результату. Цей процес триває через ітерації, поки мережа не досягне достатньої точності (Nwankpa, C. and all, 2021).

Всі приховані шари зазвичай використовують однакову функцію активації. Однак вихідний шар зазвичай використовує іншу функцію активації, ніж приховані шари. Вибір залежить від мети або типу прогнозу, зробленого моделлю.

### **1.5.3 Алгоритм оберненого розповсюдження помилки**

Метод зворотного поширення помилки (англ. backpropagation) — метод навчання багат шарового перцептронну. Це ітеративний градієнтний алгоритм, який використовується з метою мінімізації помилки роботи багат шарового перцептронну та отримання бажаного виходу. Основна ідея цього методу полягає в поширенні сигналів помилки від виходів мережі до її входів, в напрямку, зворотному прямому поширенню сигналів у звичайному режимі роботи (Nwankra, C. and all, 2021).

Пряме поширення (англ. Feedforward Propagation) - потік інформації відбувається в прямому напрямку. Вхід використовується для обчислення деякої проміжної функції в прихованому шарі, яка потім використовується для обчислення виходу.

Метод зворотного поширення помилки застосовується для повторного коригування вагів зв'язків мережі, щоб мінімізувати різницю між фактичним вихідним вектором мережі та бажаним вихідним вектором. Зворотне поширення має на меті мінімізувати функцію витрат шляхом коригування ваг та зміщень мережі. Градієнти функції вартості визначають рівень налаштування таких параметрів, як функція активації, ваги, зміщення тощо. У прямому поширенні активаційна функція є математичними "воротами" між

входом, що живить поточний нейрон, і його виходом, що йде до наступного шару.

Алгоритм зворотного розповсюдження шукає мінімум функції помилки у просторі вагових коефіцієнтів шляхом використання методу градієнтного спуску. Розв'язок навчання вважається комбінацією вагових коефіцієнтів, яка мінімізує функцію помилки. Однак, для використання цього методу, функція помилки повинна бути неперервною та диференційованою, оскільки обчислення градієнта функції помилки потрібне на кожному кроці ітерації (Nwankra, C. and all, 2021).

Метод оберненого розповсюдження помилки полягає в обчисленні градієнту втрат щодо кожного вагового коефіцієнта в мережі. Після цього коефіцієнти зміщуються в бік, протилежний градієнту для зменшення значення функції втрат. Градієнт - це похідна операції з тензором, узагальнення поняття похідної на функції з багатовимірними вхідними даними, тобто на функції, які беруть на вході тензори.

Мета функції активації - додати нелінійності нейронній мережі. Функція активації вводить додатковий крок на кожному шарі під час прямого поширення, але її обчислення того варте. Якщо є нейронна мережа, працює без активаційних функцій кожен нейрон буде виконувати лише лінійне перетворення вхідних даних за допомогою ваг та зсувів. Це тому, що не має значення, скільки прихованих шарів ми додамо до нейромережі; всі шари будуть поводитися однаково, тому що композиція двох лінійних функцій сама є лінійною функцією. Хоча нейронна мережа стає простішою, навчити будь-яку складну задачу неможливо, тому модель буде просто лінійною регресійною

моделлю.

Функцій активації нейронних мереж:

1. Бінарна ступінчаста функція  $f(x)$ . Залежить від порогового значення, яке вирішує, чи слід активувати нейрон, чи ні (1.1).

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases} \quad (1.1)$$

де  $x$  - аргумент функції активації.

Вхід, що подається на функцію активації, порівнюється з певним порогом; якщо вхід більший за нього, то нейрон активується, інакше він деактивується, тобто його вихід не передається на наступний прихований шар (рисунок. 1.4, а).

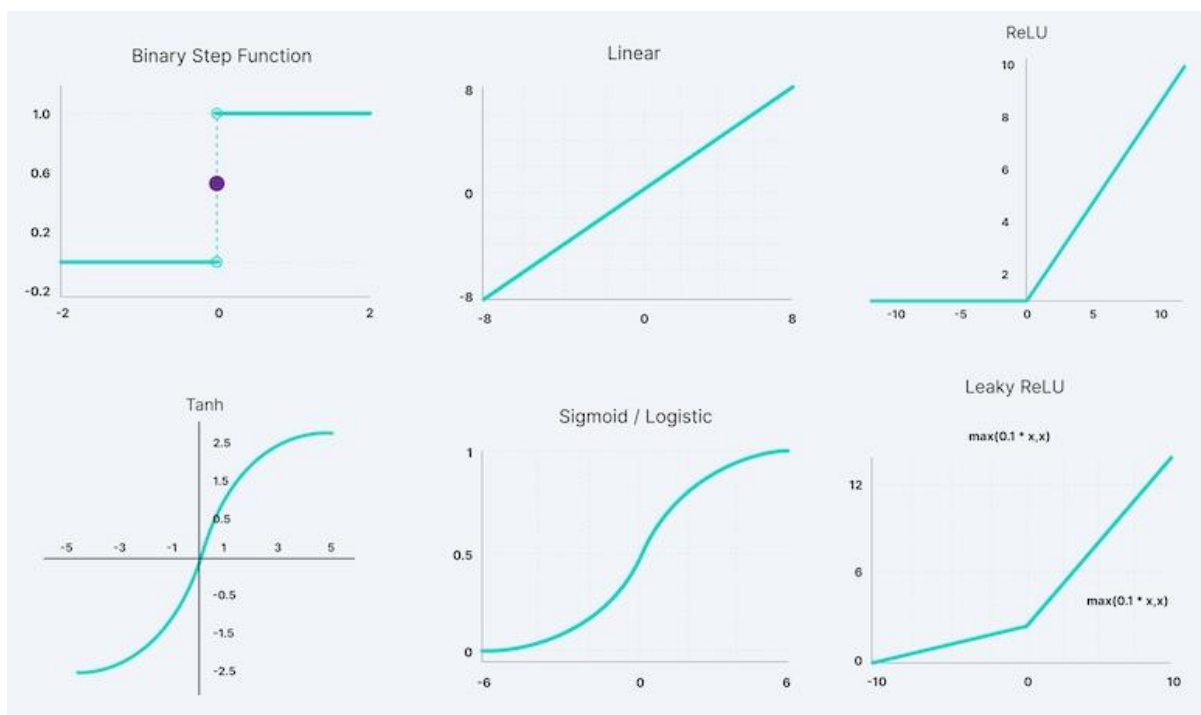


Рис. 1.4 - Функції активації нейронної мережі: а) Бінарна ступінчаста функція, б) Лінійна функція активації, в) Функція ReLU, г) Функція Тана, г) Сигмоїдна функція активації, д) Функція Leaky ReLU (Nwankpa, C. and all, 2021)

2. Лінійна функція активації, також відома як "без активації" або "тотожна функція" (помножена на  $x \cdot 1.0$ ) - це функція, де активація пропорційна входу (1.2)

$$f(x) = x \quad (1.2)$$

де  $x$  - аргумент функції активації.

Лінійна функція активації має дві основні проблеми : неможливо використовувати зворотне поширення, оскільки похідна функції є константою і не має відношення до входу  $x$ ; всі шари нейронної мережі згортаються в один, якщо використовується лінійна функція активації. Незалежно від кількості шарів у нейромережі, останній шар все одно буде лінійною функцією першого шару. Отже, по суті, лінійна функція активації перетворює нейронну мережу лише на один шар (рисунок. 1.4, б) (Nwankpa, C. and all, 2021)

3. Нелінійні активаційні функції активації дозволяють зворотне поширення, оскільки тепер похідна функція буде пов'язана з входом, і можна повернутися назад і зрозуміти, які ваги у вхідних нейронах можуть забезпечити краще передбачення. Також, нелінійні активаційні функції активації дозволяють накладати кілька шарів нейронів, оскільки вихід тепер буде нелінійною комбінацією вхідних даних, пропущених через кілька шарів. Будь-який вихід може бути представлений як функціональне обчислення в нейронній мережі. Лінійна функція активації, показана вище, є просто лінійною регресійною моделлю. Через свою обмежену потужність, вона не дозволяє моделі створювати складні відображення між входами та виходами мережі (Shiv Ram Dubey and all, 2022).

Розглянемо деякі найпопулярніші нелінійні функції активації нейронних мереж та їхні характеристики.

### 1. Функція Тана (гіперболічний тангенс)

Функція Тана дуже схожа на сигмоїдну/логістичну функцію активації, і навіть має таку саму S-подібну форму з різницею в діапазоні вихідних значень від -1 до 1. У функції Тана, чим більший вхід (більш позитивний), тим ближче вихідне значення буде до 1.0, тоді як чим менший вхід (більш негативний), тим ближче вихідне значення буде до -1.0 (1.3).

$$f(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (1.3)$$

де  $x$  - аргумент функції активації.

Вихід функції активації  $\tanh$  має нульовий центр; отже, ми можемо легко відобразити вихідні значення як сильно від'ємні, нейтральні або сильно додатні. Зазвичай функція активації  $\tanh$  використовується в прихованих шарах нейронної мережі, оскільки її значення лежать в діапазоні від -1 до 1; отже, середнє значення для прихованого шару виходить рівним 0 або дуже близьким до нього. Це допомагає відцентрувати дані і значно полегшує навчання наступного шару (рис. 1.4, г) (Nwankra, C. and all, 2021).

### 2. Сигмоїдна / логістична функція активації

Ця функція приймає будь-яке дійсне значення на вхід і виводить значення в діапазоні від 0 до 1. Чим більше вхідне значення (більш позитивне), тим ближче вихідне значення буде до 1.0, тоді як чим менше вхідне значення (більш негативне), тим ближче вихідне значення буде до 0.0, як показано нижче (1.4)

$$f(x) = \frac{1}{1+e^{-x}} \quad (1.4)$$

де  $x$  - аргумент функції активації.

Сигмоїдна/логістична функції активації зазвичай використовується для

моделей, де на меті є передбачити ймовірність як вихідний результат. Оскільки ймовірність чого-небудь існує лише в діапазоні від 0 до 1, сигмоїдна функція є правильним вибором завдяки своєму діапазону.

Переваги сигмоїдної/логістичної функції активації є диференційованість, що забезпечує плавний градієнт, тобто запобігає стрибкам у вихідних значеннях. Це представлено S-подібною формою сигмоїдної функції активації (рис. 1.4, г) (Nwankra, C. and all, 2021).

### 3. Функція ReLU

ReLU розшифровується як випрямлена лінійна одиниця. Хоча вона справляє враження лінійної функції, ReLU має похідну функцію і дозволяє зворотне розповсюдження, одночасно роблячи її обчислювально ефективною. Основна проблема полягає в тому, що функція ReLU не активує всі нейрони одночасно. Нейрони будуть деактивовані лише тоді, коли результат лінійного перетворення буде меншим за 0 (1.5).

$$f(x) = \max(0, x) \quad (1.5)$$

де  $x$  - аргумент функції активації.

Оскільки активується лише певна кількість нейронів, функція ReLU є набагато ефективнішою з точки зору обчислень у порівнянні з сигмоїдною та тангенсною функціями. Також перевагою ReLU є те, що вона прискорює збіжність градієнтного спуску до глобального мінімуму функції втрат завдяки своїй лінійній, ненасичуваній властивості

Недоліком ReLU понад всіма її перевагами є те що, негативна сторона графіка робить значення градієнта нульовим (рисунок 1.4, в) (Nwankra, C. and

all, 2021). З цієї причини під час процесу зворотного поширення ваги та зсуви для деяких нейронів не оновлюються. Це може призвести до появи мертвих нейронів, які ніколи не активуються. Всі від'ємні вхідні значення негайно стають нульовими, що знижує здатність моделі підлаштовуватися або навчатися на даних належним чином (Shiv Ram Dubey and all, 2022).

#### 4. Функція Leaky ReLU

Leaky ReLU - це покращена версія функції ReLU для вирішення проблеми вмираючого ReLU, оскільки вона має невеликий позитивний нахил у від'ємній області. Переваги Leaky ReLU такі ж, як і у ReLU, на додаток до того, що вона дозволяє зворотне розповсюдження навіть для від'ємних вхідних значень (1.6)

$$f(x) = \max(0.1x, x) \quad (1.6)$$

де  $x$  - аргумент функції активації.

Зробивши цю невелику модифікацію для від'ємних вхідних значень, градієнт лівої частини графіка стає ненульовим значенням. Отже, ми більше не зустрічатимемо мертвих нейронів у цій області (рис. 1.4, д) (Nwankpa, C. and all, 2021).

## 2. ТЕХНОЛОГІЧНІ РІШЕННЯ

### 2.1 Застосування згорткових нейронних мереж

Використання методів глибокого навчання для автоматичного розпізнавання пошкоджених будівель базується на методах сегментації та класифікації даних ДЗЗ. Однак, ці завдання зазвичай стикаються з великими обсягами даних та складністю обробки, оскільки космознімки та дані ДЗЗ надходять у великій кількості та мають високу роздільну здатність.

У цьому контексті важливу роль відіграють згорткові нейронні мережі (CNN), які забезпечують потужні інструменти для автоматичної обробки та аналізу великого обсягу аеро та космознімків. CNN є одним з найефективніших підходів до розпізнавання образів, оскільки вони можуть виявляти складні залежності та особливості у вхідних даних. CNNs були застосовані до зображень дистанційного зондування для виявлення особливостей, виділення країв і проведення класифікації на основі пікселів (Kattenborn et al., 2021).

Головна перевага CNN полягає в їхній здатності автоматично виявляти та використовувати локальні структури та шаблони в зображеннях. Завдяки своїй архітектурі, CNN здатні впевнено вирішувати завдання сегментації, що передбачає визначення та виділення окремих об'єктів або регіонів із зображень. Вони також успішно застосовуються для класифікації, де вони можуть автоматично визначати категорію або клас, до якого належить кожне зображення (Li, Y. and all, 2016).

CNN забезпечують високу точність та швидкість обробки завдяки спеціальній архітектурі, яка враховує просторову структуру зображень. Вони використовують шари згортки для виявлення локальних ознак та шаблонів у зображеннях, а потім шари пулінгу для зменшення розмірності та виділення найважливіших ознак. Це дозволяє зменшити кількість параметрів та обчислень, зберігаючи при цьому важливу інформацію.

Застосування CNN у методах сегментації та класифікації космознімків та даних ДЗЗ принесло значні переваги. Вони дозволяють автоматизувати процес обробки даних, знижуючи залежність від ручної пре- та постобробки даних. Крім того, CNN здатні ефективно працювати з великими обсягами даних, що є характерним для космічних зображень та даних ДЗЗ (Li, Y. and all, 2016).

## **2.2 Огляд основних шарів згорткової нейронної мережі**

Архітектура ЗНМ складається з складається з вхідного шару, прихованого шару (шарів) і вихідного шару. Кожен шар містить різну кількість вузлів, а синапси з'єднують вузли одного шару з вузлами сусідніх шарів.

Шари в нейронних мережах є фундаментальною складовою частиною цих моделей. Кожен шар є модулем обробки даних, який виконує певну операцію над вхідними даними і генерує вихідні дані. Шари приймають на вхід тензори (многовимірні масиви) і виконують обчислення за допомогою внутрішніх параметрів, таких як вагові коефіцієнти (Kattenborn et al., 2021).

У нейронних мережах можуть використовуватися різні типи шарів, які забезпечують різні функціональності і типи обробки даних. Деякі з найпоширеніших типів шарів включають:

- Повнозв'язані (щільні) шари: Кожен нейрон у шарі пов'язаний з кожним нейроном попереднього шару. Вони використовуються для обробки векторних даних.
- Згорткові шари: Вони виконують згортку над вхідними даними, дозволяючи виявляти просторові шаблони. Зазвичай використовуються для обробки зображень.
- Пулінгові шари: Вони зменшують розмір вхідних даних, зберігаючи найважливішу інформацію.
- Рекурентні шари: Вони використовуються для моделювання послідовних даних і мають зв'язки між нейронами, які дозволяють запам'ятовувати попередні стани.

