



С.Г. Удовенко<sup>1</sup>, Л.Е. Чала<sup>2</sup>, О.Є. Гриньова<sup>3</sup>, Т.С. Яричкіна<sup>4</sup>

<sup>1</sup> доктор технічних наук, професор, завідувач кафедри інформатики та обчислювальної техніки ХНЕУ ім. С. Кузнеця,

м. Харків, Україна, serhiy.udovenko@hneu.net, ORCID iD: 0000-0001-5945-8647

<sup>2</sup> кандидат технічних наук, доцент, доцент кафедри штучного інтелекту ХНУРЕ, м. Харків, Україна, larysa.chala@nure.ua, ORCID iD: 0000-0002-9890-4790

<sup>3</sup> старший викладач кафедри штучного інтелекту ХНУРЕ, м. Харків, Україна, olena.hrynova@nure.ua, ORCID iD: 0000-0002-3367-8067

<sup>4</sup> студентка кафедри штучного інтелекту ХНУРЕ, м. Харків, Україна, tetiana.yarychkina@nure.ua, ORCID iD: 0000-0003-2841-630X

## ЕКОЛОГІЧНИЙ МОНІТОРИНГ ЛАНДШАФТНИХ ДІЛЯНОК З ВИКОРИСТАННЯМ РЕГУЛЯРИЗОВАНИХ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

Розглядається задача розпізнавання та аналізу картографічних зображень ландшафтних об'єктів з метою локалізації районів забруднення, моніторингу змін контурів окремих ділянок зображень під впливом природних і антропогенних факторів. Пропонується архітектура глибокої нейронної мережі, що використовує техніку регуляризації штучних нейронних мереж для запобігання її перенавантаженню. Наводяться результати побудови, навчання та застосування згорткової нейронної мережі з процедурою виключення Dropout для завдань екологічного ландшафтного моніторингу. Визначено перспективи використання запропонованого підходу для розпізнавання зображень проблемних ландшафтних ділянок.

ЕКОЛОГІЧНИЙ МОНІТОРИНГ, РОЗПІЗНАВАННЯ ЛАНДШАФТНИХ ЗОБРАЖЕНЬ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, МЕТОД ВИКЛЮЧЕННЯ

Рассматривается задача распознавания и анализа картографических изображений ландшафтных объектов с целью локализации районов загрязнения, мониторинга изменений контуров отдельных участков изображений под влиянием природных и антропогенных факторов. Предлагается архитектура глубокой нейронной сети, использующей технику регуляризации искусственных нейронных сетей для предотвращения ее переобучения. Приводятся результаты построения, обучения и применения сверточного нейронной сети с процедурой исключения Dropout для задач экологического ландшафтного мониторинга. Определены перспективы использования предложенного подхода для распознавания изображений проблемных ландшафтных участков.

ЭКОЛОГИЧЕСКИЙ МОНИТОРИНГ, РАСПОЗНАВАНИЕ ЛАНДШАФТНЫХ ИЗОБРАЖЕНИЙ, СВЕРТОВАЯ НЕЙРОННАЯ СЕТЬ, МЕТОД ИСКЛЮЧЕНИЯ

The problem of recognition and analysis of landscape objects cartographic images for localization of pollution areas, monitoring of changes of images contours under the influence of natural and anthropogenic factors is considered. The architecture of a deep neural network is proposed that uses the technique of regularization of artificial neural networks to prevent its overfitting. The results of the construction, training and application of a convolutional neural network with the Dropout elimination procedure for problems of ecological landscape monitoring are presented. The prospects of using the proposed approach for image recognition of problematic landscape areas are determined.

ECOLOGICAL MONITORING, LANDSCAPE IMAGE RECOGNITION, CONVOLUTIONAL NEURAL NETWORK, EXCLUSION METHOD

### Вступ

Останнім часом отримали розповсюдження цифрові технології екологічного моніторингу навколишнього середовища і природних ресурсів, що здійснюється за допомогою систем спостережень і подальшої обробки отриманих даних. Комп'ютерна обробка реалістичних картографічних зображень ландшафтних об'єктів в геоінформаційних системах (ГІС) екологічного моніторингу передбачає в загальному випадку реалізацію етапів попередньої цифрової обробки, розпізнавання та інтерпретації, як відзначається в роботах [1, 2]. Особливу практичну спрямованість мають завдання розпізнавання об'єктів за супутниковими знімками. До суттєвих труднощів такого розпізнавання слід віднести наявність різних внутрішніх та зовнішніх чинників (освітлення, орієнтація,

