

УДК 004.75

DOI 10.30837/bi.2025.2(103).04

К. В. Сільванович¹, О. Є. Гриньова¹, Л. Е. Чала¹, С. Г. Удовенко²¹ХНУРЕ, м. Харків, Україна, kristina.silvanovych@nure.ua, ORCID iD: 0009-0008-3723-1124¹ХНУРЕ, м. Харків, Україна, olena.hrynova@nure.ua, ORCID iD: 0000-0002-3367-8067¹ХНУРЕ, м. Харків, Україна, larysa.chala@nure.ua, ORCID iD: 0000-0002-9890-4790²ХНЕУ ім. С. Кузнеця, м. Харків, Україна, serhiy.udovenko@hneu.net, ORCID iD: 0000-0001-5945-8647

НЕЙРОМЕРЕЖЕВІ ТЕХНОЛОГІЇ МОНІТОРИНГУ ТА АНАЛІЗУ РУЙНІВНИХ ПОШКОДЖЕНЬ АГРАРНИХ ДІЛЯНОК

Здійснено аналіз існуючих інтелектуальних технологій виявлення та класифікації руйнівних пошкоджень аграрних ділянок. Розроблено моделі класифікації аграрних ділянок за ступенем пошкоджень та сегментації зображень пошкоджених ділянок з використанням нейромережевої архітектури U-Net. Запропоновано прогнозну модель для визначення термінів відновлення пошкоджених ділянок з гібридним використанням архітектур TFT і LSTM та аналізу даних про стан ґрунтів і кліматичних факторів. Реалізовано інтеграцію розроблених моделей для створення інтелектуальної системи класифікації пошкоджень, сегментації уражених ділянок та прогнозування термінів рекультивациі. Для реалізації системи були обрані: платформа WPF для створення зрозумілого та сучасного інтерфейсу, ONNX Runtime для ефективної роботи моделей штучного інтелекту, а також використання CSV-файлів для структурованого зберігання й обміну даними. Результати тестування підтверджують працездатність запропонованого підходу.

НЕЙРОМЕРЕЖЕВІ ТЕХНОЛОГІЇ, МОНІТОРИНГ ПОШКОДЖЕНЬ, АНАЛІЗ ЗОБРАЖЕНЬ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, ПРОГНОЗУВАННЯ ЧАСУ ВІДНОВЛЕННЯ

K.V. Silvanovych, O.E. Hrynova, L.E. Chala, S.G. Udovenko. Neural network technologies for monitoring and analysis of destructive damage to agricultural plots. An analysis of existing intelligent technologies for detecting and classifying destructive damage to agricultural plots was carried out. Models for classifying agricultural plots by the degree of damage and segmenting images of damaged plots using the U-Net neural network architecture were developed. A predictive model was proposed for determining the terms of restoration of damaged plots by hybridising TFT and LSTM architectures, along with an analysis of data on soil states and climatic factors. The integration of the developed models was implemented to create an intelligent system for classifying damage, segmenting affected plots, and predicting the terms of reclamation. The following components were selected for the implementation of the system: the WPF platform for creating a clear and modern interface, ONNX Runtime for the efficient operation of artificial intelligence models, and the use of CSV files for structured data storage and exchange. The testing results confirm the operability of the proposed approach.

NEURAL NETWORK TECHNOLOGIES, DAMAGE MONITORING, IMAGE ANALYSIS, CONVOLUTIONAL NEURAL NETWORKS, RECOVERY TIME PREDICTION

Вступ

Внаслідок збройної агресії, розпочатої військовими силами Російської Федерації, екосистема України зазнала суттєвих руйнувань. Це, зокрема, призвело до порушення нормального стану сільськогосподарських земельних ділянок та ґрунтових ресурсів [1]. Виникла гостра потреба в здійсненні оперативного моніторингу та аналізу пошкоджень агроресурсів.

Сільське господарство завжди було основою продовольчої безпеки держави, а стан ґрунтових ресурсів безпосередньо впливає на ефективність аграрного виробництва. Проте останні десятиліття виявили значні природні фактори деградації ґрунтів, тобто процеси погіршення їхніх властивостей, спричинені природними умовами, такими як ерозія, виснаження поживних речовин, зміни водного балансу, а також вплив кліматичних коливань. Через це ґрунти стають більш вразливими до антропогенних факторів, зокрема інтенсивного землекористування.

Новим викликом у часи воєнних дій стає поєднання цих природних проблем із масштабними антропогенними ушкодженнями. При цьому на багатьох територіях виникають масштабні екологічні

проблеми – руйнування родючого шару, забруднення важкими металами, накопичення уламків техніки й вибухових речовин. Це призводить до втрати продуктивності угідь, деградації агроландшафтів та зростання ризику для здоров'я населення [2].

Україна стикається з однією з найбільших криз забруднення земель у світі: станом на грудень 2024 року, понад 138 000 км² земель (близько 20% території) забруднено мінами та вибухівкою, що блокує доступ до 14 000 км² сільгоспугідь. Це загрожує продовольчій безпеці, оскільки агросектор становить близько 70% експорту країни.

Відновлення пошкоджених земель потребує попередньої оцінки їхнього стану, придатності до ведення сільського господарства, необхідності рекультивациі та відповідності рельєфу й просторової організації умовам ефективного господарювання. Традиційні методи, такі як виїзні обстеження, відбір та лабораторний аналіз зразків, забезпечують достовірні результати, але є надто затратними за часом і ресурсами, а подекуди й неможливими через вибухонебезпечність території. Вони не дозволяють охопити велику кількість пошкоджених аграрних ділянок. Саме тому необхідні нові,

масштабовані та швидкодіючі підходи до моніторингу та аналізу пошкоджень екосистеми.

Сучасні цифрові технології, зокрема штучний інтелект, комп'ютерний зір та алгоритми глибинного навчання, відкривають нові можливості у вирішенні цієї проблеми [3]. Інтелектуальні системи, за наявності супутникових та аерофотознімків, здатні автоматично виявляти пошкодження, визначати їхні межі, класифікувати тип уражень (механічні, хімічні, термічні), ступінь пошкоджень та прогнозувати реальні терміни відновлення родючості. Зокрема, такі системи можуть відокремити ділянку, де зафіксоване забруднення, від території з повністю зруйнованим верхнім шаром ґрунту, і на основі цього сформувані карту уражень. Отримані внаслідок цього результати дозволяють ухвалювати зважені рішення (з урахуванням логістики та планування) щодо необхідності термінів рекультивациі та можливості відновлення сільськогосподарських робіт.

Розробка інтегрованих систем, здатних обробляти великі обсяги даних, відкриває новий рівень ефективності у відновленні сільськогосподарських земель. Подібні системи надають аграріям, науковцям і органам влади доступ до об'єктивної та оперативної інформації, знижуючи залежність від трудомістких обстежень і людського фактору.

Важливо враховувати, що самі зображення пошкоджених ділянок є лише вхідними даними. Для їх перетворення на корисну інформацію необхідна комплексна обробка за допомогою нормалізації, фільтрації шумів, виділення ключових ознак, порівняння з еталонними станами тощо. Виконати це вручну практично неможливо, адже коли обсяги даних зазвичай обчислюються десятками тисяч знімків, кожен із них є частиною часової серії, що відображає динаміку змін стану аналізованої території. Саме тому вирішальну роль відіграють автоматизовані інтелектуальні системи, здатні навчатися на прикладах і узагальнювати нову інформацію.

Перевагою такого підходу є не лише швидкість, а й можливість переходу від фіксації проблем до плану дій. Системи на основі штучного інтелекту здатні враховувати тип пошкоджень, кліматичні та сезонні фактори, що дає змогу формувати більш реалістичні та ефективні стратегії аграрного відновлення.

Метою цієї статті є розроблення та дослідження технологій моніторингу та аналізу пошкоджень агро-екосистеми, що дозволяють в подальшому реалізувати інтелектуальну систему класифікації пошкоджень, сегментації уражених ділянок та прогнозування термінів рекультивациі. До завдань такої системи слід віднести можливість завантаження знімків пошкоджених ділянок, отримання карти уражень, розрахунку числових показників площі та типу пошкоджень, а також формування прогнозів щодо термінів відновлення.