Комбінація різних типів шарів утворює архітектуру нейронної мережі, яка може вирішувати різноманітні завдання, включаючи класифікацію, регресію, сегментацію та генерацію образів.

### **2.3 Повнозв'язний шар нейронної мережі**

Повнозв'язний шар нейронної мережі - це нейронна мережа, яка має глибокий зв'язок, тобто кожен нейрон у щільному шарі отримує вхідні дані від усіх нейронів попереднього шару. Повнозв'язний шар нейронної мережі виконує матрично-векторне множення, а значення, що використовуються в матриці, є параметрами, які можна навчати і оновлювати за допомогою зворотного поширення.

Результатом роботи щільного шару є вектор  $n$ -вимірності. Щільний шар використовується для зміни розмірів, обертання, масштабування та перекладу

вектора. На рисунку 2.1 схематично показано три повнозв'язних шари, де вихід кожного з нейронів - є входом для нейронів наступного шару.

Основна перевага повнозв'язних шарів полягає в їх "універсальності". Вони можуть працювати з різноманітними типами вхідних даних, такими як вектори ознак, зображення або відео, без потреби в особливих припущеннях.

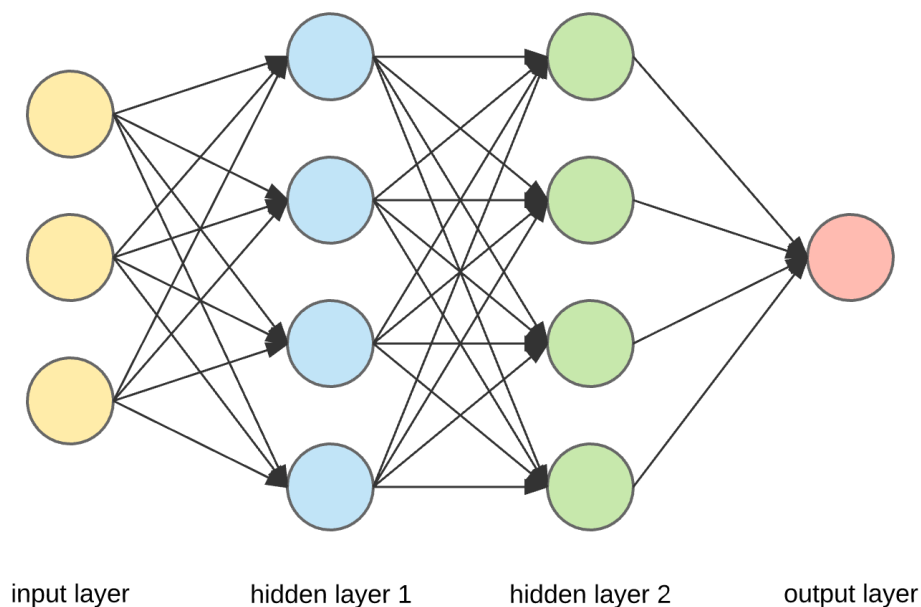


Рис. 2.1 — Повнозв'язні шари нейронної мережі з двома прихованими шарами (Convolutional Neural Networks (CNNs/ConvNets), 2022)

У поєднанні зі згортковими шарами, повнозв'язні шари широко використовуються для класифікації вектора ознак, який є вихідним результатом згорткових шарів нейронної мережі. Це дозволяє моделі використовувати проміжний репрезентативний вектор ознак, отриманий згортковими шарами, і зв'язувати його з різними класами або категоріями в рамках повнозв'язного шару для здійснення класифікації (Li, Y. and all, 2016).

## 2.4 Convolution Layer

Рівень Conv - це основний будівельний блок згорткової мережі, який виконує більшу частину обчислювальної роботи. Згортковий шар працює шляхом використання набору фільтрів або ядер, які скользять по вхідному зображенні або вектору ознак. Кожен фільтр виконує операцію згортки, яка застосовується до невеликих локальних областей даних, названих рецептивними полями. Це дозволяє виявити локальні особливості, такі як краї, форми або текстури, на різних рівнях деталей.

Після застосування згортки отримується карта ознак, яка представляє просторову ієрархію важливих ознак у вхідних даних. Зазвичай, згорткові шари використовуються разом зі стрибками (pooling) і шарами активації (activation layers), що допомагають підвищити інваріантність до зрушень та зменшити кількість параметрів (François Chollet, 2018).

Параметри шару CONV складаються з набору фільтрів, що навчаються. Кожен фільтр є невеликим у просторі (по ширині та висоті), але поширюється на всю глибину вхідного об'єму. Наприклад, типовий фільтр на першому шарі ConvNet може мати розмір 5x5x3 (тобто 5 пікселів завширшки і заввишки, і 3 - тому що зображення має глибину 3, колірні канали). Під час прямого проходу ми ковзаємо (точніше, зортаємо) кожен фільтр по ширині і висоті вхідного об'єму і обчислюємо добутки точок між входами фільтра і вхідним об'ємом у будь-якій позиції. Пересуваючи фільтр по ширині і висоті вхідного об'єму, ми створюємо двовимірну карту активації, яка показує реакції цього фільтра в кожній просторовій позиції (рис.2.2). Інтуїтивно мережа запам'ятає фільтри, які активуються, коли вона бачить якусь візуальну особливість, наприклад, край

певної орієнтації або пляму певного кольору на першому шарі, або, зрештою, цілі стільники або колесоподібні патерни на вищих шарах мережі. Тепер ми матимемо цілий набір фільтрів у кожному шарі CONV (наприклад, 12 фільтрів), і кожен з них створюватиме окрему двовимірну карту активації. Ми складемо ці карти активації вздовж виміру глибини і отримаємо вихідний об'єм (Convolutional Neural Networks (CNNs/ConvNets), 2022).

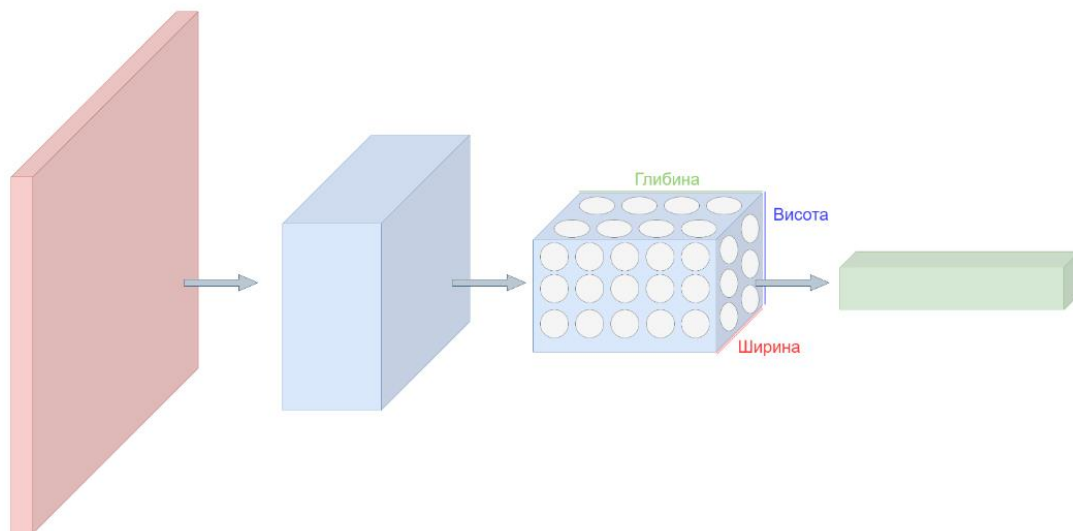


Рис.2.2 Принцип роботи згорткового шару (Convolutional Neural Networks (CNNs/ConvNets), 2022)

Згорткові шари володіють декількома ключовими характеристиками, які роблять їх особливими для обробки зображень. По-перше, шаблони, які вони вивчають, є інваріантними до зсуву. Це означає, що після того, як конкретний шаблон вивчено в одній частині зображення, він може бути розпізнаний в будь-якій іншій частині зображення за допомогою цього ж шаблону. У порівнянні з повнозв'язними шарами, яким потрібно вивчати шаблони заново, якщо вони з'являються в іншому місці, це збільшує ефективність згорткових нейронних мереж у завданнях обробки зображень. Це означає, що менше

навчальних прикладів потрібно для вивчення кожного з шаблонів.

По-друге, згорткові шари можуть вивчати просторові ієрархії шаблонів. Перший згортковий шар вивчає локальні шаблони, такі як краї, другий шар вивчає більш великі шаблони, які складаються з вихідних шаблонів першого шару, і так далі. Це дозволяє згортковим нейронним мережам вивчати все більш абстрактні візуальні представлення. Завдяки просторово-ієрархічній природі видимого світу, згорткові шари можуть підвищити рівень абстракції від простих до складніших візуальних концепцій (Kattenborn et al., 2021).

## 2.5 Pooling Layer

Pooling Layer, або шар пулінгу, є елементом нейронної мережі, який дозволяє значно зменшити просторовий розмір зображення. Ідея пулінгу полягає в тому, що якщо попередній згортковий шар вже виявив деякі важливі ознаки, то для подальшої обробки не потрібно так детально враховувати всю інформацію. Зображення ущільнюється до менш детального вигляду. Крім того, виключення непотрібних деталей сприяє запобіганню перенаванчання нейронної мережі. Зазвичай шар пулінгу вставляється після згорткового шару перед наступним згортковим шаром.

Pooling Layer або шар субдискретизації являє собою нелінійне ущільнення карти ознак, при цьому група пікселів (зазвичай розміру  $2 \times 2$ ) ущільнюється до одного пікселя, проходячи нелінійне перетворення. Найбільш розповсюджена при цьому функція максимуму (Max Pooling) (рис. 2.3) (François Chollet, 2018).

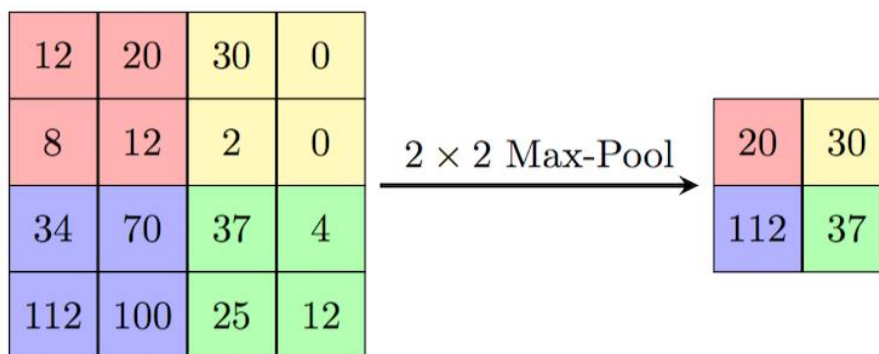


Рис.2.3 — Операція max pooling (François Chollet, 2018).

Max Pooling - це операція пулінгу, яка використовується в згорткових нейронних мережах для зменшення розміру вихідних шарів і виділення найбільш значущих ознак зображення або інших вхідних даних. Під час виконання операції Max Pooling, вихідний шар розділяється на неперекриваючі регіони (наприклад, 2x2 пікселі або блоки). У кожному регіоні зберігається тільки найбільше значення (максимум) з цих пікселів або блоків, а інші значення відкидаються. Таким чином, розмір вихідного шару зменшується, але зберігаються найбільш суттєві ознаки зображення.

Поміж інших операцій пулінгу, які використовуються в згорткових нейронних мережах, можна виділити наступні:

- Average Pooling (Середнє пулінг): У цій операції регіон розділяється на блоки, і для кожного блоку обчислюється середнє значення. Таким чином, замість вибору максимального значення, використовується середнє значення пікселів або блоків. Average Pooling також допомагає зменшити розмір шарів і знизити обчислювальну складність мережі.

- Min Pooling (Мінімальне пулінг): Ця операція обирає мінімальне значення з кожного регіону. Вона може використовуватися для виділення найменших ознак або для детекції найтемніших областей в зображенні.

- L2-norm Pooling: У цьому типі пулінгу використовується L2-норма для обчислення значення пулінгу. Вона виконує нормалізацію значень в регіоні, що дозволяє виділити сильні ознаки.

- Global Pooling: Ця операція виконує пулінг на всьому шарі в одному регіоні, без поділу на блоки або пікселі. Найпоширенішим випадком є Global Average Pooling, де обчислюється середнє значення всього шару. Це дозволяє отримати одне значення для кожного каналу або функції активації, зведенням розмірності шару до одного значення.

Крім субдискретизації з функцією максимуму можна використовувати і інші функції пулінгу, проте операція вибору максимального значення зазвичай дає кращі результати, ніж ці альтернативні рішення.

Max Pooling має кілька переваг. Воно допомагає знизити кількість параметрів та обчислювальну складність мережі, зменшує ризик перенавчання та покращує роботу зміщень об'єктів у зображенні. Крім того, воно робить згорткові нейронні мережі більш стійкими до невеликих зсувів або зміщень об'єктів на вхідному зображенні. Використання Max Pooling сприяє витягуванню найважливіших ознак зображення, забезпечує інваріантність до невеликих зміщень та зменшує розмір шарів, з якими працює

## 2.6 Dropout Layer

Проріджування (dropout) - один з найбільш ефективних і поширених прийомів регуляризації для нейронних мереж. Проріджування, яке застосовується до шару, полягає у видаленні (присвоєнні нуля) випадково вибраними ознаками в картах ознак на етапі навчання. Коефіцієнт

проріджування - це частка ознак, що обнуляються; зазвичай він вибирається в діапазоні від 0,2 до 0,5 (Srivastava N, and all, 2014).

Dropout - це техніка регуляризації, що застосовується в нейронних мережах, в більшості у випадку глибокого навчання. Dropout Layer (шар відсіювання) є спеціальним типом шару, який допомагає уникнути перенавчання та поліпшити загальну здатність моделі до узагальнення.

У Dropout Layer під час тренування моделі деякі випадкові нейрони тимчасово відключаються (відсіюються) з заданою ймовірністю. Це означає, що вони не беруть участь у передачі сигналів у мережі та не оновлюються під час зворотного поширення помилки. Внаслідок цього інші нейрони мають вирішувати завдання, не розраховуючи на наявність конкретних нейронів у мережі, що забезпечує більшу робастість моделі (рис. 2.4).

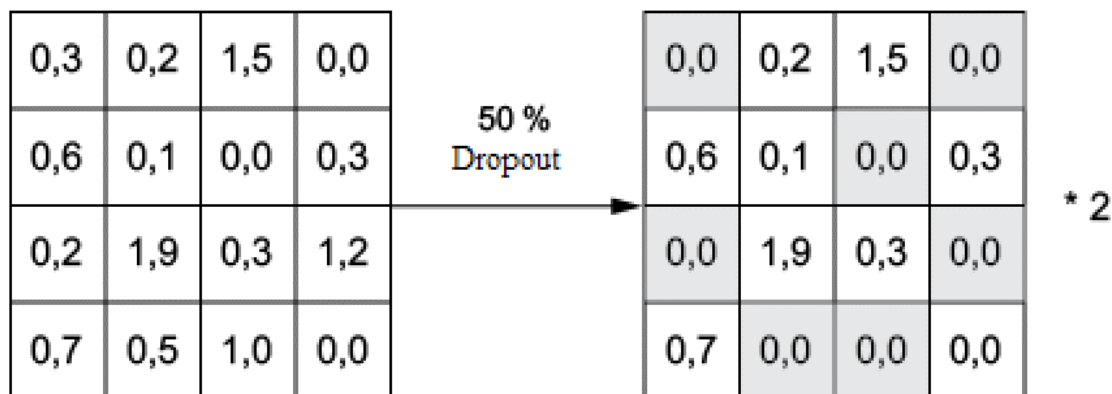


Рис. 2.4 — Застосування проріджування (François Chollet, 2018)

Після тренування, під час використання моделі для прогнозування, всі нейрони відновлюють свою активність, але їх ваги масштабуються на коефіцієнт Dropout ймовірності. Це забезпечує збереження статистичних властивостей моделі, що були вивчені під час тренування. Проріджування застосовується до карти ознак, де обнуляється половина ознак, а ті що

лишилися помножуються на коефіцієнт, обернений до коефіцієнту проріджування (François Chollet, 2018).

## 2.7 Batch Normalization Layer

Нормалізація є важливим процесом у машинному навчанні, який допомагає стабілізувати та прискорити навчання моделей. Пакетна нормалізація (Batch Normalization) є одним з типів нормалізації, що використовується для нормалізації вхідних даних у шарі нейронної мережі.

Принцип дії пакетної нормалізації полягає у нормалізації виходів попереднього шару в межах кожної партії (батчу) даних. Вона адаптивно обчислює середнє значення і дисперсію даних у межах партії та нормалізує їх до стандартного розподілу. Це дозволяє зберегти статистичні властивості даних та забезпечити стабільність навчання, навіть якщо середнє і дисперсія змінюються під час тренування.

Пакетна нормалізація (Batch Normalization) - це тип шару (BatchNormalization в Keras), який може адаптивно нормалізувати дані, навіть якщо середнє значення та дисперсія змінюються під час навчання. Його принцип дії базується на обчисленні експоненційного ковзного середнього та дисперсії даних, які спостерігаються під час процесу навчання (Ioffe S., Szegedy C., 2015). Шари BatchNormalization широко використовуються в розрахункових архітектурах згорткових нейронних мереж, які включені до бібліотеки Keras, такі як ResNet50, Inception V3 і Xception. Зазвичай шар BatchNormalization застосовується після шару згортки або повнозв'язаного шару (Hasani M., Khotanlou H, 2019).

## 2.8 Функції втрат

Функції втрат (loss functions) в згорткових нейронних мережах є математичними виразами, які використовуються для вимірювання різниці між прогнозованими значеннями моделі і справжніми мітками даних. Вони виступають об'єктивною мірою, яка дозволяє моделі оцінювати, наскільки добре вона виконує своє завдання, і сприяють процесу навчання шляхом оптимізації параметрів моделі.

У згорткових нейронних мережах, що використовуються для завдань класифікації, де потрібно призначити об'єкт до однієї з певних категорій, часто використовуються такі функції втрат:

1. Категорійна перехресна ентропія (Categorical Cross-Entropy). Ця функція втрат широко використовується в задачах багатокласової класифікації. Вона порівнює ймовірності прогнозованих класів з фактичними значеннями міток та обчислює втрати на основі розбіжності між ними.
2. Бінарна перехресна ентропія (Binary Cross-Entropy) використовується в задачах бінарної класифікації, де потрібно визначити об'єкт на два взаємовиключні класи. Вона вимірює втрати, порівнюючи прогнозовану ймовірність класу з фактичною бінарною міткою.
3. Середньоквадратична помилка (Mean Squared Error) використовується в задачах регресії, де модель прогнозує неперервні числові значення. Вона обчислює квадратичну різницю між прогнозованими значеннями та справжніми мітками (Ragab M.G. and all, 2020).

## 2.9 Оптимізатор

Для згорткових нейронних мереж широко використовуються різні оптимізатори, які відповідають за налаштування вагових коефіцієнтів мережі з метою зменшення функції втрат під час процесу навчання (Gao, Z. and all, 2020). Ось кілька розповсюджених реалізацій оптимізаторів, які використовуються для згорткових нейронних мереж:

1. Стохастичний градієнтний спуск (Stochastic Gradient Descent - SGD): Це один з найпростіших і поширених оптимізаторів. Він оновлює вагові коефіцієнти, знаходячи напрямок найшвидшого спаду функції втрат за допомогою градієнта (Bottou L., 2010).
2. Adam: Цей адаптивний метод оптимізації комбінує переваги алгоритмів RMSprop і Momentum. Він використовує експоненціально зменшувані середні квадратичні градієнти для адаптації швидкості навчання для кожного параметра окремо (Bouhassoune S. and all, 2017).
3. Adagrad: Цей метод оптимізації адаптивно регулює швидкість навчання для кожного параметра, використовуючи історію градієнтів. Він забезпечує більші корекції для рідко зустрічаючихся параметрів.
4. RMSprop: Цей метод оптимізації також використовує історію градієнтів, але він надає більшу вагу більш свіжим градієнтам. Він нормалізує градієнти, діленням на корінь середньоквадратичного значення попередніх градієнтів.

## 2.10 Метрики оцінки якості моделей. Confusion Matrix

У задачах машинного навчання для оцінки якості моделей і порівняння різних алгоритмів використовуються різноманітні метрики. Основою для розрахунку метрик є значення True Positive (TP), True Negative (TN), False Positive (FP) та False Negative (FN).

Матриця помилок (Confusion matrix) є інструментом для оцінки продуктивності моделі класифікації. Вона представляє собою таблицю, де кожен ряд відповідає дійсному класу, а кожен стовпець - прогнозованому класу. Кожна комірка матриці відображає кількість прикладів, які належать до певного дійсного класу та були прогнозовані вірно або неправильно. Матриця помилок є найпопулярнішим інструментом для візуального аналізу метрик.

Матриця помилок є табличним способом порівняння прогнозованих відповідей з фактичними результатами. Вона включає чотири можливі висновки: істинно позитивні (True Positive), хибно позитивні (False Positive), псевдонегативні (False Negative) і істинно негативні (True Negative).

Істинно позитивні (True Positive): Це випадки, коли модель правильно визначила позитивні елементи.

Хибно позитивні (False Positive): Це випадки, коли модель неправильно визначила негативні елементи як позитивні.

Псевдонегативні (False Negative): Це випадки, коли модель неправильно визначила позитивні елементи як негативні.

Істинно негативні (True Negative): Це випадки, коли модель правильно визначила негативні елементи.

Матриця помилок дозволяє зрозуміти, як добре модель виконує класифікацію, і може бути використана для обчислення різних метрик оцінки, таких як точність, чутливість, специфічність (рис.2.5) (Kulkarni A. and all, 2020).

|              |          | Predicted class      |                      |
|--------------|----------|----------------------|----------------------|
|              |          | <i>P</i>             | <i>N</i>             |
| Actual Class | <i>P</i> | True Positives (TP)  | False Negatives (FN) |
|              | <i>N</i> | False Positives (FP) | True Negatives (TN)  |

Рис. 2.5 - Матриця неточностей (Kulkarni A. and all, 2020)

### 2.11 Intersection Over Union

Оцінка перетину та об'єднання (Intersection Over Union, IoU) є стандартним показником ефективності для задачі сегментації об'єктів. При оцінці набору зображень, показник IoU вимірює подібність між передбаченою областю та істинною областю об'єкта на зображенні і визначається наступною формулою (2.1) (Rahman A., Wang Y., 2016):

$$IoU = \frac{TP}{FP+TP+FN} \quad (2.1)$$

де TP - істинно позитивні, FP - хибно позитивні, FN - псевдонегативні, TF - істинно негативні.

У чисельнику обчислюється площа перетину між передбаченою областю і істинною областю об'єкта, виражена у формі обмежувальних рамок. Знаменник представляє собою об'єднану площу, тобто площу, яка охоплює як

передбачену область, так і істинну область об'єкта, виражену також у формі обмежувальних рамок (рис.2.6).

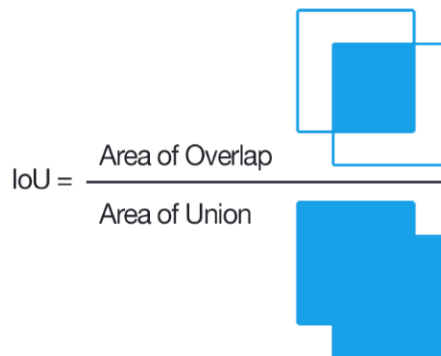


Рис.2.6 – Візуальне представлення Метрика Intersection Over Union  
(Rahman A., Wang Y., 2016)

## 2.12 Аналіз та розширення даних (аугментація)

Один з ефективних способів покращити продуктивність моделей глибокого навчання полягає у розширенні навчальних даних. Це можна зробити шляхом додавання більше різноманітних прикладів до навчальної вибірки. У розпізнаванні зображень це особливо важливо, оскільки згорткові нейронні мережі можуть використовувати трансляційну інваріантність для ефективного розпізнавання об'єктів.

Щоб забезпечити моделі здатність узагальнювати, необхідно включити в навчальну вибірку зображення з різними кутами повороту, освітленням, кольором і яскравістю (залежно від контексту). Такий робастий набір даних дозволяє моделі виявляти важливі ознаки об'єктів, навіть якщо вони

з'являються в інших частинах зображення, мають різні кольори або є спотвореними (Shorten, C., & Khoshgoftaar, T. M., 2019).

Аугментація (англ. data augmentation) в контексті машинного навчання є технікою, яка використовується для розширення набору даних для навчання штучних нейронних мереж. Ця техніка полягає в створенні нових прикладів даних шляхом застосування різноманітних перетворень до наявних зображень, звуків або текстів.

Наприклад, для зображень аугментація може включати обертання, зміщення, зміну масштабу, зміну контрастності, відображення вздовж горизонталі чи вертикалі та інші операції. Аугментація є корисним підходом для уникнення перенавчання, покращення загальної здатності моделі до узагальнення та підвищення її стійкості до варіацій у вхідних даних.

Основою аугментації є афінні перетворення. Афінні перетворення - це такі перетворення, які зберігають паралельність та пропорції між точками, прямими та площинами. До таких перетворень належать віддзеркалення, поворот, масштабування та зсув.

В даній роботі аугментація була застосована до обох навчальних вибірок.

### **2.13 Вибір мови програмування**

Для роботи в сфері науки про дані існує широкий спектр програмних бібліотек, платформ, модулів і інструментів, які ефективно втілюють загальноживані алгоритми та методи, що застосовуються у цій галузі. Проте

саме мова програмування Python стала одним з ключових інструментів для наукових обчислень, включаючи аналіз і візуалізацію великих обсягів даних. Незважаючи на те, що Python початково не був спеціально призначений для аналізу даних, завдяки великій екосистемі сторонніх модулів і пакетів, що активно розвиваються, він став придатним для вирішення практично будь-яких завдань, що виникають під час наукових досліджень.

Python відомий своєю простотою і легкістю вивчення. Синтаксис мови є зрозумілим і читабельним, що дозволяє розробникам швидко освоювати її. Це особливо важливо для початківців у галузі машинного навчання. Зокрема, Python має активну спільноту розробників, яка постійно вносить нові вдосконалення та розширює функціональність мови. Це означає, що завжди є підтримка та допомога у випадку виникнення проблем або питань. Також існує безліч онлайн-ресурсів, документації та підручників, які сприяють швидкому розвитку навичок в машинному навчанні з використанням Python (DeepLearningAI, 2022).

Python - це високорівнева інтерпретована мова програмування, яка була створена в 1991 році Гвідо ван Россумом. Вона відома своєю простотою синтаксису, зрозумілістю і легкістю вивчення, що робить її популярною серед початківців у програмуванні.

Python підтримує різноманітні стилі програмування, включаючи процедурне, об'єктно-орієнтоване і функціональне програмування. Вона має велику стандартну бібліотеку, яка надає готові інструменти для роботи з файлами, мережевими протоколами, базами даних, графікою та багато іншого. Однією з найбільших переваг Python є його велика екосистема

сторонніх бібліотек і пакетів. Вона включає бібліотеки для обробки даних (наприклад, NumPy і Pandas), наукових обчислень (SciPy), візуалізації (Matplotlib і Seaborn), машинного навчання (scikit-learn, TensorFlow, PyTorch) та багато інших (DeepLearningAI, 2022).

## 2.14 Огляд існуючих бібліотек для глибокого навчання

Вибір бібліотеки для реалізації алгоритмів машинного навчання один з ключових моментів на шляху створення успішного проекту. Наразі існує безліч пакетів, що надають можливість впровадження алгоритмів класифікації та сегментації. Найпопулярніші з них TensorFlow і PyTorch. Данні бібліотеки виконують ті ж самі основні завдання, пов'язані з глибоким навчанням. Вони спрощують збір даних, навчання моделей і створення прогнозів. Від розпізнавання облич до великих мовних моделей - багато нейронних мереж кодуються за допомогою TensorFlow або PyTorch. Колись ці бібліотеки помітно відрізнялися як зовнішньо, так і внутрішньо. З часом вони зблизилися навколо одного і того ж набору найкращих практик.

Тим не менш, у дослідники та розробники досі сперечаються про те, яка з них є найкращою. TensorFlow, випущений у 2015 році, був першим на сцені. Він домінує в комерційному полі та розробці продуктів, але багато користувачів скаржаться на його складність. PyTorch, випущений у 2016 році, вважається більш простим у вивченні та швидким у впровадженні. Він є фаворитом серед науковців і неухильно набирає популярність у промисловості. Однак, як відомо, він має проблеми з масштабуванням.

### 2.14.1 TensorFlow

TensorFlow - це бібліотека з відкритим вихідним кодом для розробки, навчання та розгортання моделей глибокого навчання. TensorFlow була вперше випущена у 2015 році компанією Google Brain. Спочатку її інтерфейс не був зручним для користувача і мав надлишкові API, що ускладнювало побудову та впровадження моделей. Багато з цих проблем було вирішено з часом за допомогою оновлень, а також шляхом інтеграції Keras (див. нижче) як інтерфейс за замовчуванням (TensorFlow, 2023).

TensorFlow має численні пакети для побудови моделей глибокого навчання та їх масштабування для комерційного розгортання. Користувачі TensorFlow можуть звертатися до сотень попередньо навчених моделей у Dev Hub та Model Garden. Dev Hub містить моделі plug-and-play, тоді як Model Garden призначений для більш досвідчених користувачів, яким зручно здійснювати кастомізацію. Він ефективно використовує пам'ять, дозволяючи паралельно навчати декілька нейронних мереж. Додатки TensorFlow можуть працювати на різних апаратних системах, включаючи CPU, GPU, TPU тощо.

TensorFlow Lite оптимізовано для мобільних та вбудованих моделей машинного навчання. Користувачі можуть вільно завантажувати та ділитися своїми експериментами з машинного навчання на [Tensorboard.dev](https://tensorboard.dev).

Найкраще підходить для створення готових до виробництва моделей глибокого навчання в масштабі. Проте, деякі користувачі все ще скаржаться, що інтерфейс досить складний.

### 2.14.2 PyTorch

PyTorch - це універсальна бібліотека з відкритим вихідним кодом для машинного навчання та науки про дані, зокрема глибокого навчання. Платформа пропонує безліч алгоритмів для швидкого конструювання та навчання глибоких штучних нейронних мереж (Pytorch, 2023).

Facebook випустив PyTorch у 2016 році - через рік після TensorFlow - і він швидко став популярним серед науковців та інших дослідників, зацікавлених у швидкому створенні прототипів. Це було пов'язано з його спрощеним інтерфейсом і тим, що за замовчуванням він виконує операції миттєво (на відміну від додавання їх до графа для подальшої обробки, як це робив TensorFlow).

PyTorch має багато можливостей, аналогічних можливостям TensorFlow. Дійсно, за роки, що минули з моменту їх запуску, кожна бібліотека була оновлена, щоб включити функції, які найбільше подобаються користувачам в іншій.

PyTorch має власні бібліотеки для попередньо навчених моделей. PyTorch Hub орієнтований на академічних користувачів, які хочуть експериментувати з дизайном моделей, а Ecosystem Tools містить попередньо навчені моделі. PyTorch ефективно використовує пам'ять і дозволяє навчати декілька моделей паралельно. Він підтримує різні типи обладнання.

PyTorch найчастіше використовують для швидкого створення прототипів моделей глибокого навчання. Код Pytorch працює швидко та ефективно. Тим не менше, деякі користувачі повідомляють, що PyTorch не справляється з великими проектами, великими наборами даних і складними

робочими процесами. Розробники, які створюють AI-продукти для масштабного розгортання, часто віддають перевагу TensorFlow.