Впровадження подібних технологій сприятиме підвищенню ефективності моніторингу та відбудови

пошкоджених аграрних територій. Це має бути певним стратегічним внеском у продовольчу безпеку, післявоєнне відновлення та збереження природного потенціалу країни (ЦСР 2 «Подолання голоду» та ЦСР 15 «Збереження екосистем суші»).

Відповідно до поставленої мети, необхідно вирішити наступні завдання:

- аналіз існуючих інтелектуальних технологій виявлення та класифікації пошкоджень аграрних ділянок;
- розроблення технології виявлення пошкоджених ділянок з використанням супутникових знімків Sentinel-2 та нейромережових згорткових моделей;
- розроблення моделей класифікації аграрних ділянок за ступенем пошкоджень та сегментації зображень пошкоджених ділянок з використанням нейромережових технологій;
- розроблення прогновної моделі для визначення термінів відновлення пошкоджених ділянок з використанням алгоритмів машинного навчання та аналізу історичних даних про стан ґрунтів та кліматичних факторів;
- інтеграція розроблених моделей для створення інтелектуальної системи класифікації пошкоджень, сегментації уражених ділянок та прогнозування термінів рекультивациі.
- експериментальне дослідження запропонованих моделей та технологій.

1. Сучасні технології виявлення та класифікації пошкоджених аграрних ділянок

Розвиток цифрових технологій суттєво змінив підходи до ведення сільського господарства.

Найбільшого поширення набули методи комп'ютерного зору та глибинного навчання, зокрема згорткові нейронні мережі (CNN) [4]. Вони спроможні автоматично аналізувати супутникові та аерофотознімки, знаходити пошкоджені ділянки, сегментувати уражені зони та навіть визначати характер деградації ґрунтів. Наприклад, впровадження архітектур типу U-Net показало високу ефективність для задач сегментації сільськогосподарських територій, що дозволяють отримувати точні карти уражень.

Окрему нішу займають моделі, оптимізовані для здійснення часового прогнозування. Рекурентні нейронні мережі (RNN), модифікації LSTM (Long Short-Term Memory) та нейромережі на основі трансформерів (Temporal Fusion Transformers) можуть аналізувати не лише зображення, а й часові ряди кліматичних даних, рівень вологості ґрунтів, кількість опадів чи сезонні цикли. Це дає змогу оцінювати не тільки масштаби пошкоджень, а й орієнтовні терміни відновлення родючості.

Технологія OneSoil (Європа/США) використовує супутникові дані для моніторингу (NDVI-аналіз, виявлення стресу рослин на базі Sentinel-2/Landsat) та

зонування полів, але без фокусу на рекультивациі [5]. Функціональність – формування карт врожайності та рекомендацій по добривам (з точністю сегментації ~0.75–0.85 IoU). Недолік – відсутність фокусу на воєнних пошкодженнях (не сегментує вирви чи метали).

Технологія CropIn (Індія) використовує нейромережеві моделі (CNN+RNN) для прогнозування врожаю та моніторингу шкідників з дронів/супутників, акцент на ланцюги постачань але менш адаптована до деградації [6]. Функціональність – здійснення ризик-аналізу та оптимізації посівів (з точністю моделей ~80–90%). Недолік – орієнтація на тропічний клімат, без сегментації деградації від вибухів.

Технологія Mine Action Project in Ukraine є діючим проектом з протимінної діяльності в Україні, що реалізується UNDP у співпраці з FAO (Продовольча та сільськогосподарська організація Об'єднаних Націй) та WFP (Всесвітня продовольча програма) для тестування ефективності розмінування з використанням штучного інтелекту (ШІ). Функціональність – виявлення та обробка зображень з дронів з використанням згорткових нейронних мереж (точність виявлення >95%). Ця технологія генерує також карти забруднення та пріоритизує зони пошкодження [7].

Для України такі технології матимуть особливе значення в умовах післявоєнної відбудови. Вони дозволять отримувати швидку та об'єктивну картину стану земель на великих територіях, що практично неможливо зробити традиційними методами. Завдяки цьому аграрії, науковці й державні структури отримають унікальний інструмент для прийняття рішень від визначення зон, придатних до посівів, до планування програм рекультивациі й довгострокових стратегій відновлення агросфери.

Слід відзначити, що переважна більшість сучасних технологій виявлення та класифікації пошкоджених аграрних ділянок базується на використанні штучних нейронних мереж (ШНМ). ШНМ є одним із провідних інструментів сучасного штучного інтелекту, здатним моделювати складні нелінійні залежності та виявляти приховані закономірності у великих масивах даних. Завдяки своїй універсальності ШНМ знайшли широке застосування у задачах комп'ютерного зору та прогнозування стану середовища [8].

У контексті дослідження пошкоджених сільськогосподарських земель нейронні мережі відіграють ключову роль у двох напрямках:

- комп'ютерний зір дозволяє аналізувати зображення із дронів або супутників для виявлення та класифікації пошкоджень. Тут основними інструментами виступають згорткові нейронні мережі (Convolutional Neural Networks, CNN), що спеціалізуються на роботі з даними, які мають просторову структуру (зображення, карти, відео);

- нейромережеве прогнозування дає оцінку довгострокових наслідків руйнувань та визначення

строків відновлення ґрунтів. Для цього можуть бути задіяні моделі роботи з часовими рядами – рекурентні мережі (Long Short-Term Memory, LSTM) та новітні архітектури трансформерів (зокрема Temporal Fusion Transformer-TFT) [9].

Особливістю CNN є здатність автоматично виявляти ознаки на зображеннях від простих (лінії, контури) до складних (форми вирв від вибухів, забруднені ділянки). Це робить їх надзвичайно ефективними у задачах класифікації та сегментації зображень сільськогосподарських угідь. Застосування CNN дозволяє створювати системи, які автоматично розрізняють «уражені» та «неуражені» території, а також деталізують межі пошкоджень. Базову архітектуру такої мережі наведено на рис. 1.

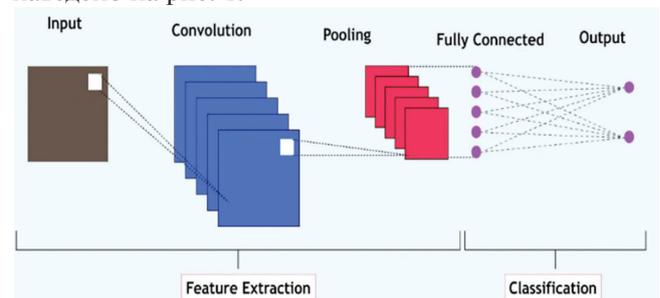


Рис. 1. Базова архітектура мережі CNN

Схематично CNN є послідовністю шарів, кожен з яких перетворює один активаційний об'єм в інший за допомогою функції, що диференціюється. Для організації згорткової нейронної мережі, яка здійснює вилучення ознак (feature extraction) та класифікацію (classification) уражених територій, застосовуються 3 основних шари: згортки (convolution), пулінгу (pooling), повнозв'язний шар (fully connected).

Виділення контурів (сегментація) пошкоджених ділянок сільськогосподарських земель може бути ефективно здійснене з застосуванням нейромережевої архітектури U-Net. Вона побудована за принципом симетричної структури «енкодер-декодер», де перша частина (енкодер) відповідає за вилучення ознак на різних рівнях, а друга (декодер) – за відновлення просторової структури та формування карти сегментації [10].

Архітектура U-Net відзначається тим, що поєднує багаторівневі ознаки завдяки так званим зв'язкам «skip connections», що дозволяє зберігати як деталі локального рівня, так і глобальний контекст. Це робить модель придатною для аналізу знімків середньої та високої роздільності, отриманих із дронів або супутників.

Позначимо вхідне зображення як $X \in R^{H \times W \times C}$, де H, W – висота та ширина зображення, C – кількість каналів.

Енкодер з L шарами (зазвичай $L = 3$), визначено за формулою:

$$E_l = f_l(E_{l-1}) \text{ для } l = 1, \dots, L, \quad (1)$$

де f_l – операції згортки, нормалізації та активації у l -му шарі, $E_0 = X$.

Декодер з використанням skip-зв'язків реалізується наступним чином:

$$D_l = g_l(D_{l+1}, E_l) \text{ для } l = L-1, \dots, 1, \quad (2)$$

де g_l – операції апсемплінгу, конкатенації та згортки.

Вихідна маска сегментації з використанням U-Net має наступний вигляд:

$$Y = \sigma(D_1), \quad (3)$$

де σ – функція активації softmax.

Знімки для аналізу можуть братися як із супутників Sentinel-2 (особливо корисною є смуга B4, яка відображає зміни у рослинності), так і з дронів, що забезпечує більшу деталізацію локальних пошкоджень. Базова архітектура CNN та архітектура U-Net застосовуються для автоматизації процесу оцінки пошкоджених земель. Вони дозволяють забезпечувати швидке та точне картографування зон ураження, що значно зменшує потребу у ручному аналізі великих масивів даних.

Для прогнозування термінів відновлення ґрунтів необхідно враховувати не лише просторові характеристики, але й часову динаміку їх змін. Для цього доцільно задіювати рекурентні нейронні мережі (Recurrent Neural Networks, RNN), здатні ефективно працювати з послідовними даними. Класичні RNN мають певні обмеження, пов'язані з ефектом «згасання градієнтів», що ускладнює їх навчання на довгих часових рядах. Для вирішення цієї проблеми зазвичай використовують архітектуру LSTM, яка завдяки спеціальним осередкам пам'яті здатна зберігати інформацію на великих часових інтервалах [11]. Моделі LSTM були створені для роботи з послідовними даними, що є критично важливим у завданнях прогнозування відновлення ґрунтів. Стан ґрунту залежить не лише від масштабу руйнувань, а й від динаміки кліматичних факторів, типу сільського господарства та часу. Мережа LSTM здатна враховувати довготривалі залежності, що робить її придатною як для короткострокових, так і для довгострокових прогнозів.

Часова динаміка LSTM визначається за формулою:

$$h_t = LSTM(F_{combined}, h_{t-1}), \quad (4)$$

де h_t – прихований стан для часу t .

До системи виявлення та класифікації пошкоджених аграрних ділянок доцільно доцільно інтегрувати й нейромережеву модель TFT, архітектура якої поєднує як статичні дані (тип ґрунту, вид зброї, географічне положення), так і динамічні (опади, температурні режими, сезонні цикли). Перевагою TFT є її інтерпретованість, адже ця модель не лише передбачає строк відновлення, а й визначає, які фактори найбільше вплинули на прогноз. Це особливо важливо у системах підтримки прийняття рішень для аграрного сектору.

В мережі TFT для уваги до важливих факторів використовується поточний прогноз часу відновлення:

$$\hat{y}_t = TFT(h_{t-1}, \text{StaticFeatures}), \quad (5)$$

де \hat{y}_t – прогноз часу відновлення для моменту t .

Наведений аналіз свідчить про перспективність комбінованого використання архітектури CNN + U-Net для задачі сегментації та архітектури CNN + LSTM + TFT для задачі прогнозування. В межах такого підходу LSTM аналізує часові ряди, пов'язані з кліматичними та сезонними змінами, тоді як TFT інтегрує додаткові статичні дані – тип ґрунту, характер ураження, географічне положення. Це дозволяє досягти більшої точності у прогнозуванні й робить систему стійкою до різномірних вхідних даних.

2. Архітектура запропонованої системи моніторингу та аналізу пошкоджень аграрних ділянок

Архітектуру системи, що пропонується, поєднує ключові моделі для оцінки стану сільськогосподарських земель, пошкоджених унаслідок військових дій. Кожна модель відповідає за своє завдання: виявлення та класифікацію пошкоджених ділянок, сегментацію пошкоджень і прогнозування часу відновлення. Система побудована так, щоб бути простою у використанні, масштабованою та ефективною навіть за обмеженістю даних моніторингу.

В системі реалізовано трирівневу клієнт-серверну архітектуру, що дозволяє чітко розмежувати логіку користувача, бізнес-логіку та обчислювальні модулі штучного інтелекту. Такий підхід забезпечує високу модульність, спрощує підтримку, масштабування системи та можливість подальшого розширення без зміни всієї структури. Архітектура системи містить презентаційний рівень, логічний рівень та рівень моделей.

Презентаційний рівень (UI – User Interface) є інтерфейсним рівнем взаємодії користувачів із системою. У системі інтерфейс розроблено з використанням Windows Presentation Foundation (WPF) – сучасного інструменту від Microsoft, що дозволяє будувати гнучкі, масштабовані та стильні десктопні застосунки на мові C#.

Інтерфейс надає користувачам системи можливість: завантажувати зображення з локального комп'ютера; запускати процедури виявлення, класифікації та сегментації пошкоджених ділянок; здійснювати прогнозування терміну ліквідації наслідків; переглядати результати обробки; зберігати результати в зручному форматі (PDF або DOCX).

Елементи керування (кнопки, комбо-боксы, текстові блоки, зображення тощо) розміщено на формі MainWindow.xaml. Для кожного елемента передбачено обробник подій у відповідному .cs-файлі.

Логічний рівень (Business Logic Layer) відповідає за опрацювання всієї логіки взаємодії між користувачем і моделями. Він реалізований у коді MainWindow.xaml.cs, де обробляються події активації кнопок, здійснюється перевірка вхідних даних, координується виклик моделей ONNX, обробляються результати та передаються в UI, формується звіт.

Клас MainWindow містить також змінні для зберігання проміжних результатів: шлях до зображення,

результат класифікації, згенерована маска та прогнозовані значення. Всі моделі запускаються асинхронно, щоб не блокувати інтерфейс користувача.

Рівень моделей (AI Models / Data Layer) відповідає за всю обробку даних з використанням нейромережових технологій. До нього входять класи, які завантажують та запускають моделі ONNX M1, M2 та M3, збережені у форматі, сумісному з ONNX Runtime. Для кожної з моделей створені окремі класи:

- OnnxModelRunner (M1): виконує класифікацію зображення на дві категорії: уражене / неуржене поле ;
- UnetModelRunner (M2): реалізує сегментацію зображення за допомогою технології U-Net та формування маски пошкодженої області;

- LstmOnnxRunner (M3): використовується для прогнозування ймовірної площі ураження та часу відновлення посівів на основі послідовних аграрних і погодних даних.

Всі моделі попередньо створено та навчені в середовищі Python (з використанням PyTorch, Pandas, Sklearn), а потім експортовано у формат ONNX (Open Neural Network Exchange). Це дозволяє легко запускати їх у .NET-застосунках без втрати продуктивності. Розглянемо детальніше особливості розроблених моделей.

Модель M1 призначена для виявлення пошкоджених ділянок та класифікації їхніх зображень за ознаками таких уражень, як вирви, забруднення від осколків чи хімічних речовин. Тут використовується згортоква нейронна мережа (CNN) із застосуванням техніки регуляризації Shortcut, що дозволяє уникнути проблеми згасання градієнта та втрати важливих деталей шляхом створення прямих зв'язків між шарами CNN.

Для активації згорткових шарів, а також для повнозв'язного шару, обрано функцію PReLU (Parametric Rectified Linear Unit). На виході модель формує ймовірності належності аналізованих зображень до певних класів через функцію Softmax, що забезпечує коректну інтерпретацію ймовірностей класів. Після класифікації зображень на наступному етапі аналізуються зображення з класу «уражені ділянки».

Створення моделі відбувається за допомогою бібліотеки від Microsoft ML.NET, яка добре працює із C#. Початкове тренування моделі реалізується у середовищі Python через каркас TensorFlow. Потім сформована модель переводиться у формат ONNX і підключається до C#.

Знімки для аналізу надходять із супутників Sentinel-2 (зокрема зі смуги B4, яка добре відображує зміни в рослинності) або з дронів [12]. Перед надходженням до моделі аналізовані зображення проходять попередню підготовку: нормалізацію для вирівнювання яскравості та контрастності, а також аугментацію (наприклад, з використанням поворотів чи зміни масштабу) для покращення узагальнюючої здатності моделі та її стійкості до варіацій у даних.

Ця модель інтегрується в систему як перший етап

аналізу. Користувач завантажує зображення через інтерфейс WPF, і модель за лічені секунди визначає наявність або відсутність пошкоджень на аналізованій ділянці. Якщо ділянка класифікується як «уражена», зображення передається до моделі M2 для більш детального аналізу. Результати зберігаються у CSV-файлі для подальшого використання або передачі до бази даних, якщо система масштабуватиметься.