### 2.14.3 Keras

Keras - це високорівнева бібліотека моделювання, яка надає прості будівельні блоки для створення глибоких нейронних мереж. Вона не включає низькорівневі операції, такі як тензорні маніпуляції і диференціювання, але використовує спеціалізовані і оптимізовані бібліотеки для цих завдань. Крім того, Keras не обмежена однією конкретною бібліотекою підтримки тензорів, Зараз Keras підтримує три такі бібліотеки: TensorFlow, Theano і Microsoft Cognitive Toolkit (CNTK) (Keras, 2023).

Keras являє собою зручний для початківців інструментарій для роботи з нейронними мережами. Це інтерфейс для роботи з TensorFlow. Історично склалось, що Інженер Google Франсуа Шуале випустив Keras у 2015 році як API для низки бібліотек глибокого навчання. Починаючи з 2020 року, Keras є ексклюзивом для TensorFlow. Keras виконує високорівневі завдання побудови нейронних мереж в TensorFlow, і як такий містить фундаментальні модулі, такі як функції активації, шари, оптимізатори і багато іншого.

Keras підтримує повнозв'язні нейронні мережі, згорткові нейронні мережі та рекурентні нейронні мережі, а також утилітарні шари, включаючи пакетну нормалізацію, відсіювання та об'єднання. Він призначений для спрощення кодування глибоких нейронних мереж.

Найкраще підходить для розробки мереж глибокого навчання. Проте доступний лише для користувачів TensorFlow. Якщо ви використовуєте

TensorFlow, ви використовуєте Keras.

В даній роботі як фреймворк для глибокого навчання було обрано Keras. Головними перевагою Keras є модульність та власне API, яке підтримує одразу з фреймворка як бекенд. Крім того Keras має простий самодокументований синтаксис для створення та навчання моделей.

### 3. ОГЛЯД СУЧАСНИХ АРХІТЕКТУР ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

#### 3.1 Архітектура LeNet

Архітектура LeNet - це класична нейронна мережа, яка була розроблена для задач розпізнавання зображень, зокрема для класифікації рукописних цифр. Вона була представлена Яном ЛеКуном в 1998 році (Lecun Y and all, 1998). Схематично зображена архітектура LeNet показана на рис. 3.1.

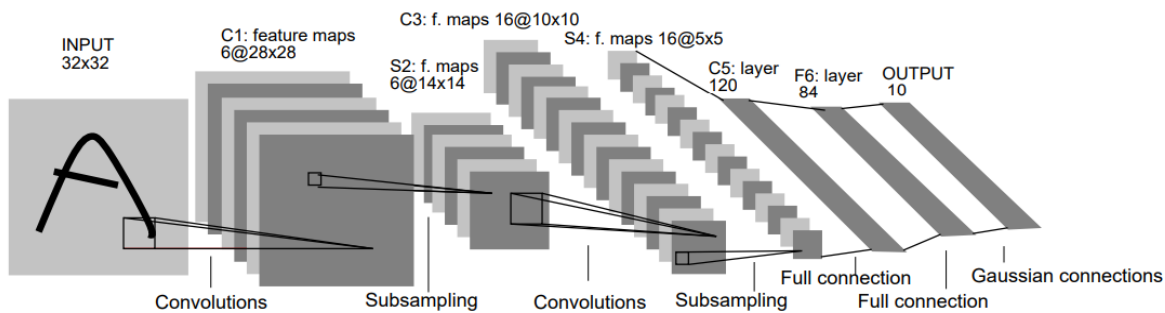


Рис. 3.1 — Перша архітектура згорткової нейронної мережі, запропонована Яном ЛеКуном в 1998 році (Lecun Y and all, 1998)

LeNet складається з послідовного розташування шарів згортки, підвибірки та повнозв'язаних шарів. Основні компоненти архітектури LeNet:

1. Згорткові шари (Convolutional layers): Виконують згортку зображення з допомогою фільтрів для вилучення важливих ознак зображення.
2. Шари підвибірки (Subsampling layers): Зменшують розмір вихідних зображень, зберігаючи важливу інформацію.
3. Повнозв'язані шари (Fully Connected layers): Приймають векторизовані функції з попередніх шарів та виконують класифікацію або регресію.

Архітектура LeNet вважається однією з перших успішних згорткових нейронних мереж, що використовувалися для розпізнавання зображень, і вона визначила основу для подальшого розвитку більш потужних моделей, таких як AlexNet, VGG і ResNet.

### **3.2 Архітектура AlexNet**

Архітектура AlexNet - це нейронна мережа, яка отримала широке визнання у 2012 році, коли вона перемогла в змаганні ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Архітектура була розроблена Алексом Кріжевським та його колегами зі Stanford University.

Однією з ключових інновацій архітектури AlexNet було використання глибокої нейронної мережі з великою кількістю параметрів, що дозволило досягти кращої точності класифікації. Також важливим було використання графічних обчислювальних пристроїв (GPU) для прискорення навчання мережі. Архітектура AlexNet стала відомою як одна з перших успішних глибоких згорткових нейронних мереж і знаменувала початок епохи глибокого навчання у зображеннях (Krizhevsky A. and all, 2012)

Архітектура AlexNet була революційною в світі комп'ютерного зору та зображення. Вона показала, що глибокі нейронні мережі можуть успішно вирішувати завдання класифікації зображень, що раніше були викликом для традиційних методів.

Архітектура AlexNet, яка зображена на рисунку 3.2, подібна до мережі LeNet, розробленої Яном Лекуном. Однак, в AlexNet використовується більше фільтрів на згорткових шарах і більше самі згорткові шари. Мережа включає



Архітектура EfficientNet складається з основних блоків, які повторюються кілька разів. Основний блок включає згорткові шари, функцію активації Swish, згорткові шари з фактором зменшення (reduction factor) і оператор згорткового пулінгу. Кількість повторень блоків і їх параметри варіюються залежно від масштабу моделі (рис.3.3). Ці блоки далі мають різну кількість підблоків, кількість яких збільшується, коли ми переходимо від EfficientNetB0 до EfficientNetB7 (Mingxing Tan and Quoc V. Le., 2018).

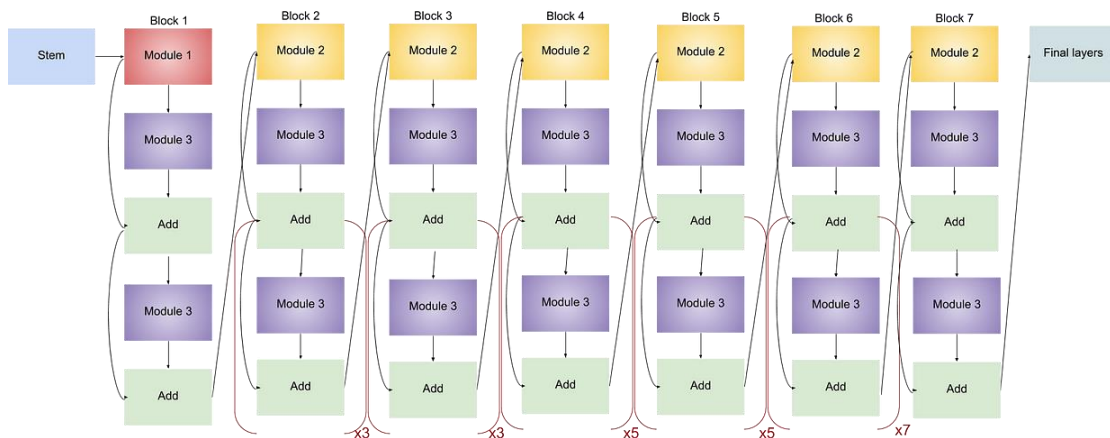


Рис.3.3 - Архітектура EfficientNet5

Архітектура EfficientNet досягає високої точності на різних завданнях зорового сприйняття, таких як класифікація зображень і семантична сегментація, з використанням меншої кількості параметрів і обчислювальних ресурсів порівняно з іншими архітектурами.

### 3.4 Архітектура ResNet

Архітектура ResNet (Residual Neural Network) є однією з найвпливовіших архітектур згорткових нейронних мереж, розроблених дослідниками з Microsoft Research у 2015 році. Вона була створена з метою розрішення

проблеми занепаду точності глибоких нейронних мереж при збільшенні їх глибини.

Основна інновація ResNet полягає у використанні блоків зі з'єднанням коротким шляхом, відомих як "residual blocks". Ці блоки дозволяють передавати інформацію напямую через мережу, оминаючи декілька послідовних шарів, що зменшує вплив проблеми зникнення градієнту та дозволяє побудувати надзвичайно глибокі мережі (Ronneberger, O. and al., 2015).

Дана структура дозволяє легко створювати глибокі мережі з сотнями або навіть тисячами шарів. Крім того, ResNet показує високу ефективність при навчанні на великих наборах даних, допомагаючи досягти вражаючих результатів у завданнях класифікації зображень та інших задачах зорового сприйняття. На прикладі цієї мережі було встановлено, що згортова нейронна мережа може використовувати тільки згорткові шари і якість розпізнавання значно збільшується при збільшенні глибини мережі. Дана мережа є однією з найбільш ефективних згорткових нейронних мереж на сьогоднішній день (рис.3.4).

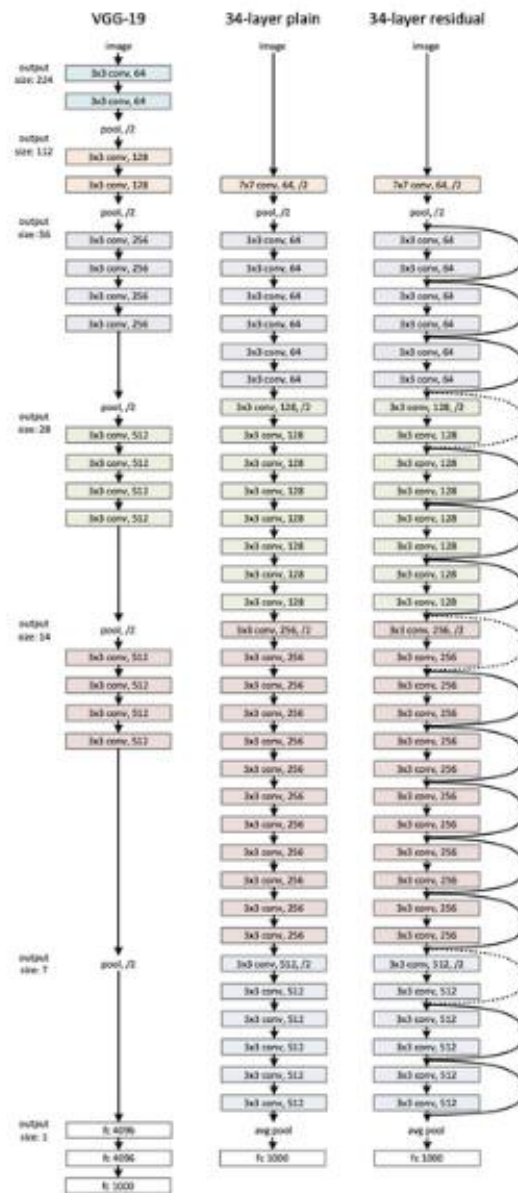


Рис. 3.4. — Архітектура ResNet (Ronneberger, O. and al., 2015)

### 3.5 Архітектури класу UNet

Дослідження моделей класифікації пошкоджень на основі CNN зосереджені на підвищенні швидкості та зручності використання.

Типовим використанням згорткових мереж є завдання класифікації, де вихід є однією з кількох міток. Проте для роботи з даними ДЗЗ, особливо в обробці супутникових знімків, необхідно, щоб на вихід модель надавала данні, що включають в себе локалізацію, тобто мітку класу передбачається призначати кожному пікселю зображення. Сегментація супутникових знімків є окремим напрямком застосування згорткових нейронних мереж. Тому доцільно залучити якийсь адаптивний механізм, здатний виділити найбільш інформативні ознаки з набору вхідних даних і згенерувати на їх основі інтерпретований (семантично значущий) результат. Таким найбільш придатним механізмом є підхід семантичної сегментації з використанням U-Net (Охрімчук, Р., Тишаєв, І., 2020).

U-Net - це згорткова нейронна мережа, яка була розроблена у 2015-ому році дослідниками з університету Фрейнбургу (Ronneberger, O. and al., 2015). Архітектура UNet має форму "U" і складається з двох основних частин: енкодера та декодера. Енкодер складається з послідовності згорткових шарів, які зменшують розмір зображення та збільшують кількість каналів ознак. Декодер використовує шари зворотного побудовного шляху для збільшення розміру зображення та поєднання ознак з енкодера для отримання більш детальної інформації. Це дозволяє виконувати точну сегментацію об'єктів на зображенні (рис. 3.5).

Для архітектури типу U-Net можна використовувати енкодер з різних архітектур, таких як AlexNet, VGG16, VGG19, ResNet18, ResNet50, ResNet34, InceptionV3, GoogleNet. При цьому декодер потрібно буде створювати вручну. Важливою умовою для вхідних зображень є наступне:

- Вхідне зображення повинно мати квадратну форму.
- Ширина та висота зображення повинні бути однаковими (наприклад, 32, 64, 128, 256 і т. д.).
- Глибина та архітектура нейронної мережі будуть визначатись розміром та кількістю вхідних зображень.

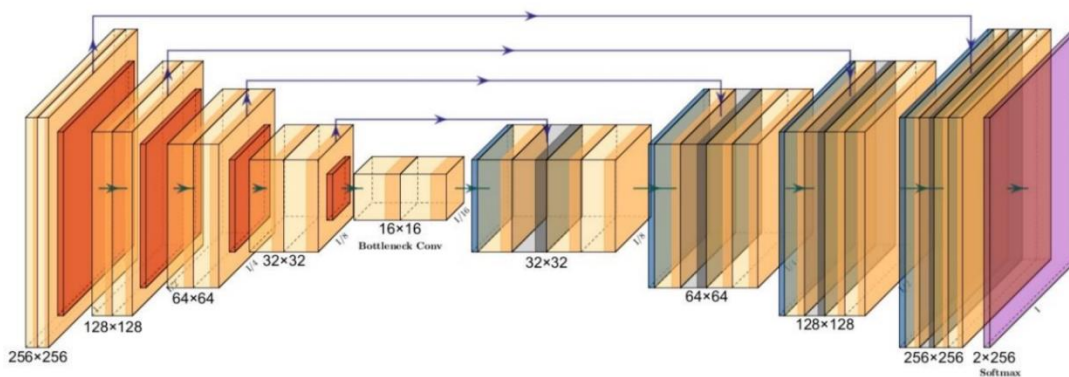


Рис.3.5 - Архітектура U-Net (Ronneberger, O. and al., 2015)

Особливістю U-Net є те, що кожен блок декодера додає до результуючого вектора вектор кодера на тому ж етапі. Це дозволяє мережі зберігати дрібні деталі об'єкта, тому ця архітектура успішно використовується для задач семантичної сегментації супутникових знімків (Кравченко, Ю., Тишаєва, А., 2018).

Як архітектура згорткової нейронної мережі для семантичної сегментації та класифікації була обрана архітектура U-Net з енкодерами EfficientNet5 та ResNet34 відповідно. Результат є ансамблем реалізацій від цих двох архітектур.

## 4. Практика реалізації проекту

### 4.1 Джерела даних для тренування

В даній роботі розглядались наступні джерела даних:

1) Векторний шар житлових будинків для міста Амастріче станом на 2016 рік є важливою частиною матеріалів дослідження. Файл надано ТОВ «ТВІС ЮКРЕЙН» у якості навчальних матеріалів для написання дипломної роботи магістра (рис.4.1). Формат даних є популярним ESRI Shapefile. Вектор включає 966 будинків, та покриває площу 5.98 km<sup>2</sup>.



Рис.4.1. - Векторний шар житлових будинків для міста Амастріче станом на 2016 рік

2) Як матеріали дослідження були використані, дані дистанційного зондування Землі, а саме космознімки з роздільною здатністю 0.3 метра за 21.05.2016 (до землетрусу) та за 07.07.2017 (після землетрусу), що були надані ресурсом SASPlanet у рамках використання студентської ліцензії (рис.4.2).