Модель M2 призначена для детального аналізу та сегментацію зображень, класифікованих як «уражені», щоб визначити точне розташування, форму, площу та інтенсивність пошкоджень. Для цього впроваджено архітектуру U-Net – згорткову нейронну мережу з енкодер-декодерною структурою, яка широко застосовується для семантичної сегментації, зокрема в задачах аналізу супутникових зображень. U-Net складається з енкодера, який поступово зменшує розмірність зображення, і декодера, що відновлює просторову інформацію, дозволяючи точно визначити контури, площу, тип та рівень інтенсивності уражень ґрунту, що є критично важливим для подальшого аналізу та прийняття рішень. Дана архітектура ефективно виділяє контури уражених ділянок, таких як вирви чи забруднені зони, завдяки своїй здатності зберігати просторову інформацію через пропуски з'єднання між енкодером і декодером. Крім того U-Net демонструє високу стійкість до шумів на зображеннях, що є особливо важливим при роботі з аерофотознімками та супутниковими даними низької якості. Завдяки цьому модель забезпечує точні результати навіть за умов обмеженої роздільної здатності вхідних зображень. Варіант архітектури мережі U-Net, що використано в моделі M2, наведено на рис. 2.

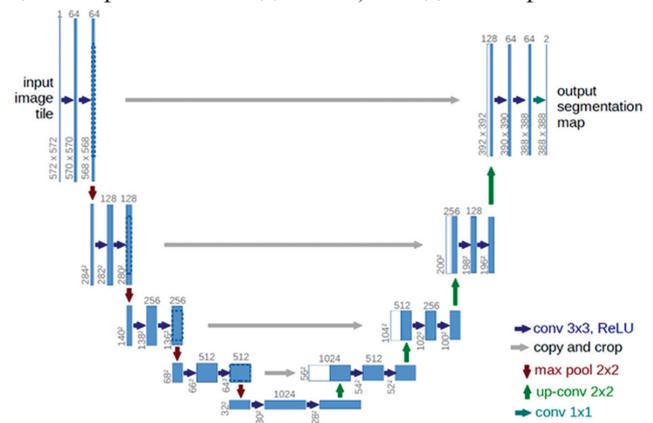


Рис. 2. Архітектура мережі U-Net (модель M2)

Енкодерна частина мережі U-Net складається з кількох згорткових шарів, які поступово зменшують розмірність зображення, виділяючи ключові ознаки, такі як краї вирв чи зміни в текстурі ґрунту. Декодерна частина відновлює зображення до початкової роздільної здатності, створюючи піксельну маску, де кожен піксель позначений як «пошкоджений» або «непошкоджений». Для активації шарів знову використовується функція PReLU, щоб забезпечити стабільне навчання.

Модель тренується в Python за допомогою бібліоте-

ки PyTorch, що пропонує гнучкі інструменти для роботи з U-Net і обробки великих зображень. Для навчання використовуються датасети, що містять супутникові знімки Sentinel-2 і аерофотознімки з дронів з анотаціями, які позначають пошкоджені ділянки. Смуга B4 Sentinel-2 є особливо ефективною для виявлення змін у рослинності, тому використовуємо її як основний канал, додаючи смуги B2 і B3 для покращення контрастності [13]. Попередня обробка в цій моделі потребує формування датасету з фотографій пошкоджених земель та метаданих, таких як тип використаної зброї, площа ураження та час, необхідний для відновлення. Метадані структуруються та стандартизуються для забезпечення коректності подальшого аналізу. Додатково виконується попередня обробка зображень, зокрема нормалізація яскравості та контрастності, а також видалення шуму за допомогою гаусівського фільтра, щоб уніфікувати дані.

Наступним кроком є тестування та валідація моделі штучного інтелекту (МШІ) M2, де використовуються метрики Mean Squared Error (MSE) для оцінки точності прогнозування часу, Intersection over Union (IoU) – для визначення площі ураження, а також Recall, Precision та F1-score для класифікації пошкоджень. Ці метрики дозволяють переконатися, що модель точно виділяє межі пошкоджень із мінімальними помилками. Верифікація моделі M2 виконується на тестовому наборі даних, що містить раніше невикористовувані зображення, щоб оцінити її узагальнену здатність.

Концептуальний аналіз МШІ M2 зосереджується на виявленні помилок, їх усуненні через донавчання та ансамблеві методи. Оцінюється вплив погодних умов і якості зображень, а також можливість інтеграції супутникових та дронів даних для покращення прогнозування.

Після тренування модель M2 експортується у формат ONNX і інтегрується в C# через ONNX Runtime, що забезпечує швидку обробку зображень у реальному часі. Таким чином, ця модель працює як другий етап реалізації функцій системи (після того, як модель M1 визначила зображення як «уражені», мережа U-Net створює детальну піксельну карту пошкоджень, чітко окреслюючи контури уражених зон). Сформована карта пошкоджень відображається в інтуїтивному інтерфейсі WPF, де користувач може побачити білі області пошкоджень на чорному тлі разом із числовими даними про площу ураження (в квадратних метрах або гектарах). Така візуалізація допомагає користувачам (фермерам чи екологам) швидко оцінити масштаби проблеми та планувати подальші дії.

Модель M3 призначена для прогнозування ймовірної площі ураження та часу відновлення посівів на основі аналізу таких факторів, як площа пошкодження, характер пошкодження (вибухи, хімічне забруднення тощо), тип ґрунту та кліматичні умови. Для цього в моделі M3 використано комбінацію двох

архітектур: Temporal Fusion Transformer (TFT) для обробки статичних і різномірних даних, таких як тип зброї чи геолокація, і Long Short-Term Memory (LSTM) для аналізу динамічних змін з урахуванням сезонних чи погодних факторів. Цей підхід дозволяє створювати точні прогнози навіть у динамічних умовах.

TFT – це гібридна модель, яка поєднує механізми уваги (attention) і обробки часових рядів. Ця модель аналізує статичні дані (наприклад, тип ґрунту чи місце розташування), поточні динамічні дані (наприклад, опади чи температура) і формує узагальнені результати прогнозування. Архітектура LSTM, у свою чергу, відстежує довгострокові залежності (зокрема, поступове відновлення родючості ґрунту), що залежить від сезонних циклів. Спільне застосування TFT і LSTM в моделі M3 забезпечує точні оцінки часу рекультивації з урахуванням поточного стану поля та динаміки його зміни.

Модель M3 тренується в Python за допомогою бібліотеки PyTorch, що здатна підтримувати архітектури TFT і LSTM. Вхідними даними цієї моделі є результати сегментації (площа та інтенсивність пошкоджень), а також дані з відкритих джерел (наприклад, тип ґрунту чи кліматичні показники, збережені у CSV-файлах). Для підвищення точності моделі використовуються налаштування за допомогою бібліотеки KerasTuner, що дозволяють визначити найкращі значення таких параметрів як розмір шарів і швидкість навчання. Оцінка якості моделі здійснюється з використанням метрики MAE (Mean Absolute Error) для перевірки точності прогнозу часу відновлення, а також метрики R для загальної оцінки якості моделі. Готова модель підключається до C# через формат ONNX Runtime для швидкого формування прогнозів. У вихідному звіті моделі зазначається, скільки часу потребує відновлення (прогнозований час відновлення відображається в інтерфейсі WPF) і які методи рекультивації мають бути найбільш ефективними. Дані звітів зберігаються в CSV, щоб можна було повернутися до них в разі потреби [14].

Інтеграція моделей. Моделі M1, M2 та M3 є базовими складовими елементами системи моніторингу та аналізу пошкоджень аграрних ділянок.

Передача результатів між цими моделями реалізується через внутрішні змінні та допоміжні структури (у тому числі з використанням тимчасових файлів). Усі етапи обробки є асинхронними, що дозволяє забезпечити плавну взаємодію з інтерфейсом системи і не блокувати роботу застосунку під час обчислень. Завдяки чіткому розмежуванню функцій моделей, послідовність їх роботи (класифікація сегментація прогноз) зберігає гнучкість, дозволяючи в майбутньому доповнювати систему новими функціональними блоками.

Архітектура системи забезпечує повну автоматизацію аналізу зображень, дозволяючи швидко обробляти дані та формувати практичні висновки й рекомендації для користувачів. Усі етапи обробки інтегровані в

єдиний процес у рамках системи підтримки прийняття рішень (DSS), що робить систему повноцінним аналітичним інструментом. Завдяки модульній структурі система залишається гнучкою, дозволяючи легко додавати нові функції, наприклад, інтеграцію з геоінформаційними системами.

У системі реалізовано чіткий і структурований потік даних, що забезпечує безперебійну обробку інформації та злагоджену взаємодію між усіма компонентами: інтерфейсом користувача, логікою обробки, моделями штучного інтелекту (M1, M2, M3) та модулем генерації звітів.

Потік даних у системі розпочинається із завантаження користувачем зображення земельної ділянки у форматі .jpg або .png через інтерфейс WPF. Після цього зображення автоматично передається до логічного ядра програми, яке координує подальшу їх послідовну обробку з використанням моделей M1, M2 та M3. Структура даних розробленої системи організована таким чином, щоб забезпечити зручність обміну між компонентами, швидкий доступ до проміжних результатів та можливість формування фінального звіту без втрати інформації. Враховуючи поетапну обробку та модульність архітектури, дані зберігаються у вигляді окремих змінних у кодї, тимчасових файлів, а також графічних об'єктів.

Після завантаження зображення його шлях зберігається у вигляді рядка (imagePath), а саме зображення завантажується у вигляді об'єкта BitmapSource для подальшої обробки в пам'яті. Результати класифікації (currentPrediction) і прогнозу (currentForecast) зберігаються окремо у вигляді текстових змінних, які відображаються у відповідних текстових полях інтерфейсу.

Маска, отримана в процесі сегментації, зберігається двічі: як об'єкт у пам'яті (currentMask типу BitmapSource) і як зображення у форматі .png, що дозволяє використати її в PDF/DOCX звіті. Усі ці об'єкти передаються до модуля генерації звітів у структурованому вигляді, без необхідності повторної обробки. Для забезпечення узгодженості між модулями використовуються тимчасові файлові структури, такі як CSV-файли, де зберігалися результати класифікації, площа пошкодження, координати маски та прогноз, що спрощувало логіку обробки та дозволяло легко масштабувати систему чи інтегрувати її з базами даних або хмарними сервісами.

Остаточні результати зберігаються у CSV-файлі та відображаються в інтерфейсі у вигляді зрозумілого текстового звіту, що містить числові дані та практичні поради.

Для забезпечення надійності потоку даних у системі реалізовано механізми контролю якості. Наприклад, якщо зображення має низьку роздільну здатність або не відповідає заздалегідь зазначеним вимогам, система видає повідомлення про помилку, пропонуючи користувачу перевірити дані.

Така організація потоку даних у поєднанні зі структурою DSS робила ефективним інструментом для оцінки стану земель і планування їхнього відновлення. Вона в подальшому може бути вдосконалена шляхом інтеграції з додатковими базами даних (для роботи з більшими обсягами інформації) або підключення до реальних супутникових потоків для аналізу в реальному часі.

3. Програмна реалізація функцій системи та результати тестування

Для програмної реалізації функцій системи було використано сучасний набір технологій і інструментів, що забезпечують ефективну реалізацію функціоналу машинного навчання, зручного десктопного інтерфейсу, обробки зображень і формування звітів. Архітектура системи була спроектована з акцентом на продуктивність, модульність і можливість масштабування, що дозволяє легко адаптувати її до нових вимог і забезпечувало стабільну роботу моделей M1, M2 та M3 у реальних умовах.

Основним середовищем для створення, навчання та тестування моделей ШІ були мова Python та низка підтримуваних цією мовою бібліотек, а саме:

– PyTorch: ця бібліотека стала основою для побудови всіх трьох моделей системи – згортової нейронної мережі (CNN) для класифікації (M1), архітектури U-Net для сегментації (M2) та комбінації Long Short-Term Memory (LSTM) і Temporal Fusion Transformer (TFT) для прогнозування (M3) [15];

– Pandas: використовувалася для обробки табличних даних, таких як кліматичні показники (температура, опади), характеристики ґрунтів чи метадані про типи уражень;

– NumPy: застосовувалася для виконання числових операцій, зокрема нормалізації та трансформації даних перед подачею їх у моделі;

– scikit-learn: використовувалася для крос-валідації моделей і оцінки їхньої якості за такими метриками, як Precision, Recall, F1-score (для класифікації та сегментації) і (для прогнозування);

– Albumentations: забезпечувала аугментацію зображень (обертання, зміна яскравості, додавання шумів), що дозволяло підвищити стійкість моделей до різних умов освітлення чи якості знімків.

Після завершення навчання моделі експортувалися у формат ONNX (Open Neural Network Exchange), що дозволяло використовувати їх у C# додатку без залежності від Python. Такий підхід забезпечував швидку інтеграцію моделей у десктопне середовище та оптимізував продуктивність системи.

Клієнтська частина була реалізована на платформі .NET 8.0, що гарантувала підтримку асинхронного програмування. Тут використовувалися такі технології:

– WPF (Windows Presentation Foundation): ця технологія стала основою для створення інтуїтивного

графічного інтерфейсу користувача. В інтерфейсі відображалися зображення ділянок, маски сегментації, текстові результати класифікації та прогнози, а також елементи керування для завантаження файлів і перегляду звітів;

– ONNX Runtime for .NET: використовувалася для запуску моделей ШІ у форматі ONNX безпосередньо в C# середовищі. Це дозволяло обходитися без Python залежностей, забезпечуючи високу швидкість виконання моделей і їхню інтеграцію з десктопним додатком. ONNX Runtime оптимізувала обчислення, використовуючи апаратне прискорення (CPU або GPU);

– System.Drawing та System.Windows.Media.Imaging: ці бібліотеки відповідали за обробку зображень, зокрема завантаження вхідних знімків, рендеринг масок сегментації та їх збереження у форматі .png для звітів. Вони забезпечували якісну візуалізацію результатів у реальному часі;

– асинхронне програмування (async/await): використовувалося для паралельного виконання обчислень зокрема, обробки зображень моделями) і оновлення інтерфейсу, що дозволяло уникнути затримок і забезпечувало плавну взаємодію користувача з системою.

Для стабільної роботи моделі прогнозування (М3) та забезпечення точності аналізу система передбачає інтеграцію з зовнішніми джерелами даних:

– супутникові знімки: система розрахована насамперед на роботу із зображеннями від Sentinel-2 і Landsat, які надають високоякісні дані для аналізу сільськогосподарських земель. Ці джерела можуть використовуватися для отримання знімків у реальному часі або з архівів, що дозволяє відстежувати стан ділянок у динаміці;

– відкриті погодні API: для прогнозування часу відновлення земель використовуються дані з таких сервісів, як Open-Meteo і Meteostat, які надають часові ряди кліматичних показників (опади, температура, вологість). Ці дані є важливими для моделі М3, оскільки впливають на оцінку рекультивацийних заходів;

– публічні аграрні датасети: для навчання моделей використовуються набори даних FAO (Food and Agriculture Organization) і CropHarvest, що містять приклади уражених земель, типи ґрунтів і характеристики рослинності. Це дозволяє моделям «навчатися» на реальних сценаріях і підвищує їхню точність.

Набір даних для побудови та тестування системи моніторингу та прогнозування пошкоджень сільськогосподарських земель (з використанням моделей М1, М2 та М3) складався з двох частин: зображень уражених і неуражених ділянок, поділених на тренувальну, валідаційну та тестову вибірки з категоріями damaged/undamaged. Кожна вибірка має відповідні маски сегментації у форматі PNG та CSV-файли з метаданими із додатковою інформацією про об'єкти. Така структура дозволяє об'єднати зображення, числові та текстові дані для навчання комбінованих моделей.