Рис.4.2 - Вихідні знімки дистанційного зондування: А - знімок за 21.05.2016 (до землетрусу), Б - знімок за 07.07.2017 (після землетрусу) (Brudko K. and all, 2022)

3) Вектор пошкоджень, створений вручну на основі космоснімку що датується після події, для невеликої ділянки, загальною площею близько 0.5 км<sup>2</sup> (рис. 4.3), який використовувався для тренування моделі на класифікації ділянок пошкоджень сегментованих контурів будівель.



Рис. 4.3 — Вектор пошкоджень

## 4.2 Приведення растрових даних

Для згорткових нейронних мереж вхідні дані складаються з послідовності тензорів та їх міток (або масок-тензорів). Щоб привести дані до необхідного формату, всі існуючі дані перетворюються на тензори, тобто зображення.

У випадку моделі сегментації, формат вхідних даних полягає в зображенні-масці. Оскільки маска початково представлена у векторному форматі, її необхідно перетворити на бінарний растр, де нульові значення відповідають відсутності цільового об'єкта, а одиничні значення - присутності об'єкта. Важливо, щоб маска та зображення мали однакову просторову роздільну здатність, ширину та висоту.

Для моделі класифікації, вхідні дані представлені у форматі зображення-мітки, де мітка є  $n$ -компонентним тензором, де  $n$  - кількість класів. У випадку бінарної класифікації, зображення, що містять цільовий об'єкт, позначаються міткою  $[0, 1]$ , тоді як зображення без об'єкта - міткою  $[1, 0]$ .

## 4.3 Створення навчальної вибірки

Набір навчальних даних для обох моделей сегментації та класифікації були створені за допомогою методу випадкових кроків. Після перетворення векторного файлу на растровий бінарний формат, наступним кроком є розбиття знімку та маски на менші зображення (рис.4.4). Цей процес здійснюється за допомогою методу випадкових кроків, який дозволяє створити велику та репрезентативну вибірку з невеликих зображень.

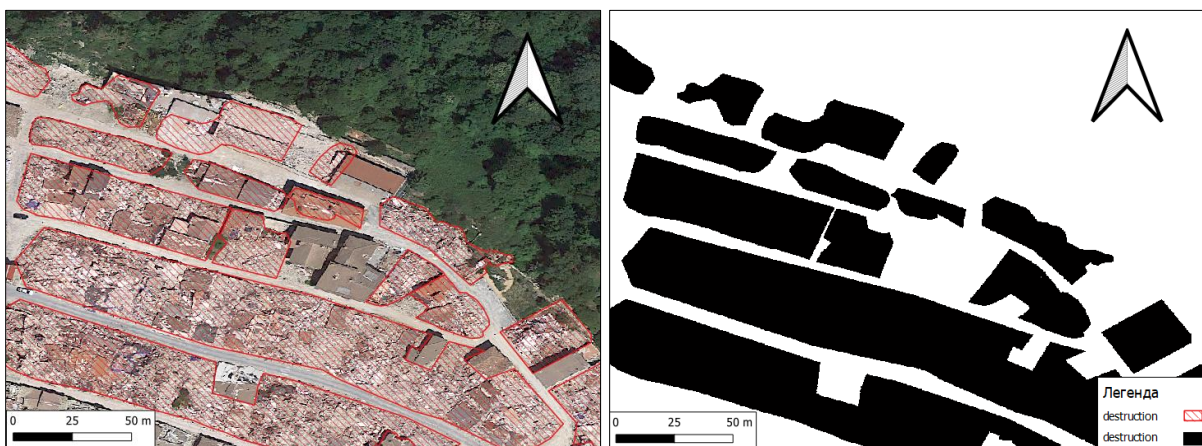


Рис. 4.4 — Приведення векторного файлу до потрібного формату

Основна ідея методу випадкових кроків полягає у створенні набору полігональних квадратів з довільним кутом повороту, який фіксується в таблиці атрибутів. Здійснюється вирізання зображення за екстентом кожного полігона. Потім отримані зображення повертаються на кут, протилежний куту повороту квадрата та виконується обрізання зайвих країв зображення.

Таким чином, застосування методу випадкових кроків дозволяє поділити знімок та маску на менші зображення, що сприяє створенню репрезентативної вибірки для подальшого аналізу та обробки.

Навчальна вибірка для моделі сегментації побудована на основі вектору житлових будинків міста Аматрісе. Згідно з робочим процесом, вихідний вектор житлових будинків для міста Аматріс, актуальний на 2016 рік, необхідно розділити на 3 частини у співвідношенні: 70%, 20% і 10% відповідно для навчальної, валідаційної та тестової вибірок. Важливо, щоб дані в цих 3 вибірках мали подібний розподіл. Загальна кількість об'єктів житлової інфраструктури на досліджуваній території становить 966. Навчальна частина векторних даних була зведена до растрового бінарного формату. Ці дані були

використані як навчальна вибірка для навчання згорткової нейронної мережі для сегментації супутникових знімків: виділення контурів житлових будинків (Brudko K. and all, 2022).

Якість моделі глибокого навчання безпосередньо залежить від кількості вибірок навчальної вибірки. Важливою частиною підготовки валідної навчальної вибірки є аугментація. Ця техніка використовується для збільшення різноманітності навчальної вибірки шляхом застосування випадкових (але реалістичних) перетворень, таких як поворот зображення, зміна розміру, дзеркальне відображення та масштабування (Brudko K. and all, 2022).

Під час цього дослідження аугментацію було застосовано до набору даних, який є навчальною вибіркою для моделі сегментації, кількість зразків було збільшено до 100 000 обрізаних зображень розміром 128x128 (рис. 4.5). Основні перетворення зображень, які були застосовані до навчальних даних, - це афінні перетворення. Першою була використана операція дзеркального відображення. Ця операція дозволяє збільшити набір даних без спотворення вхідних зображень і без додавання сторонніх ознак. Також до навчальної вибірки було застосовано доповнення шляхом повороту зображень на довільний кут (від 0 до 360).

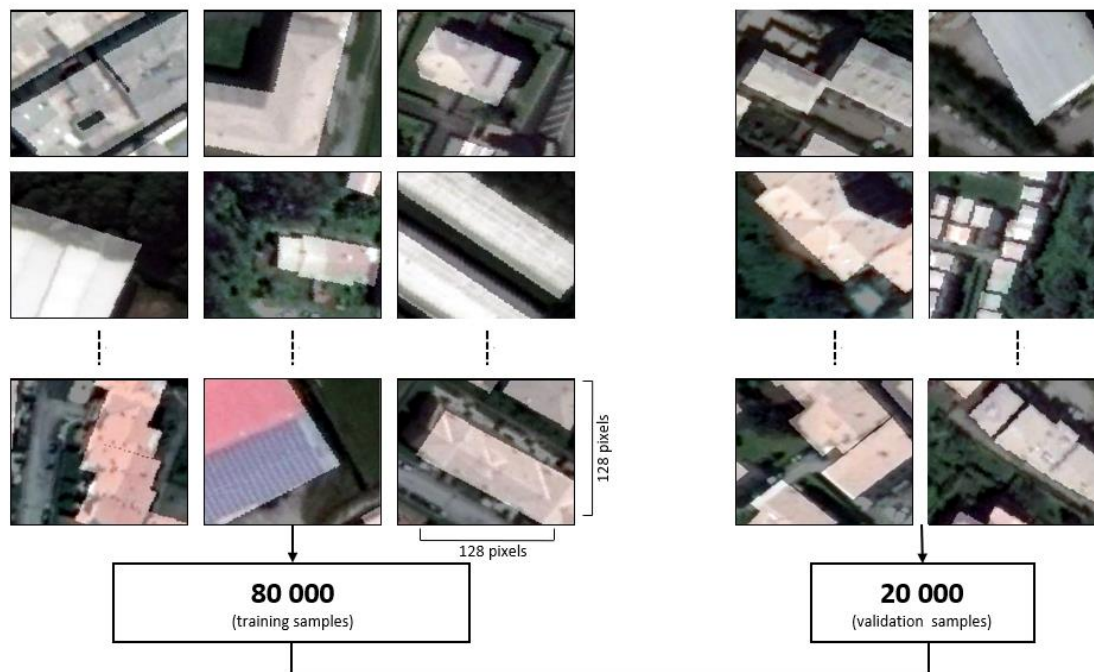


Рис.4.5 - Навчальна вибірка: Базове зображення (напівпрозорий шар) накриває вхідний RGB-шар (Brudko K. and all, 2022)

Навчальна вибірка для класифікації будинків на пошкоджені та не пошкоджені була створена на основі вектору пошкоджень. Отже, векторний файл пошкоджень потрібно перетворити на растровий бінарний формат та створити об'єкт зображення - мітка. Таким чином, отриманий вхідний набір даних для класифікаційної моделі складатиметься з зображень, де будівлі та їх пошкодження позначаються відповідними мітками. Цей підхід дозволяє моделі використовувати зображення-мітку як вхідні дані, що дозволяє здійснювати класифікацію будівель на пошкоджені та непошкоджені.

За допомогою методу випадкових кроків, що доцільний для лімітованої території, було отримано набір зображень розміром 128x128 з мітками (пошкоджені чи непошкоджені). Після цього за допомогою аугментації було розширено вибірку зруйнованих будинків, оскільки була присутня

незбалансованість класів у вхідному наборі даних. Таким чином було отримано навчальну вибірку з 4500 семплів.

#### 4.4 Підготовка та створення моделі

Як фреймворк для побудови моделі було використано бібліотеку Keras для мови програмування Python. Keras надає можливість використовувати вже навчені моделі (на великих відкритих наборах даних) та їхні архітектури через спеціальний модуль `keras.applications` (Brudko K. and all, 2022).

Keras дозволяє будувати моделі у формі блоків, які складаються з декількох шарів або окремих шарів. Для створення найпростішої згорткової моделі достатньо ініціалізувати базові шари.

Шар вхідних даних — клас `keras.layers.Input`. `Input()` використовується для створення екземпляра керасівського тензора. Тензор Кераса - це символічний тензороподібний об'єкт, який ми доповнюємо певними атрибутами, що дозволяють нам будувати модель Кераса, просто знаючи входи та виходи моделі. Наприклад, якщо `a`, `b` і `c` - тензори Кераса, то можна зробити так: `model = Model(input=[a, b], output=c)`. Приймає на вхід кортеж `(None, width, height, bands)`, де— `None` вказує на те, що зображень може бути будь-яка кількість.

```
from keras.layers import Input
input_layer = Input(shape=(img_height, img_width, 8),
name='image_input')
```

Згортковий шар — `keras.layers.Conv2D` або `keras.layers.SeparableConv2D`. На вхід цього шару подаємо кількість фільтрів, розмір ядра згортки та функцію активації.

```

from keras.layers import Conv2D
x = Conv2D(filters=nfilters, kernel_size=(size, size),
           padding=padding, kernel_initializer=initializer) (tensor)

```

Шар субдискретизації — `keras.layers.MaxPool2D` ініціалізуємо параметри пулінгу, за змовчуванням дорівнює 2.

```
conv1_out = MaxPooling2D(pool_size=(2, 2)) (conv1)
```

У якості засобу від перенавчання, був включений шар `Dropout`. Поєднуючись як з повнозв'язними шарами (`Dense`) так і зі згортковими шарами (`Convolution`, `Separate Convolution`), даний шар приймає лише - коефіцієнт проріджування.

```

from keras.layers import Dropout
conv4_out = Dropout(0.5) (conv4_out)

```

Готові архітектури можна використовувати повністю або тільки згорткову частину. Для створення моделі сегментації було використано архітектуру `U-net` з ядром `efficientnetb5`. Приклад побудови узагальненої схематичної моделі сегментації зображення за допомогою фреймворку `Keras` продемонстровано нижче:

```

from keras.models import *
from keras.layers import *
from keras.optimizers import *
import keras
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau

def conv_block(tensor, nfilters, size=3, padding='same',
               initializer="he_normal"):
    x = Conv2D(filters=nfilters, kernel_size=(size, size),
              padding=padding,
              kernel_initializer=initializer) (tensor)
    x = BatchNormalization() (x)
    x = Activation("relu") (x)

```

```

    return x

def deconv_block(tensor, residual, nfilters, size=3, padding='same',
strides=(2, 2)):
    y = Conv2DTranspose(nfilters, kernel_size=(size, size),
                        strides=strides, padding=padding)(tensor)
    y = concatenate([y, residual], axis=3)
    y = conv_block(y, nfilters)
    return y

def Unet(img_height, img_width, nclasses, filters=64):
    # down
    input_layer = Input(shape=(img_height, img_width, 8),
name='image_input')
    conv1 = conv_block(input_layer, nfilters=filters)
    conv1_out = MaxPooling2D(pool_size=(2, 2))(conv1)

    # up
    deconv6 = deconv_block(conv5, residual=conv4, nfilters=filters * 8)
    deconv6 = Dropout(0.5)(deconv6)

    # output
    output_layer = Conv2D(filters=nclasses, kernel_size=(1,
1))(deconv9)
    output_layer = BatchNormalization()(output_layer)
    output_layer = Activation('sigmoid')(output_layer)

    model = Model(inputs=input_layer, outputs=output_layer,
name='Unet')
    return model

model = Unet(128, 128, nclasses=1, filters=64)

```

Вище частково представлений код створення моделі згорткової нейронної мережі. Тут модель являє собою граф тензорів, де кожен шар пов'язаний з іншими шарами. Таким чином, кожен шар згортки, повинен отримувати у якості параметру попередній шар. Для цього після ініціалізації кожного шару використовуються круглі дужки, в яких вказується вихідний тензор іншого шару, який є входом для поточного шару. Наприклад після

ініціалізації пулінгу `MaxPooling2D` було зарадно його розмір `pool_size=(2, 2)` та передано шар вихідного тензору (`conv1`). Після створення графу ініціалізуємо мовель за допомогою класу `Model`, який приймає перший і останній тензори як вхідні. Щоб візуалізувати граф моделі та переглянути зміну розмірів тензорів на вході і виході, можна використовувати функцію `model.summary()`. Результат цієї функції наведено на рис. 4.6.

Код для створення моделі сегментації наведений в додатку А.

```

decoder_stage3a_conv (Conv2D) (None, 256, 256, 8) 5760 decoder_stage3_concat[0][0]
decoder_stage3a_bn (BatchNormal (None, 256, 256, 8) 32 decoder_stage3a_conv[0][0]
decoder_stage3a_relu (Activatio (None, 256, 256, 8) 0 decoder_stage3a_bn[0][0]
decoder_stage3b_conv (Conv2D) (None, 256, 256, 8) 576 decoder_stage3a_relu[0][0]
decoder_stage3b_bn (BatchNormal (None, 256, 256, 8) 32 decoder_stage3b_conv[0][0]
decoder_stage3b_relu (Activatio (None, 256, 256, 8) 0 decoder_stage3b_bn[0][0]
decoder_stage4_upsampling (UpSa (None, 512, 512, 8) 0 decoder_stage3b_relu[0][0]
decoder_stage4a_conv (Conv2D) (None, 512, 512, 4) 288 decoder_stage4_upsampling[0][0]
decoder_stage4a_bn (BatchNormal (None, 512, 512, 4) 16 decoder_stage4a_conv[0][0]
decoder_stage4a_relu (Activatio (None, 512, 512, 4) 0 decoder_stage4a_bn[0][0]
decoder_stage4b_conv (Conv2D) (None, 512, 512, 4) 144 decoder_stage4a_relu[0][0]
decoder_stage4b_bn (BatchNormal (None, 512, 512, 4) 16 decoder_stage4b_conv[0][0]
decoder_stage4b_relu (Activatio (None, 512, 512, 4) 0 decoder_stage4b_bn[0][0]
final_conv (Conv2D) (None, 512, 512, 1) 37 decoder_stage4b_relu[0][0]
sigmoid (Activation) (None, 512, 512, 1) 0 final_conv[0][0]
=====
Total params: 21,870,142
Trainable params: 582,536
Non-trainable params: 21,287,606

```

Рис. 4.6 - Граф тренування моделі

З цього повідомлення можна зрозуміти скільки розрахункових параметрів є на кожному з шарів моделі, які тензори та якого розміру подаються на кожен вхід до шару

В даній роботі були використані архітектури ResNet34 та efficientnetb5 для класифікації та сегментації. Keras надає можливість використовувати вже натреновані моделі (на великих відкритих датасетах) та їх архітектури через спеціальний модуль `keras.applications`.