Для навчального датасету було обрано обмежену сукупність реальних вхідних зображень та програмно згенеровані сегментаційні маски пошкоджень. Штучно сформовано окремий файл формату CSV з метаописом датасету. У файлі CSV зафіксовані додаткові характеристики кожного випадку руйнування (інформація про місце і час події, тип руйнування та джерело походження даних). Заповнення цього файлу здійснювалося не лише на основі супутникових знімків та фотографій з дронів, але й з відкритих новинних ресурсів (зокрема з публікацій військових порталів «Militalnyi», де регулярно подаються підтвержені факти знищення військової техніки чи наслідків бойових дій). Такий підхід дозволив поєднати в одному наборі як візуальні дані, так і текстову інформацію, що значно розширило можливості аналізу й забезпечило основу для побудови ефективних прогнозних моделей. Весь набір даних структуровано у вигляді двох основних директорій: директорія images містить оригінальні знімки у форматі .jpg; директорія masks містить відповідні сегментаційні маски у форматі .png, що позначають межі пошкоджених ділянок. Приклад пари «знімок — маска» наведено на рис. 3.

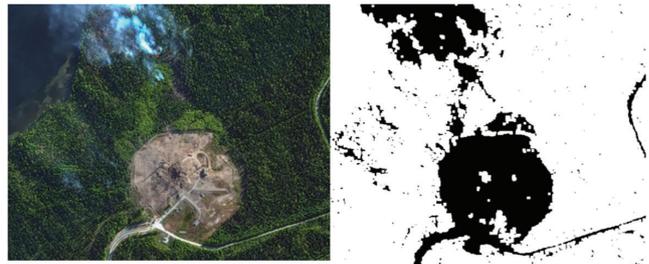


Рис. 3. Приклад вхідного знімка та відповідної маски пошкодженої ділянки

Інтеграція зовнішніх джерел робить систему гнучкою і готовою до роботи в різних умовах, наприклад, у регіонах із різними кліматичними чи ґрунтовими характеристиками. У майбутньому систему можна розширити, додавши підтримку інших API (наприклад, для моніторингу забруднень чи аналізу хімічного складу ґрунтів) або підключивши реляційні бази даних для збереження історії аналізів.

Поєднання мов Python (для створення моделей) та C# (.NET + WPF для інтерфейсу та запуску моделей) забезпечує баланс між гнучкістю машинного навчання і стабільністю та зручністю десктопного застосунку. Обрані інструменти дозволили реалізувати комплексну, автономну систему, яка може працювати локально без підключення до серверів, а також масштабуватися до веб або хмарної архітектури в майбутньому.

Розглянемо деякі особливості програмної реалізації моделей системи та результати її тестування.

Для класифікації зображень (модель М1) було використано згорткову нейронну мережу (CNN), побудовану на основі ResNet-18 [16]. Архітектура була модифікована для роботи з обмеженим набором зображень і забезпечення високої точності:

- змінено останній шар: `fc = nn.Linear(512, 2)`;
- додано технологію Dropout перед повнозв'язним шаром для зменшення перенавчання;
- використано функцію активації PReLU для адаптації до темних зон.

Програмний код для навчання моделі M1 наведено нижче:

```
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import datasets, transforms, models
from torch.utils.data import DataLoader

model =
models.resnet18(weights=models.ResNet18_Weights.
IMAGENET1K_V1)
model.fc = nn.Sequential(
    nn.Dropout(0.5), # Add Dropout
    nn.Linear(model.fc.in_features, NUM_CLASSES)
device = torch.device("cuda" if torch.cuda.is_available()
else "cpu")
model.to(device)
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.0001,
weight_decay=1e-4)
```

Тренування моделі M1 здійснювалося у середовищі PyTorch на невеликому датасеті: 36 зображень у train, 10 зображень у validation. Через обмежену кількість даних було застосовано аугментації (RandomFlip, ColorJitter) та знижено learning rate (0.0001).

Ключові параметри:

- Loss: CrossEntropyLoss;
- Optimizer: Adam;
- Epoch: 15;
- Batch Size: 4;
- Augmentation: доступна в train_transforms;
- EarlyStopping: активований (patience=3).

Отримані результати підтверджують успішне навчання моделі.

На валідаційному наборі було досягнуто 100% точності, що свідчить про високу ефективність класифікації саме на доступних прикладах. Мінімальне значення втрат на валідації (Val Loss) зафіксовано на 10-й епісі (0.0158).

Такого результату вдалося досягти завдяки використанню технології регуляризації Dropout, а також аугментації зображень, що дозволили штучно збільшити варіативність навчальних даних. Низький learning rate забезпечив стабільне навчання без різких коливань у значеннях втрат.

Після успішного навчання та експорту моделі у формат ONNX, вона була інтегрована в десктопний застосунок на мові C# із використанням бібліотеки Microsoft.ML.OnnxRuntime. Цей формат дозволяє виконувати нейронну мережу незалежно від середовища навчання (наприклад, без запуску Python), що ідеально підходить для .NET-додатків.

У програмі реалізовано окремий клас OnnxModelRunner, відповідальний за завантаження моделі, передобробку зображення та отримання результату класифікації. Модель читається з файлу m1_model.onnx, який розміщується в директорії Models/.

Зображення, яке завантажується користувачем, спочатку конвертується у формат тензора (Tensor<float>). Перед подачею в модель воно: масштабується до розміру 224×224 пікселі; нормалізується; переводиться з формату Bitmap до багатовимірного масиву DenseTensor<float>.

Передобробка реалізується в окремому методі PreprocessImage, який також входить до складу OnnxModelRunner.cs.

Результат у вигляді двох чисел (ймовірностей для класу «уражена» і «неуражена») зчитується з виходу моделі. Далі проводиться порівняння: якщо ймовірність класу «уражена» вища за порогове значення, то формується відповідне повідомлення, що відображається у інтерфейсі WPF за допомогою елемента ResultTextBlock. Також результат зберігається у змінній currentPrediction, що дає змогу використати його надалі у звітах (у форматах PDF або DOCX).

Цей програмний модуль забезпечує швидке та стабільне розгортання моделі класифікації M1 безпосередньо у графічному інтерфейсі.

Для сегментації зображень пошкоджених ділянок (модель M2) було використано згорткову нейронну мережу з архітектурою U-Net, що складається з двох частин: стискуючої (енкодер) та розширюючої (декодер). В енкодері вхідне зображення проходить через кілька рівнів згорткових шарів із функцією активації PReLU, після чого стискається за допомогою операції max pooling. Це дозволяє зменшити розмір зображення, зберігаючи при цьому найбільш значущі ознаки. Декодер виконує зворотну операцію – поступово відновлює просторову роздільну здатність за допомогою транспонованих згорткових шарів (ConvTranspose2d) і об'єднує (через skip connections) відповідні рівні з енкодера. У фіналі модель формує карту пікселів, де кожен піксель має значення від 0 до 1 – ймовірність, що він належить до пошкодженої зони.

Реалізація архітектури в проєкті була виконана за допомогою фреймворку PyTorch..

На етапі навчання для кожного зображення, що міститься в каталозі, виконується попередня обробка. Всі зображення масштабуються до фіксованого розміру 128×128 пікселів (це оптимальний розмір для швидкого навчання на малих об'ємах даних). Після цього виконується нормалізація: значення пікселів діляться на 255, щоб привести їх до діапазону [0, 1], що є стандартом для нейронних мереж. Поряд із зображенням обробляється і відповідна маска – чорно-біле зображення з тією ж роздільною здатністю, де білим кольором позначені пошкоджені області, а чорним – фон. Маска також масштабується до 128×128, конвертується у однональний тензор [1, H, W] та нормалізується.

Далі зображення та маска передаються у модель у вигляді тензорів типу `torch.Tensor`. U-Net обробляє зображення, проходячи крізь усі згорткові та декодувальні шари, після чого формує вихідну карту ймовірностей, де кожному пікселю призначається значення від 0 до 1, яке інтерпретується як ймовірність того, що цей піксель належить до пошкодженої ділянки.

В результаті сегментації формується бінарна маска, яка зберігається в окремий файл (`segmented_mask.png`) і водночас передається до інтерфейсу користувача в WPF для візуалізації. Повний код навчання моделі для генерації масок наведено у додатку Г. У WPF-модулі ця маска прив'язується до елемента `MaskImage`, де користувач може переглянути сегментовану ділянку прямо на екрані.

Функціональною реалізацією цього етапу в C# є клас `UnetModelRunner`, що відкриває модель `unet_model.onnx`, проводить необхідні перетворення зображення, запускає модель та повертає об'єкт типу `Bitmap`, який далі передається у `Image.Source`.

Під час навчання моделі на обмеженому наборі даних модель M2 демонструвала поступове покращення. Втрати на тренувальному наборі (loss) зменшилися з 4.18 у першій епісі до 2.11 у десятій. Це підтверджує, що навіть на малому наборі даних мережа здатна навчитися узагальнювати просторові патерни, характерні для пошкоджених ділянок.

Після завершення навчання модель M2 експортувалася у формат ONNX для використання у .NET-застосунку. Файл `unet_model.onnx` зберігається у директорії `Models/` і використовується в десктопному застосунку на етапі сегментації.

Для прогнозування часу відновлення пошкоджених ділянок на основі результатів сегментації (модель M3) було використано комбіновану архітектуру ШНМ, що поєднує можливості технологій TFT і LSTM.

Модель M3 була реалізована у фреймворку `PyTorch` як комбінація двох підходів до аналізу часових рядів:

- LSTM відповідає за обробку послідовних змін параметрів у часі (наприклад, зміни вологості, опадів або температури, що впливають на швидкість відновлення ґрунту);

- TFT забезпечує обробку як динамічних, так і статичних вхідних ознак із використанням механізму уваги, що дає змогу фокусуватися на ключових змінах у часовому ряді та адаптивно зважувати інформацію.

Архітектура передбачає наявність таких вхідних даних:

- нормалізована площа ураження (за результатами сегментації);
- кліматичні характеристики (метеодані за останні 12 місяців);
- тип ґрунту (категоріальний вхід);
- тип ураження (визначений за площинними шаблонами або за класифікатором).

Ці входи об'єднуються в єдиний багатовимірний тензор, який подається на вхід LSTM та декодується

TFT-блоком. Програмний код для реалізації комбінації LSTM та TFT наведено нижче:

```
class SimpleLSTMTransformer(nn.Module):
    def __init__(self, input_size, hidden_size, output_size=1):
        super(SimpleLSTMTransformer, self).__init__()
        self.lstm = nn.LSTM(input_size, hidden_size, batch_first=True)
        self.linear = nn.Linear(hidden_size, hidden_size)
        self.transformer = nn.TransformerEncoder(
            nn.TransformerEncoderLayer(d_model=hidden_size, nhead=2, batch_first=True),
            num_layers=1)
        self.output = nn.Linear(hidden_size, output_size)
    def forward(self, x):
        x, _ = self.lstm(x)
        x = self.linear(x)
        x = self.transformer(x)
        return self.output(x[:, -1, :])
```

У вхідному тензорі моделі M3 присутні як **статичні параметри** (тип ґрунту, геолокація) так і **динамічні часові ряди** (вологість, температура, кількість опадів, площа пошкоджень у часі). Дані було згенеровано шляхом моделювання та симуляції, з урахуванням реальних шаблонів клімату. Вхідні значення масштабуються до діапазону [0, 1] за допомогою програми `MinMaxScaler` із бібліотеки `sklearn`. Для time-series входу формується масив розмірності (batch, time_steps, features), де `time_steps=50` і `features=2`.

Навчання моделі під час її тестування проводилося протягом 150 епох. У процесі навчання втрати зменшувалися поступово – з початкового значення 10.8050 до мінімуму 0.8116 на 120-й епісі. Графік втрат мав деякі коливання, однак загальний тренд залишався низхідним. Результати навчання комбінованої моделі прогнозування свідчать про поступове покращення здатності моделі формувати точні прогнози.

Отримані результати узгоджувалися з наступною логікою: при більшій площі пошкодження модель M3 прогнозувала триваліший час відновлення, що підтверджує її коректну поведінку. Водночас слід враховувати, що модель не враховує важливі зовнішні чинники, зокрема економічні ресурси, політичні рішення чи наявність техніки для рекультивациі – її прогноз базується виключно на фізичних характеристиках.

Після завершення навчання модель експортувалася у формат ONNX для використання у C# -застосунку.

Вихідні результати роботи моделі M3 (прогнозований час відновлення у місяцях) автоматично передаються до інтерфейсу користувача і зберігаються у змінній `currentForecast` для подальшого їх занесення до відповідного звіту.

Головне вікно інтерфейсу запропонованої системи побудоване за принципом трирівневої логіки: ліворуч

розміщені кнопки керування, а праворуч – вікна для відображення результатів обробки (рис. 4).

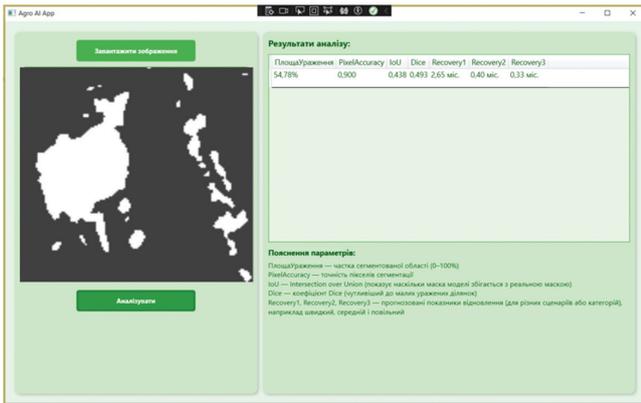


Рис. 4. Вікно інтерфейсу desktop-додатку

Після запуску додатку користувач бачить панель з такими основними етапами:

- завантаження зображення – відкриває діалогове вікно для вибору фото;
- класифікація (M1) – визначає, чи зображення є ураженим;
- сегментація (M2) – при потребі створює маску пошкоджень;
- прогноз (M3) – оцінює орієнтовний час відновлення;
- формування звіту – дозволяє зберегти результати у форматах PDF або DOCX.

На рис. 5 наведено приклад візуалізації результатів обробки зображення.

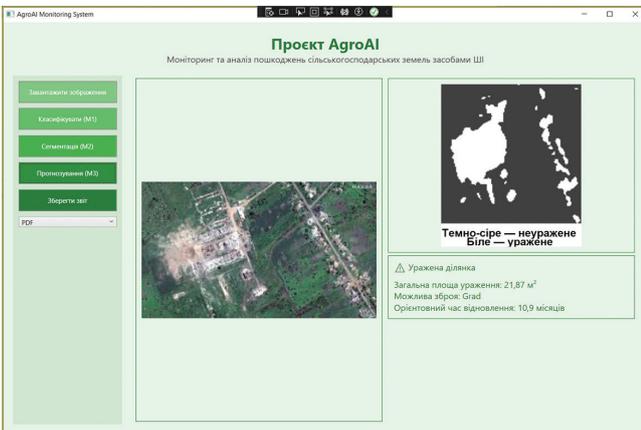


Рис. 5. Приклад візуалізації результатів обробки зображення

Усі результати обробки відображаються в інтерфейсі у вигляді тексту та графіки (маска, карта пошкоджень, числові значення). Всі етапи виконуються покроково, і кожен з них ініціалізується лише після успішного виконання попереднього.

Реалізація функціональних можливостей системи має сприяти більш ефективному плануванню використання земельних ресурсів і прискоренню процесів рекультивациі, що відповідає цілі сталого розвитку 15 – «Захист і відновлення екосистем суші». Це підкреслює значення технологій штучного інтелекту не лише в контексті оперативного реагування на воєнні наслідки,

а й як стратегічного інструменту для довгострокового сталого розвитку сільського господарства [17].

Слід зазначити, що у процесі тестування системи було виявлено низку певних обмежень. Зокрема, класифікаційна модель через малий обсяг навчальних даних схильна до перенавчання. Якість сегментації залежить від вхідного зображення: в ідеальних умовах вона є висока, але може погіршуватися при зміні умов зйомки. Модель прогнозування, що показала прийнятні результати на тестових даних, може потребувати подальшої адаптації для реальних сценаріїв, де до фізичних параметрів додаються соціальні та економічні фактори.

Крім того, необхідно визначити деякі проблеми використання сегментаційних масок. Хмари (для моделей M2 та M3) розпізнавались та позначались в цих масках тим самим кольором, що і уражена ділянка, що може призвести до помилок в детекції, оскільки в оптичних супутникових зображеннях хмари часто імітують деградацію ґрунтів, особливо в аграрних регіонах з сезонними хмарами, де помилки можуть досягати 30% без препроцесингу.