Наведені вище архітектури можна використовувати повністю або тільки згорткову частину. Для того, щоб створити модель класифікації, достатньо імпортувати тільки згорткову частину та додати повнозв'язну. Код, який наведений нижче демонструє створення моделі для класифікації на основі архітектури ResNet34 з натренованими ваговими коефіцієнтами "imagenet":

```
from keras import applications
from keras.layers import Dense, Flatten
from keras.models import Model

conv_part = applications.resnet34.ResNet34(weights= None,
include_top=False, input_shape= (img_height, img_width, 3))
f1 = Flatten()(conv_part.output)
dense1 = Dense(128, activation = 'relu')(f1)
dense2 = Dense(1, activation = 'sigmoid')(dense1)
model = Model(conv_part.input, dense2)
```

Після створення моделі в середовищі Python, її можна зберегти до файлу з розширенням `.h5`. При збереженні, модель зберігає свої вагові коефіцієнти, що означає, що може бути збережена натренована модель, а також зберігає свою структуру.

```
checkpoint = ModelCheckpoint(r"../UNET/MODELS/UNET_Model.h5",
monitor = 'val_loss',
verbose = 1,
save_best_only = True,
save_weights_only = False,
mode = 'auto')
```

## 4.5 Навчання моделі

Код аугментації зображень та завантаження навчальної вибірки для моделі розміщено у додатку В. Завантаження та компіляція моделі відбувається за допомогою модуля keras:

```
reduce = ReduceLROnPlateau(monitor='val_loss', factor=0.1,
                           patience=10, verbose=1, mode='auto', min_delta=0.001,
                           cooldown=0, min_lr=0.00001)
history = model.fit(image_train, mask_train, batch_size=8,
                   callbacks=[checkpoint, reduce], epochs=100000,
                   validation_data=(image_val, mask_val),
                   shuffle=True, verbose=1)
```

Для компіляції моделі потрібно вказати функцію втрат, оптимізатор і метрику точності. Якщо метрика точності вказана як 'acc', Keras автоматично обере відповідну функцію точності. Функція втрат і оптимізатор можуть бути визначені як назви функцій у вигляді рядка або самостійно ініціалізовані з параметрами. Крім того, Keras надає можливість створювати власні функції втрат і метрик точності, що є корисним у складних завданнях.

Після компіляції модель готова для тренування, але для збереження історії навчання потрібно використовувати метод fit класу keras.models.Model.

На основі спостережень за показниками точності під час тренування моделі можна зробити висновки щодо ефективності використання даної архітектури та про особливості даних (рис. 4.7).

```

225/225 [=====] - 269s 995ms/step - loss: 1.0163 - iou_score: 0.4989 - val_loss: 1.5656 - val_iou_score: 0.2622
Epoch 2/15000
225/225 [=====] - 101s 449ms/step - loss: 0.5910 - iou_score: 0.6814 - val_loss: 1.6913 - val_iou_score: 0.2285
Epoch 3/15000
225/225 [=====] - 101s 447ms/step - loss: 0.4378 - iou_score: 0.7585 - val_loss: 1.1294 - val_iou_score: 0.4612
Epoch 4/15000
225/225 [=====] - 98s 436ms/step - loss: 0.3360 - iou_score: 0.8115 - val_loss: 0.8627 - val_iou_score: 0.6311
Epoch 5/15000
225/225 [=====] - 101s 449ms/step - loss: 0.2629 - iou_score: 0.8504 - val_loss: 0.2297 - val_iou_score: 0.8774
Epoch 6/15000
225/225 [=====] - 101s 447ms/step - loss: 0.2076 - iou_score: 0.8801 - val_loss: 0.1597 - val_iou_score: 0.9122
Epoch 7/15000
225/225 [=====] - 99s 439ms/step - loss: 0.1654 - iou_score: 0.9033 - val_loss: 0.1326 - val_iou_score: 0.9272
Epoch 8/15000
225/225 [=====] - 103s 455ms/step - loss: 0.1335 - iou_score: 0.9212 - val_loss: 0.1149 - val_iou_score: 0.9378
Epoch 9/15000
225/225 [=====] - 100s 445ms/step - loss: 0.1081 - iou_score: 0.9355 - val_loss: 0.0988 - val_iou_score: 0.9481
Epoch 10/15000
225/225 [=====] - 101s 446ms/step - loss: 0.0892 - iou_score: 0.9464 - val_loss: 0.0916 - val_iou_score: 0.9536
Epoch 11/15000
225/225 [=====] - 102s 453ms/step - loss: 0.0744 - iou_score: 0.9551 - val_loss: 0.0770 - val_iou_score: 0.9619
Epoch 12/15000
225/225 [=====] - 100s 445ms/step - loss: 0.0624 - iou_score: 0.9622 - val_loss: 0.0708 - val_iou_score: 0.9671
Epoch 13/15000
225/225 [=====] - 101s 450ms/step - loss: 0.0532 - iou_score: 0.9678 - val_loss: 0.0674 - val_iou_score: 0.9692
Epoch 14/15000
225/225 [=====] - 100s 443ms/step - loss: 0.0473 - iou_score: 0.9718 - val_loss: 0.0620 - val_iou_score: 0.9731
Epoch 15/15000
225/225 [=====] - 99s 441ms/step - loss: 0.0431 - iou_score: 0.9748 - val_loss: 0.0608 - val_iou_score: 0.9757
Epoch 16/15000

```

Рисунок 4.7 Процес тренування моделі сегментації

Ідеальною ситуацією є постійне покращення показників на навчальній вибірці, а при достатній кількості епох точність повинна бути досить високою (95% і вище), а втрати - низькими. Низька точність на навчальній вибірці може бути обумовлена неправильною анотацією даних або наявністю помилок у мітках зображень. У такому випадку важливо переглянути вхідні дані та видалити непридатні зразки. Однак, вивчення лише природи даних може бути недостатнім, якщо архітектура мережі не відповідає складності даних. Це може бути через недостатню кількість фільтрів у згорткових шарах, які не відтворюють важливі ознаки належним чином і стають обмежувачим фактором моделі. Недостатня глибина моделі або, навпаки, надмірну глибину моделі також може призвести до зниження точності при зростанні кількості епох тренування.

Слідкування за функцією втрат і точністю на тестових даних дозволяє оцінювати, наскільки добре модель узгоджується з результатами на незалежних даних. Погіршення цих показників може вказувати на

перенавчання моделі (overfitting), а якщо показники залишаються стабільними і не задовольняють вимогам, це може свідчити про застрягання в локальному мінімумі або про недостатню швидкість навчання.

Процес повторного тренування, налаштування гіперпараметрів мережі та очищення вхідних даних зазвичай повторюються допоки усі вищезазначені показники не досягнуть бажаних значень. Такий підхід дозволяє вдосконалити модель і підлаштувати її під конкретну задачу, забезпечуючи більш точні результати.

#### **4.6 Реалізація алгоритму сегментації**

За допомогою методу випадкових кроків було створено масив вхідних зображень для навчання моделі. Для цього було використано модуль `numpy`, реалізація методом зрізів. Це значно пришвидшує продуктивність коду та зменшує час на обробку перед передачею знімків на вхід мережі, крім того не потрібно зберігати нарізані заздалегідь знімки у необхідному для навчання моделі форматі.

Розмір вхідного тензору  $128 \times 128$  є досить великим, проте такі налаштування потрібні для того, щоб згорткова частина виокремила найбільше необхідних абстрактних ознак.

Таким чином, кожна подальша операція пулінгу буде зменшувати розмір тензору вдвічі. Через втрачання пікселів, може зменшуватись точність виявлення ознак, тож глибина мережі частково залежить від розміру вхідного зображення. Як наслідок, і якість класифікації буде залежати від розміру вхідного зображення, тобто його розмір повинен задовольняти параметри і

масштабування моделі та можливість виокремлення найбільшої кількості абстрактних ознак або, у цьому випадку, пошкоджень.

Метод випадкових кроків для генерації вхідного датасету моделі є критично важливим, адже в умовах коли територія для підготовки навчальної вибірки є лімітована важливо створити якість зображення мітки без ресурсоемної підготовки.

Навчання моделі для сегментації контурів будівель було зупинено при втратах перевірки 0,043 і точності перевірки 0,98. Точність перевірки вимірювалася на частині набору даних, яку модель ніколи не бачила (Brudko K. and all, 2022).

Результатом сегментації з використанням методів глибокого навчання є маска ймовірності, яка має той самий розмір, що й вхідне зображення, де кожен піксель є ймовірністю присутності цільового класу. Результат сегментації записується в окрему растрову маску такого ж розміру, як і вхідне зображення. Растр ймовірностей результату сегментації продемонстровано на рисунку 4.8, а (Brudko K. and all, 2022).

Пост обробка сегментації є важливим етапом для отримання вектора контурів будівель. Бінаризація результату сегментації, яку проводили за пороговим значенням, яке обирається експериментально. Зазвичай це значення  $\geq 0.5$  (рис. 4.8, б).

На рисунку 6.4, б можна помітити ділянки розміром 1-2 пікселя, які не несуть корисної інформації, а лише створюють шум. Позбутися таких ділянок можна за допомогою морфологічної обробки. Існують чотири основні операції морфологічної обробки, а саме ерозія, розширення, відкриття і закриття. Перші дві операції можна порівняти з буферизацією векторних даних, де

маємо негативні та позитивні значення розміру буферу відповідно. Останні дві операції відповідають послідовності ерозії та розширення, або навпаки, відповідно. Морфологічна обробка з операцією відкриття (opening) показала себе найкраще у цьому конкретному випадку і дозволила якісно розділити контури будинків та зберегти важливу інформацію. (рис. 4.8, в) (Brudko K. and all, 2022).

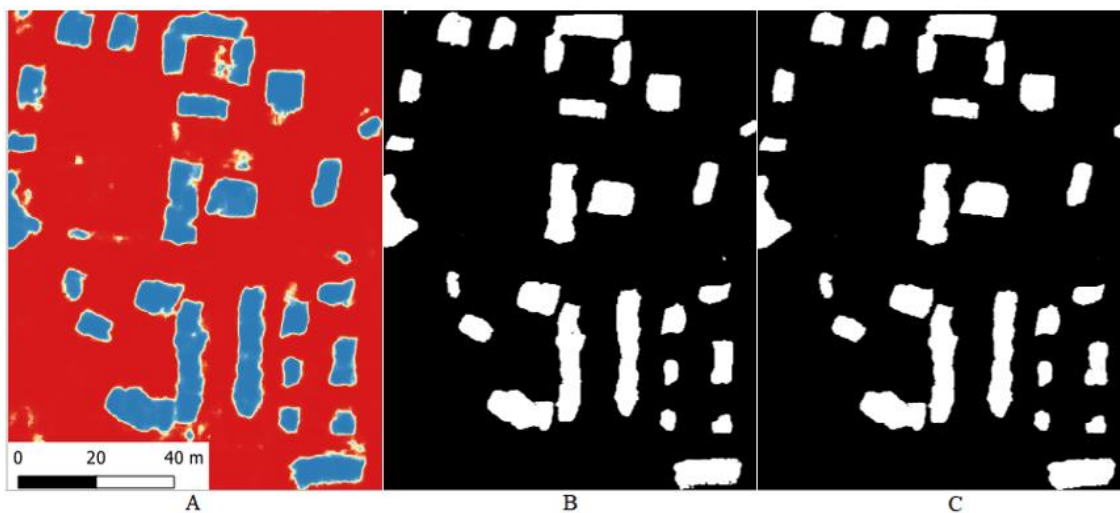


Рисунок 4.8 - Пост обробка результатів сегментації: а) растр ймовірностей результатів сегментації, б) результат постобробки растру бінаризації, в) результат морфологічної обробки растру (Brudko K. and all, 2022)

Після морфологічної обробки бінарний растр був перетворений у векторне полігональне представлення (рис. 4.9).



Рис.4.9 - Результат векторного предствлення сегментації контурів будинків

#### 4.7 Реалізація алгоритму класифікації

Основні завдання класифікації зображень за допомогою згорткової нейронної мережі це виділення репрезентативних ознак об'єкта на зображенні з різними масштабами, за допомогою згорткових шарів та шарів підвищення дискретності та класифікація виділених ознак, що здійснюється за допомогою повнозв'язної частини згорткової нейронної мережі.

Такі згорткова нейронна мережа використовується для побудови бінарної класифікації, де кількість нейронів на останньому повнозв'язному шарі відповідатиме кількості класів.

Процес підготовки навчальної вибірки відповідає процесу описаному у розділі 6.4 «Реалізація алгоритму сегментації».

Після класифікації зображення результатом є ймовірність, яка вказує на ймовірність належності зображення до цільового класу: Пошкоджених або не

пошкоджених будівель. Ймовірність знаходиться в діапазоні від 0 до 1 і записується у вигляді окремого растру, де розмір пікселя відповідає розміру пікселя вихідного зображення.

Для виконання задачі класифікації сегментованих контурів будинків на 2 класи пошкоджені та не пошкоджені була використана архітектура U-net з ядром ResNet34 (Brudko K. and all, 2022). Важливим фінальним кроком при створенні моделі класифікації є функція активації. Для даної задачі найкраще підходить бінарна класифікація, оскільки ми маємо лише 2 вихідних класи. Найбільш підходящою функцією активації для Binary Classification є Sigmoid/Logistic Activation Function. Підготовка датасету для класифікація зображень базується на методі випадкових кроків, що є доцільним у випадку обмеженої території дослідження. Розміри сітки (ширина і висота) були згенеровані, щоб відповідати розміру вхідного тензора для моделі. Результатом класифікації є растр ймовірностей віднесення зображення до цільового класу.

Для того щоб оцінити точність моделі класифікації було використано матрицю помилок (Confusion matrix), що популярним інструментом для оцінки точності моделі класифікації. Вона представляє собою таблицю, яка дозволяє порівняти прогнозовані класи з фактичними класами у вхідних даних. Зазвичай, у матриці помилок розташовані чотири значення:

True Positives (TP) - кількість правильно класифікованих позитивних прикладів, де модель правильно визначила, що певні приклади належать до певного класу.

False Positives (FP) - кількість неправильно класифікованих позитивних прикладів, де що модель помилково визначила, що певні приклади належать до певного класу, хоча насправді вони не належать.

False Negatives (FN) - кількість неправильно класифікованих негативних прикладів, де модель помилково не визначила, що певні приклади належать до певного класу, хоча насправді вони мали би належати.

True Negatives (TN) - кількість правильно класифікованих негативних прикладів, де модель правильно визначила, що певні приклади не належать до певного класу (Охрімчук, Р., Тишаєв, І., 2020).

*Таблиця № 4.1 Матриця помилок моделі класифікації*

|                    |                       | Справжній клас            |                          |         |
|--------------------|-----------------------|---------------------------|--------------------------|---------|
|                    |                       | Позитивна аномалія        | Не позитивна аномалія    | Загалом |
| Прогнозований клас | Позитивна аномалія    | 860<br>істинно позитивних | 20<br>хибно позитивних   | 880     |
|                    | Не позитивна аномалія | 14<br>хибно негативних    | 72<br>істинно негативних | 86      |
| Загалом            |                       | 874                       | 92                       | 966     |

За допомогою матриці помилок можна обчислити загальну точність моделі. Загальна точність моделі може бути визначена, розрахувавши

відношення правильно класифікованих прикладів (TP + TN) до загальної кількості прикладів (4.1) (Охрімчук, Р., Тишаєв, І., 2020).

$$\text{точність} = \frac{\Sigma \text{ істинно позитивних} + \Sigma \text{ істинно негативних}}{\Sigma \text{ загальної сукупності}} \quad (4.1)$$

Таким чином загальна точність моделі класифікації складає 89%

Кроки постпроцесінгу також включали бінаризацію та морфологічну обробку. Результат роботи моделі класифікації був перенесений на векторизований результат сегментації контурів будівель (рис. 4.10).



Рисунок 4.10 – Векторизований результат моделей сегментації та класифікації (Brudko K. and all, 2022)

## ВИСНОВКИ

Задача автоматизованої класифікації пошкоджень будівель через складність визначення ознак пошкоджень є досить нетривіальною. Використання даних дистанційного зондування може відігравати значну роль в оцінці результатів автоматизованої класифікації пошкоджень будівель.

Використання методів машинного навчання у роботі з космічними знімками дозволяє покращити точність, ефективність та якість аналізу цих зображень.

Методи машинного навчання, такі як класифікатори на основі дерева рішень, методи опорних векторів (SVM), нейронні мережі тощо, використовуються для класифікації об'єктів на основі ДЗЗ. Найбільш популярна за останні роки сегментація даних ДЗЗ, тобто виділення окремих об'єктів або регіонів на зображенні дозволяє аналізувати та вимірювати параметри об'єктів, таких як розмір, форма та площа.

Сегментація CNN (імплементації U-Net) є ефективним інструментом для дешифрування пошкоджень житлової інфраструктури. Тому використання технології U-Net є цілком прийнятним і доцільним, особливо за наявності достовірних першочергових даних, придатних для навчання нейронних мереж (Охрімчук, Р., Тишаєв, І., 2020).

Було запропоновано алгоритми сегментації та класифікації пошкодженої інфраструктури внаслідок масштабного землетрусу потужністю 6,2 МВт стався в Центральній Італії у місті Амастріче. Для цього було використано методи глибокого навчання для автоматичного розпізнавання

пошкоджених будівель, що дозволяє швидко і просторово оцінити вплив природних загроз, тим самим прискорюючи реагування на надзвичайні ситуації. Шляхом використання автоматизованих методів, побудованих на основі згорткових нейронних мереж, було ефективно та відносно швидко отримано інформацію на основі даних дистанційних зондувань, виділяючи контури об'єктів будівельної інфраструктури, оцінювати їх площу та відносити їх до певного класу з високою точністю. При цьому результати сегментації контурів будівель на даних космознімків, що датуються до катастрофічної події становить 97%, а результати класифікації сегментованих контурів будівель на 2 бінарні класи (пошкоджені та не пошкоджені) становлять 89%.

Результати дослідження показали, що нейронна мережа, навчена на репрезентативному наборі зразків, може розрізняти як пошкоджені, так і непошкоджені типи будівельної інфраструктури. При цьому була досягнута мета оцінки наслідків руйнування інфраструктури будівель внаслідок сейсмічних подій за допомогою методів машинного навчання. Таким чином кількість зруйнованих будівель становить 406, від загальної кількості будівель місцевості 966 на території площею 5.98 km<sup>2</sup> міста Амастріче країни Італія.

Результати класифікації можуть бути використані для представлення наборів даних для отримання статусу пошкоджень та значно прискорити реагування рятувальних служб. Навчені моделі можна використовувати в майбутньому, оскільки вони працюють під вільною ліцензією.

Подальші дослідження повинні зосередитися на збільшенні розміру вибірки для підвищення точності результатів моделі та способів її оцінки.

Дана робота пройшла апробацію на XVI International Scientific Conference “Monitoring of Geological Processes and Ecological Condition of the Environment” 15–18 November 2022, Kyiv, Ukraine. Сертифікат, що підтверджує участь та програма конференції представлена у додатках Г та Г відповідно.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

Valentijn, T., Margutti, J., van den Homberg, M., Laaksonen, J. Multi-Hazard and Spatial Transferability of a CNN for Automated Building Damage Assessment. *Remote Sens.*, 2022, p. 2-5.

Tong, X.; Hong, Z.; Liu, S.; Zhang, X.; Xie, H.; Li, Z.; Yang, S.; Wang, W.; Bao, F. Building-damage detection using pre-and post-seismic high-resolution satellite stereo imagery: A case study of the May 2008 Wenchuan earthquake. *ISPRS J. Photogramm. Remote Sens.* 2012, 68, 13–27.

Rastiveis, H.; Samadzadegan, F.; Reinartz, P. A fuzzy decision making system for building damage map creation using high resolution satellite imagery. *Nat. Hazards Earth Syst. Sci.* 2013, 13, 455

François Chollet *Deep Learning with Python* United States of America 2018, 386p.

Menderes, A.; Erener, A.; Sarp, G. Automatic detection of damaged buildings after earthquake hazard by using remote sensing and information technologies. *Procedia Earth Planet. Sci.* 2015, 15, 257–262

Voigt, S., Kemper, T., Riedlinger, T., Kiefl, R., Scholte, K., Mehl, H., 2007. Satellite image analysis for disaster and crisis-management support. *IEEE Transactions on Geoscience and Remote Sensing* 45(6), 1520–1528.

Світличний О.О., Плотницький С.В. Основи геоінформатики: Навчальний посібник / За заг. ред. О.О. Світличного. - Суми: ВТД «Університетська книга», 2006. - 295 с. ISBN 966-680-234-1

Min Ji, Lanfa Liu Identifying Collapsed Buildings Using Post-Earthquake Satellite Imagery and Convolutional Neural Networks: A Case Study of the 2010 Haiti Earthquake *Remote Sens.* 2018.

- Saito, K.; Spence, R. Rapid damage mapping using post-earthquake satellite images. In Proceedings of the 2004 IEEE International Geoscience and Remote Sensing Symposium, Anchorage, AK, USA, 20–24 September 2004; Volume 4, pp. 2272–2275
- Rathje, E.M.; Crawford, M.; Woo, K.; Neuenschwander, A. Damage patterns from satellite images of the 2003 Bam, Iran, earthquake. *Earthq. Spectra* 2005, 21, 295–307
- Turker, M.; Sumer, E. Building-based damage detection due to earthquake using the watershed segmentation of the post-event aerial images. *Int. J. Remote Sens.* 2008, 29, 3073–3089
- Miura, Hiroyuki et al. Building Damage Assessment Using High-Resolution Satellite SAR Images of the 2010 Haiti Earthquake *Earthquake Spectra* (2016), 32(1): 591 <https://doi.org/10.1193/033014EQS042M>
- Li, X., Yang, W., Ao, T., Li, H., Chen, W., 2011. An improved approach of information extraction for earthquake-damaged buildings using high-resolution imagery. *Journal of Earthquake and Tsunami* 5, 389–399.
- Zhai, W.; Huang, C. Fast building damage mapping using a single post-earthquake PolSAR image: A case study of the 2010 Yushu earthquake. *Earth Planets Space* 2016, 68, 86
- Wu, F.; Gong, L.; Wang, C.; Member, S.; Zhang, H. Signature analysis of building damage with TerraSAR-X new staring spotLight mode data. *IEEE Geosci. Remote Sens. Lett.* 2016, 13, 1696–1700.
- Rastiveis, H.; Eslamizade, F.; Hosseini-Zirdoo, E. Building damage assessment after earthquake using post-event LiDAR data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* 2015, 40, 595

Theodoridis, S.; Koutroumbas, K. Pattern Recognition, 4th ed.; Academic Press: New York, NY, USA, 2009

Peijun Li, Haiqing Xu, Jiancong Guo. Urban building damage detection from very high resolution imagery using OCSVM and spatial features. July 2010, International Journal of Remote Sensing 31(13):3393-3409. DOI:10.1080/01431161003727705

Xue Wang, Peijun Li. Extraction of earthquake-induced collapsed buildings using Very High Resolution imagery and airborne Lidar data. April 2015, International Journal of Remote Sensing 36(8):2163-2183. DOI:10.1080/01431161.2015.1034890

James Bialas, Thomas Oommen, Umaa Rebbapragada, Eugene Levin. Object-based classification of earthquake damage from high-resolution optical imagery using machine learning. September 2016, Journal of Applied Remote Sensing 10(3):036025. DOI:10.1117/1.JRS.10.036025

Ji M., Liu L., Buchroithner M. Identifying Collapsed Buildings Using Post-Earthquake Satellite Imagery and Convolutional Neural Networks: A Case Study of the 2010 Haiti Earthquake. October 2018, Remote Sensing 10(11):1689. DOI:10.3390/rs10111689

Zheng, Z.; Zhong, Y.; Wang, J.; Ma, A.; Zhang, L. Building damage assessment for rapid disaster response with a deep object-based semantic change detection framework: From natural disasters to man-made disasters. Remote Sens. Environ. 2021, 265, 112636

Ma, H., Liu, Y., Ren, Y., Wang, D., Yu, L., & Yu, J. (2020). Improved CNN Classification Method for Groups of Buildings Damaged by Earthquake, Based on High Resolution Remote Sensing Images. Remote Sensing, 12(2), 260. doi:10.3390/rs12020260

Valentijn, T., Margutti, J., van den Homberg, M., Laaksonen, J. Multi-Hazard and Spatial Transferability of a CNN for Automated Building Damage Assessment. *Remote Sens.*, 2022, p. 2-5.

Iglovikov V., Mushinskiy S., Osin V. Satellite Imagery Feature Detection using Deep Convolutional Neural Network: A Kaggle Competition arXiv:1706.06169v1 [cs.CV] 19 Jun 2017

Susu Xu, Joshua Dimasaka, David J. Wald, Hae Young Noh. Seismic multi-hazard and impact estimation via causal inference from satellite imagery. *Journal: nature communications*, 2022-12-17, Vol 13, Issue 1, pp. 1-13

Cappucci S. , Buffarini G., Giordano L., Hailemikael S. Local Geology and Seismic-Induced Damages: The Case of Amatrice (Central Italy). October 2020, Conference: Computational Science and Its Applications – ICCSA 2020. DOI:10.1007/978-3-030-58802-1\_68

Rossi, A. The 2016–2017 earthquake sequence in Central Italy: macroseismic survey and damage scenario through the EMS-98 intensity assessment. *Bull. Earthq. Eng.* 17(5), 2407–2431 (2019). <https://doi.org/10.1007/s10518-019-00556-w>

Amatrice, Norcia, Visso Seismic Sequence, INGV. Accessed 28 Mar 2020. Режим доступу: <http://terremoti.ingv.it/it/ultimi-eventi/1023-sequenza-sismica-in-italia-centrale-aggiornamenti.html>

Italian Strong Motion Network (Rete Accelerometrica Nazionale - RAN). <http://www.protezionecivile.gov.it/en/risk-activities/seismic-risk/activities/italian-strong-motion-network>. Accessed 04 May 2020

Fiorentino G., Angelo Fort, Enrico Pagano, Fabio Sabetta. Damage patterns in the town of Amatrice after August 24th 2016 Central Italy earthquakes. March 2018, *Bulletin of Earthquake Engineering* 16(1). DOI:10.1007/s10518-017-0254-z

Pizzi, A., Galadini, F. Pre-existing cross-structures and active fault segmentation in the northern-central Apennines (Italy). *Tectonophysics* 476, 304–319 (2009)

Regione Lazio, Microzonazione Sismica Livello 3 - Ordinanza n. 24/2017.  
[http://www.regione.lazio.it/prl\\_ambiente/?vw=contenutidetail&id=238](http://www.regione.lazio.it/prl_ambiente/?vw=contenutidetail&id=238)

Vapnik V. and Cortes C., «Support-Vector Networks», *Machine Learning* 20, no. 3 (1995): 273–297.

Nwankpa, C., Ijomah, W., Gachagan, A. and Marshall, S. Activation Functions: Comparison of trends in Practice and Research for Deep Learning. March 18, 2021, *Open Journal of Statistics*, Vol.11 No.2,

Shiv Ram Dubey, Satish Kumar Singh, Bidyut Baran Chaudhuri. Activation Functions in Deep Learning: A comprehensive Survey and Benchmark. July 2022, *Neurocomputing* 503(11). DOI:10.1016/j.neucom.2022.06.111

Li, Y.; Hao, Z.; Lei, H. Overview of convolutional neural networks. *J. Comput. Appl.* 2016, 36, 2508–2515.

Kattenborn T., Leitloff J., Schiefer F., Hinz S. Review on Convolutional Neural Networks (CNN) in Vegetation Remote Sensing. March 2021, *ISPRS Journal of Photogrammetry and Remote Sensing* 173(2):24-49. DOI:10.1016/j.isprsjprs.2020.12.010

Convolutional Neural Networks (CNNs/ConvNets). Режим доступа: <https://cs231n.github.io/convolutional-networks/>

Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research*, Department of Computer Science University of Toronto, Toronto, Ontario, M5S 3G4, Canada, 2014.

Hasani M., Khotanlou H. An Empirical Study on Position of the Batch Normalization Layer in Convolutional Neural Networks. December 2019, Conference: 2019 5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS). DOI:10.1109/ICSPIS48872.2019.9066113

Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 2015

Ragab M.G., Abdulkadir S.A., NorShakirah A., Al-Tashi Q. A Novel One-Dimensional CNN with Exponential Adaptive Gradients for Air Pollution Index Prediction. December 2020, Sustainability 12(23):10090. DOI:10.3390/su122310090

Gao, Z.; Zhang, Y.; Li, Y. Extracting features from infrared images using convolutional neural networks and transfer learning. Infrared Phys. Technol. 2020, 105, 103237.

Bottou L. Large-Scale Machine Learning with Stochastic Gradient Descent. 2010, NEC Labs America, Princeton NJ 08542, USA

Boughorbel S., Jarray F., El-Anbari M. Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric PLOS ONE [https://doi.org/10.1371/journal.pone.0177678 ], 2017

Kulkarni A., Chong D., Batarseh Feras A.. Foundations of data imbalance and solutions for a data democracy. Data Democracy, 2020. ISBN 978-0-12-818366-3. DOI https://doi.org/10.1016/C2018-0-04003-7

Rahman A., Wang Y., Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation Department of Computer Science, University of Manitoba, Canada, 2016

Lecun Y., Bottou L., Bengio Y., Haffner P. Gradient-based Learning Applied To Document Recognition, Procc. of the IEEE, 1998

Krizhevsky A., Sutskever I., Hinton G., ImageNet Classification with Deep Convolutional Neural Networks, 2012

Mingxing Tan, Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. International Conference on Machine Learning, 2019 DOI <https://doi.org/10.48550/arXiv.1905.11946>

Kravchenko, Y.G., Tishaieva, A.M., Perkhaliuk, R.I. The Convolution Neural Network for automatic objects detection in Earth satellite imagery. Conference Proceedings, 17th International Conference on Geoinformatics - Theoretical and Applied Aspects, May 2018, V. 2018, p.1-5.