У подальшому пропонується додати клас «не розпізнано», до якого буде віднесена частина зображення з відповідним кольором позначення, у тому числі і хмари. При цьому бінарна модель (уражене/неуражене) перетворюється на трикласову, зменшуючи false positives на 15–25% в разі наявності з хмар на зображеннях. Це дозволяє уникати примусової класифікації невизначених пікселів, що особливо корисно для невеликих датасетів, де присутній ризик перенавчання. Для препроцесингу зображень в цьому разі можна використати алгоритми виявлення хмар перед сегментацією, такі як RS-Net або attention-based U-Net, які аналізують мультиспектральні канали для створення масок хмар. Додатково можуть бути використані GAN-моделі для видалення хмар з реконструкцією [18].

Результати виконання етапів роботи запропонованої інтелектуальної системи виявлення та аналізу руйнівних пошкоджень аграрних ділянок (включно з визначенням типом пошкодження, оціненою площею ураження, можливим типом застосованої зброї та прогнозованим терміном відновлення) об'єднуються в єдиний звіт. Цей звіт також містить карту пошкоджень у вигляді сегментованого зображення, що надає користувачу не лише числову інформацію, а й візуальне подання результатів.

У підсумку користувач отримує чітку й структуровану аналітичну довідку, яка може бути збережена у форматах PDF або Word і використана для подальших дій, включаючи планування рекультивацийних заходів.

Висновки

В статті наведено результати розроблення та дослідження інтелектуальної системи виявлення та аналізу руйнівних пошкоджень аграрних ділянок з

використанням сучасних нейромережових технологій. Система базується на послідовному використанні трьох моделей штучного інтелекту, функціями яких є:

- класифікація пошкоджень за допомогою згорткової нейронної мережі (CNN);
- сегментація уражених ділянок із використанням архітектури U-Net;
- прогнозування часу відновлення земель на основі комбінації Temporal Fusion Transformer (TFT) і Long Short-Term Memory (LSTM).

Така послідовність дозволяє системі: визначити наявність пошкоджень за результатами аналізу зображень аграрних ділянок; створювати детальні карти уражених зон з точними контурами; прогнозувати термін відновлення з урахуванням площі ураження, типу ґрунту, кліматичних умов і характеру пошкоджень; формувати звіти у форматах PDF або DOCX, які користувач може використовувати для планування рекультивациі.

Всі моделі системи попередньо створюються та навчаються в середовищі Python (з використанням бібліотек PyTorch, Pandas, Sklearn), а потім експортуються у формат ONNX. Це дозволяє ефективно використовувати їх у .NET-застосунках з забезпеченням швидкої обробки даних.

Результати тестування підтверджують працездатність запропонованої системи.

Система має значний потенціал для розвитку: її можна інтегрувати з супутниковими сервісами для аналізу даних у реальному часі, підключити до геоінформаційних систем (GIS) для створення детальних аграрних карт або синхронізувати з державними реєстрами для здійснення більш детальної аналітики. Наприклад, додавання модуля для порівняння знімків у динаміці може сприяти прогнозуванню зміни стану земель, а інтеграція з базами даних про мінну небезпеку підвищувати безпеку планування рекультивацийних робіт.

Список літератури:

- [1] Armed Forces of Ukraine destroyed the Russian Grad multiple rocket launcher with a drone in the Donetsk region. *Military*. URL: <https://military.com/en/news/armed-forces-of-ukraine-destroyed-the-russian-grad-multiple-rocket-launcher-with-a-drone-in-the-donetsk-region/> (дата звернення: 25.05.2025).
- [2] Екологічний моніторинг ландшафтних ділянок з використанням регуляризованих штучних нейронних мереж. / С. Удовенко та ін. *Біоніка інтелекту*. 2022. Т. 1 № 94. С. 13–22. URL: [https://doi.org/10.30837/bi.2020.1\(94\).03](https://doi.org/10.30837/bi.2020.1(94).03) (дата звернення: 20.05.2025).
- [3] Drozd S., Kussul N., Shelestov A. Satellite-Based Analysis of Forest Damage in Ukraine's Protected Areas. 13th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. Gliwice, Poland. 4–6 September, 2025.
- [4] Schmidhuber J. Deep learning in neural networks: An overview. *Neural Networks*. 2015. Т. 61. С. 85–117. URL: <https://doi.org/10.1016/j.neunet.2014.09.003> (дата звернення: 04.09.2025).
- [5] OneSoil | Free App for Precision Farming. OneSoil | Free Farming App for Precision Agriculture. URL: <https://onesoil.ai/en> (дата звернення: 04.09.2025).
- [6] Cropin | SaaS-based AgTech | Smart Farming App | Agriculture Technology. URL: <https://www.cropin.com/> (дата звернення: 04.09.2025).
- [7] Проект протимінної діяльності в Україні. Проект протимінної діяльності в Україні. URL: <https://www.undp.org/uk/ukraine/projects/proyekt-protyminnoyi-diyalnosti-v-ukrayini> (дата звернення: 04.09.2025).
- [8] Методи комп'ютерного зору і глибинних нейронних мереж для еколого-економічного аналізу : монографія / Н. М. Куссуль та ін. Київ : Наук. думка, 2024. 474 с.
- [9] Temporal Fusion Transformers for interpretable multi-horizon time series forecasting / B. Lim та ін. *International Journal of Forecasting*. 2021. Т. 37, № 4. С. 1748–1764. URL: <https://doi.org/10.1016/j.ijforecast.2021.03.012> (дата звернення: 04.09.2025).
- [10] Aramendia A. I. The U-Net : A Complete Guide. Medium. URL: <https://medium.com/@alejandro.itoaramendia/decoding-the-u-net-a-complete-guide-810b1c6d56d8> (дата звернення: 26.05.2025).
- [11] Beck M., Pöppel K., Spanring M., Auer A., Prudnikova O., Kopp M., Klambauer G., Brandstetter J. «xLSTM: Extended Long Short-Term Memory». *NeurIPS 2024 Spotlight*, 2024. URL: <https://doi.org/10.48550/arXiv.2405.04517> (дата звернення: 04.09.2025).
- [12] Collect Earth Online Home. Collect Earth Online - Satellite Image Viewing & Interpretation Systema. URL: <https://www.collect.earth/> (дата звернення: 25.05.2025).
- [13] Silvanovych K., Hrynova O. Leveraging ai for agricultural land monitoring and reclamation. *Information Systems and Technology: Results and Prospects*, Kyiv. 2025. P. 295–297. URL: https://ist.fit.knu.ua/_files/ugd/016074_36d0f427916c46abb6491a7572bb63ec.pdf (дата звернення: 01.06.2025).
- [14] Microsoft (2024). ML.NET and ONNX Runtime for .NET developers. URL: <https://dotnet.microsoft.com/en-us/apps/ai/ml-dotnet> (дата звернення: 05.09.2025).
- [15] PyTorch Forecasting Documentation – pytorch-forecasting documentation. PyTorch Forecasting Documentation – pytorch-forecasting documentation. URL: <https://pytorch-forecasting.readthedocs.io/en/stable/> (дата звернення: 21.05.2025).
- [16] He K., Zhang X., Ren S., & Sun J. (2016). Deep Residual Learning for Image Recognition. *CVPR 2016*. URL: <https://ieeexplore.ieee.org/document/7780459> (дата звернення: 05.09.2025).
- [17] Сільванович К. В., Гриньова О. Є. Моніторинг та відновлення сільськогосподарських земель засобами штучного інтелекту. *Радіоелектроніка та молодь у XXI столітті*, м. Харків. 2025. С. 51–53. URL: <https://openarchive.nure.ua/entities/publication/6a9d7017-8bdb-4118-a8c8-8ed170ce91a8> (дата звернення: 01.06.2025).
- [18] Застосування генеративно-змагальних мереж для покращення якості сегментації супутникових знімків / О. В. Шкаліков та ін. XIX Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики», м. Київ. 2022. С. 375–378. URL: <https://ela.kpi.ua/handle/123456789/52532> (дата звернення: 03.09.2025).

Надійшла до редколегії 15.10.2025