DeepLearnongAI. Режим доступу: <https://www.deeplearning.ai/>

Keras official documentation [Electronic Sources]. Retrieved from: <https://keras.io/>  
Pytorch [Электроний ресурс]. – Режим доступу: <https://pytorch.org/>

Tensorflow official web site [Электроний ресурс]. – Режим доступу: [www.tensorflow.org](http://www.tensorflow.org)

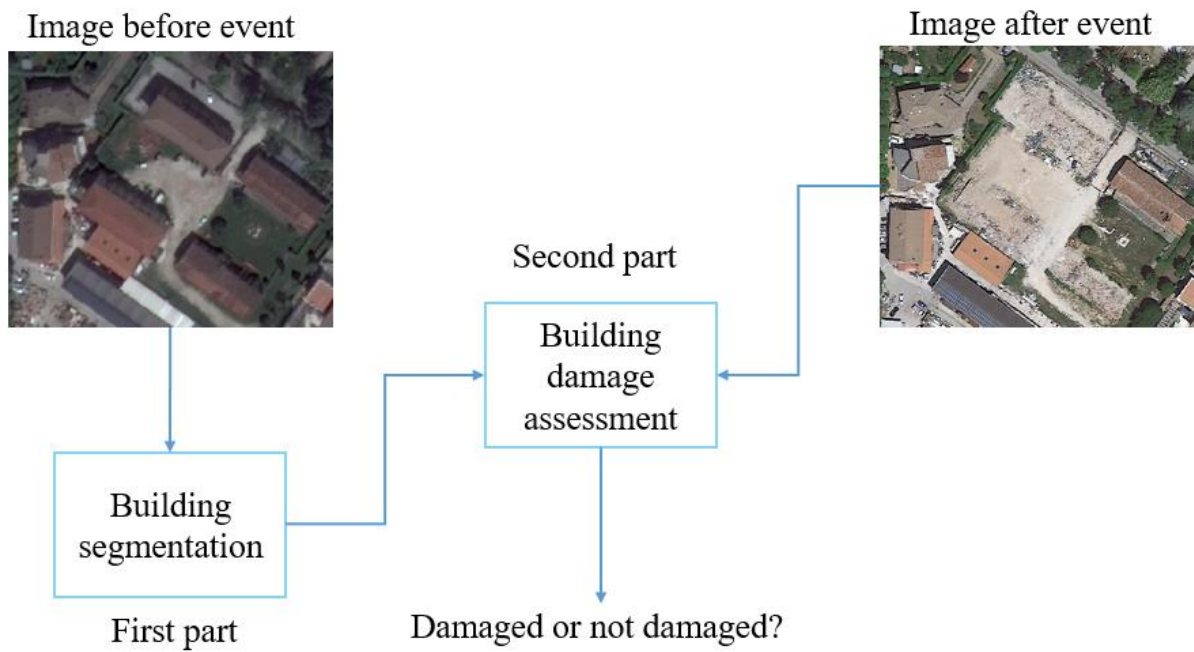
Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1). doi:10.1186/s40537-019-0197-0

Okhrimchuk, R., Tishaiev, I., Zatserkovnyi, V. Anticlines Prediction Using Deep Learning. Conference Proceedings, NSG2020 26th European Meeting of Environmental and Engineering Geophysics, Dec 2020, Volume 2020, p.1-5.

Brudko K., Okhrimchuk R., Demidov V.. Automatic Recognition and Damage Evaluation of Building Infrastructure in Seismic Active Zones using Machine Learning. Conference Proceedings, 16th International Conference Monitoring of Geological Processes and Ecological Condition of the Environment, Nov 2022, Volume 2022, p.1 – 5. DOI: <https://doi.org/10.3997/2214-4609.2022580199>

## ДОДАТОК А

### Data Flow



## ДОДАТОК Б

### Код для створення моделей

```

from keras.models import *
from keras.layers import Dropout
from keras.optimizers import *
import keras

from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau

config = tf.ConfigProto(device_count={'GPU': 8, 'CPU': 32})
sess = tf.Session(config=config)
keras.backend.set_session(sess)

from tensorflow import *

def conv_block(tensor, nfilters, size=3, padding='same',
initializer="he_normal"):
    x = Conv2D(filters=nfilters, kernel_size=(size, size),
padding=padding, kernel_initializer=initializer)(tensor)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)
    x = Conv2D(filters=nfilters, kernel_size=(size, size),
padding=padding, kernel_initializer=initializer)(x)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)
    return x

def deconv_block(tensor, residual, nfilters, size=3, padding='same',
strides=(2, 2)):
    y = Conv2DTranspose(nfilters, kernel_size=(size, size),
strides=strides, padding=padding)(tensor)
    y = concatenate([y, residual], axis=3)
    y = conv_block(y, nfilters)
    return y

def Unet(img_height, img_width, nclasses, filters=64):
    # down
    input_layer = Input(shape=(img_height, img_width, 8),
name='image_input')
    conv1 = conv_block(input_layer, nfilters=filters)
    conv1_out = MaxPooling2D(pool_size=(2, 2))(conv1)
    conv2 = conv_block(conv1_out, nfilters=filters * 2)
    conv2_out = MaxPooling2D(pool_size=(2, 2))(conv2)
    conv3 = conv_block(conv2_out, nfilters=filters * 4)
    conv3_out = MaxPooling2D(pool_size=(2, 2))(conv3)
    conv4 = conv_block(conv3_out, nfilters=filters * 8)
    conv4_out = MaxPooling2D(pool_size=(2, 2))(conv4)
    conv4_out = Dropout(0.5)(conv4_out)
    conv5 = conv_block(conv4_out, nfilters=filters * 16)

```

```

conv5 = Dropout(0.5)(conv5)
# up
deconv6 = deconv_block(conv5, residual=conv4, nfilters=filters * 8)
deconv6 = Dropout(0.5)(deconv6)
deconv7 = deconv_block(deconv6, residual=conv3, nfilters=filters * 4)
deconv7 = Dropout(0.5)(deconv7)
deconv8 = deconv_block(deconv7, residual=conv2, nfilters=filters * 2)
deconv9 = deconv_block(deconv8, residual=conv1, nfilters=filters)
# output
output_layer = Conv2D(filters=nclasses, kernel_size=(1, 1))(deconv9)
output_layer = BatchNormalization()(output_layer)
output_layer = Activation('sigmoid')(output_layer)

model = Model(inputs=input_layer, outputs=output_layer, name='Unet')
return model

model = Unet(128, 128, nclasses=1, filters=64)

model.compile(loss='binary_crossentropy',
              optimizer=keras.optimizers.Adam(lr=0.001, beta_1=0.9,
beta_2=0.999,
              epsilon=None, decay=0.0,
amsgrad=False),
              metrics=[tf.keras.metrics.MeanIoU])

model.summary()
checkpoint = ModelCheckpoint(r"../UNET/MODELS/UNET_Model.h5",
monitor = 'val_loss',
verbose = 1,
save_best_only = True,
save_weights_only = False,
mode = 'auto')

reduce = ReduceLRonPlateau(monitor='val_loss', factor=0.1, patience=10,
verbose=1, mode='auto', min_delta=0.001,
                          cooldown=0, min_lr=0.00001)
history = model.fit(image_train, mask_train, batch_size=8,
callbacks=[checkpoint, reduce], epochs=100000,
            validation_data=(image_val, mask_val), shuffle=True,
verbose=1)

```

## ДОДАТОК В

### Код для аугментації зображень

```

import os
import numpy as np
from osgeo import gdal_array

# Block imagies
patterns = []

path_pattern_class_1 =
'/home/ai/Roma/Projects/Central_Italy_Earthquake_128x128'

first_class = os.listdir(path_pattern_class_1)

for samle in first_class:
    imag = gdal_array.LoadFile(path_pattern_class_1 + "/" + samle)
    patterns.append(np.array(imag))

    rotated_90 = np.rot90(imag, axes=(1, 2))
    patterns.append(np.array(rotated_90))

    rotated_180 = np.rot90(rotated_90, axes=(1, 2))
    patterns.append(np.array(rotated_180))

    rotated_270 = np.rot90(rotated_180, axes=(1, 2))
    patterns.append(np.array(rotated_270))

    diagonal_reverse = np.fliplr(imag)
    patterns.append(np.array(diagonal_reverse))

rotated_90 = rotated_180 = rotated_270 = diagonal_reverse = imag = samle =
first_class = path_pattern_class_1 = None

patterns = np.array(patterns)
patterns = np.transpose(patterns, (0, 2, 3, 1))

print(patterns.shape)

# Block masks

labels = []

path_pattern_mask_1 =
'/home/ai/Katya/Projects/Central_Italy_Earthquake_128x128'

first_mask = os.listdir(path_pattern_mask_1)

for samle in first_mask:
    imag = gdal_array.LoadFile(path_pattern_mask_1 + "/" + samle)

    labels.append(np.array(imag))

```

```

rotated_90 = np.rot90(imag, axes=(0, 1))
labels.append(np.array(rotated_90))

rotated_180 = np.rot90(rotated_90, axes=(0, 1))
labels.append(np.array(rotated_180))

rotated_270 = np.rot90(rotated_180, axes=(0, 1))
labels.append(np.array(rotated_270))

diagonal_reverse = np.fliplr(imag)
labels.append(np.array(diagonal_reverse))

rotated_90 = rotated_180 = rotated_270 = diagonal_reverse = imag = samle =
first_mask = path_pattern_mask_1 = None
labels = np.array(labels)

print("Augmantation for masks has been successfully")
print("Number of samples have been increased to:")
print(labels.shape)
print(np.unique(labels))

labels = np.expand_dims(labels, 3)

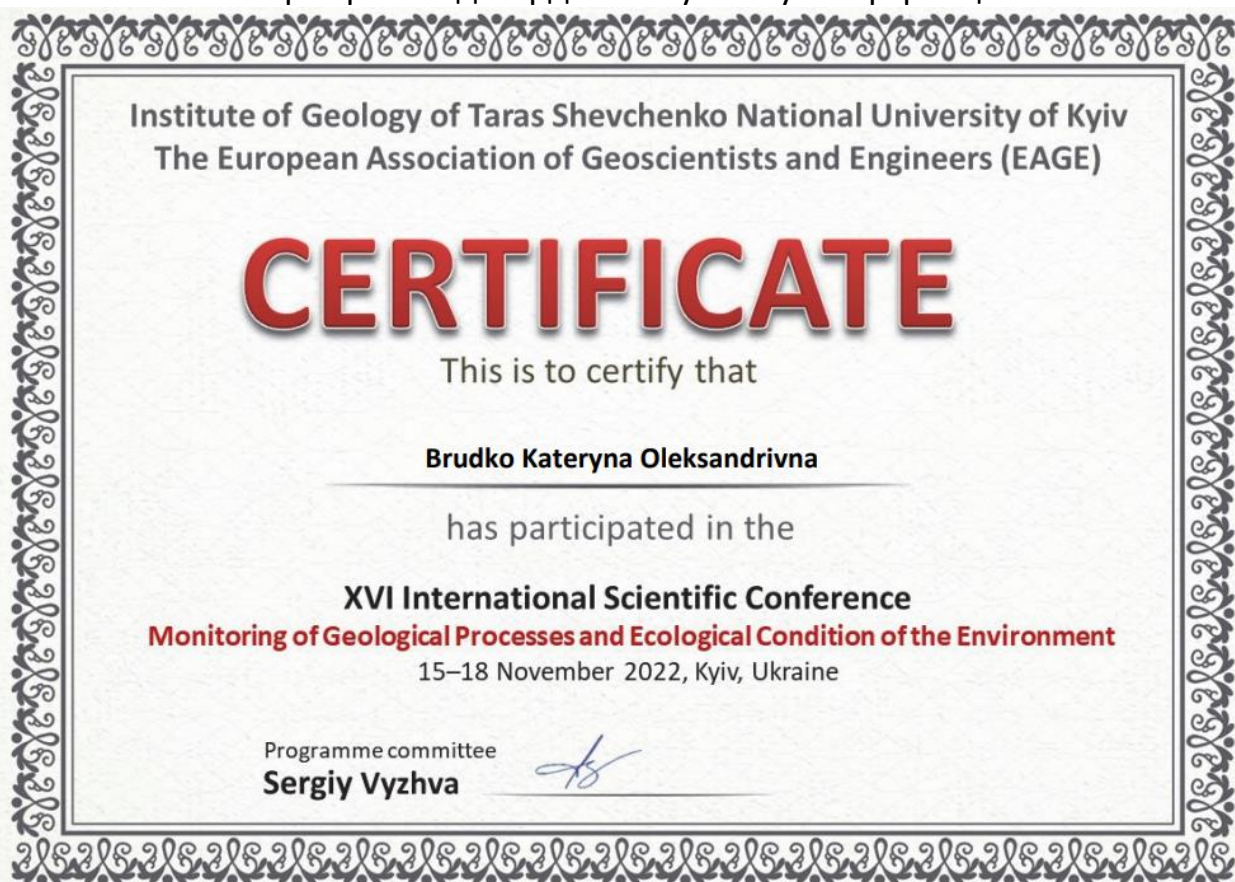
labels.shape
print(labels.shape)

data_length = patterns.shape[0]
(image_train, image_val) = patterns[(int)(0.1 * data_length):],
patterns[:(int)(0.1 * data_length)]
(mask_train, mask_val) = labels[(int)(0.1 * data_length):],
labels[:(int)(0.1 * data_length)]

```

**ДОДАТОК Г**

Сертифікат підтвердження участі у конференції



## ДОДАТОК Г

### Програма конференції

|       |  |
|-------|--|
| 14.00 | <p><b>Mon22-168</b></p> <p><b>Land use transitions in Lithuanian protected areas between 1990-2018</b></p> <p><i>*M. Inácio, M. Lapelê, M. Kallinauskas (Environmental Management Center, Mykolas Romeris University), A. Jasiniavičiūtė (Landscape Management division in State Service for Protected areas under the Ministry of Environment), P. Pereira (Environmental Management Center, Mykolas Romeris University)</i></p>  |
| 14.15 | <p><b>Mon22-180</b></p> <p><b>Using Local Indicators of Spatial Association (LISA) to assess spatial relation between urban parks' structure and users' activities in Vilnius (Lithuania)</b></p> <p><i>L. V. Pinto (CERNAS, Polytechnic Institute of Coimbra, and Environmental Management Center, Mykolas Romeris University), C. S. S. Ferreira (Department of Physical Geography, Stockholm University, and Bolin Centre for Climate Research), P. Pereira (Environmental Management Center, Mykolas Romeris University)</i></p> |
| 14.30 | <p><b>Mon22-199</b></p> <p><b>Automatic recognition and damage evaluation of building infrastructure in seismic active zones using machine learning</b></p> <p><i>*K. Brudko, R. Okhrimchuk, V. Demidov (Taras Shevchenko National University of Kyiv)</i></p>   |
| 14.45 | <p><b>Mon22-213</b></p> <p><b>Semantic segmentation of Western Crimean coastline for high-resolution satellite images using deep learning based on U-Net architecture</b></p> <p><i>*R. Okhrimchuk, V. Demidov, K. Brudko (Taras Shevchenko National University of Kyiv)</i></p>   |
| 15.00 | <p><b>Mon22-244</b></p> <p><b>Geoinformational analysis of Oleshky Sands desertification threats</b></p> <p><i>*S. Moku, D. Llashenko (Taras Shevchenko National University of Kyiv)</i></p>   |
| 15.15 | <p><b>Mon22-252</b></p> <p><b>Geoinformatics modeling of hydrological parameters on the example of the Prut river basin within the Ivano-Frankivsk region</b></p> <p><i>I. Chepurnyi, Ya. Fedyk, T. Chepurna, *V. Rushchak (Ivano-Frankivsk National Technical University of Oil and Gas)</i></p>  |
| 15.30 | Break  |
| 15.45 | <p><b>Mon22-253</b></p> <p><b>Geospatial database of karst processes manifestations for the territory of the administrative region as the basis of the geoportal</b></p> <p><i>*B. Karpinskyi, I. Chepurnyi, T. Chepurna (Ivano-Frankivsk National Technical University of Oil and Gas)</i></p>  |
| 16.00 | <p><b>Mon22-267</b></p> <p><b>Data-driven preprocessing of gravity data in oilfield GIS monitoring system in Kazakhstan</b></p> <p><i>*Y. I. Dubovenko (Institute of Geophysics National Academy of Sciences of Ukraine), A. B. Nazirova (Almaty University of Power Engineering and Telecommunications named after Gumarbek Daukeyev, Almaty, Kazakhstan), F. N. Abdoldina (International Information Technology University, Almaty, Kazakhstan)</i></p>  |
|       | Poster   |
| 16.20 | <p><b>Mon22-056</b></p> <p><b>The use of GIS and remote sensing data in determining the condition of green areas in Kyiv</b></p> <p><i>*N. Korohoda, O. Halahan, O. Kovtoniuk (Taras Shevchenko National University of Kyiv)</i></p>   |
| 16.30 | <p><b>Mon22-104</b></p> <p><b>Optimization of data sets and improvement of methods for calculating the soil brightness index</b></p> <p><i>*O. Hordlienko, O. Trofymchuk, Y. Anpilova (Institute of Telecommunications and Global Information Space of NASU), Y. Butenko (Institute of Water Problems and Land Reclamation),</i></p>   |



XVI International Scientific Conference "Monitoring of Geological Processes  
and Ecological Condition of the Environment"  
15–18 November 2022, Kyiv, Ukraine