стан, наявність захмарених ділянок) що призводять до спотворення фрагментів зображення. Аналіз і подальша інтерпретація знімків є важливою частиною реалізації багатьох ГІС додатків (наприклад, локалізації районів забруднення, моніторинг зміни контурів окремих ділянок зображень під впливом природних і антропогенних факторів). Завданням обробки просторових даних в ГІС екологічного моніторингу природних ресурсів (зокрема, ландшафтних об'єктів) може бути поліпшення (відновлення) зображення за деяким критерієм, так і спеціальне перетворення зображень. Оброблювані зображення, як правило, спотворені шумами та викривленнями різної природи. Функціями пониження шумів в задачах обробки просторових даних в ГІС можуть бути як поліпшення візуального сприйняття аналізованих зображень, так і

збільшення чіткості при виділенні контурів об'єктів, що є необхідним для їх подальшого розпізнавання.

До найбільш актуальних завдань екологічного моніторингу навколишнього середовища і природних ресурсів за результатами супутникових спостережень слід, зокрема, віднести відстеження зміни ландшафту внаслідок планової або незаконної вирубки лісів та видобутку бурштину. В роботі [3] відзначається, що через надмірну вирубку лісів страждають не тільки самі дерева, але і інші рослини, що знаходяться на землі, тварини, для яких ця частина лісу була домом, а згодом і люди, так як через великі вирубки відбувається порушення рівня підземних вод, через що частішають випадки повеней. Однак навіть ліс, вирубаний незаконно, має шанс на відновлення, чого не можна сказати про ділянки, де після вирубки лісів несанкціоновано видобувається бурштин. Для отримання цього цінного ресурсу потрібно не тільки вирубати ліс, але також використовувати мотопомпи для підйому бурштину на поверхню, що призводить до вимивання ґрунту та заболочування навколишньої місцевості [4]. Таким чином, актуальним є запобігання несанкціонованих дій з вирубки лісів та видобутку бурштину, а також визначення тих місць, де видобуток відбувається вже тривалий час. Ефективними засобами здійснення такого запобігання можуть бути класифікація та розпізнавання зображень проблемних ландшафтних ділянок в ГІС екологічного моніторингу з використанням сучасних штучних нейронних мереж (ШНМ) [5], зокрема автоенкодерів [6].

У даній роботі пропонується архітектура глибокої нейронної мережі, що використовує техніку регуляризації загорткової ШНМ для запобігання її перенавчання, а також наводяться результати побудови, навчання та застосування такої мережі для завдань екологічного ландшафтного моніторингу.

### 1. Принципи регуляризації штучних нейронних мереж

Для розпізнавання зображень, отриманих за результатами супутникових спостережень, набули поширення технології, що використовують згорткові нейронні мережі (ЗНМ) – клас глибоких нейронних мереж (ГНМ) прямого поширення [7]. ЗНМ побудовані на використанні одного із різновидів багат шарових перцептронів, що мінімізують обсяг попередньої обробки вхідної інформації. Навчання ЗНМ відноситься до типу задач навчання «з учителем», де система навчається розпізнавати образи за допомогою різного роду адаптивних схем. Розпізнавання за такими схемами характеризується тим, що заздалегідь відома правильна класифікація кожного навчаючого образу. Базова архітектура ЗНМ полягає в послідовному використанні згорткових шарів, субдискретизуючих шарів і наявності повнозв'язних шарів на виході. При цьому застосовуються наступні парадигми: локальне сприйняття, розподілені ваги і субдискретизація. Локальне сприйняття полягає в тому, що на вхід одного нейрона подається не все

зображення, а лише деяка його частина. Концепція розподілених ваг припускає, що для великої кількості зв'язків використовується суттєво обмежений набір ваг. Субдискретизація полягає в зменшенні просторової розмірності зображення. Чергування шарів дозволяє складати карти таким чином, щоб на практиці забезпечити здатність розпізнавання складних ієрархій ознак.

Швидкість навчання і кількість нейронів у шарах ЗНМ визначаються зазвичай емпірично або з використанням апріорних даних щодо відомих протестованих мереж з аналогічною архітектурою. Швидкість навчання не є постійною величиною, а лінійно зменшується зі збільшенням числа ітерацій, впливаючи на швидкість зменшення величини корекції ваг. Визначення кількості нейронів у шарі є важливим етапом в застосуванні ЗНМ. На сьогодні не існує універсальних рекомендацій до вибору кількості прихованих шарів та кількості нейронів в них. Очевидно, що їх мала кількість не дозволяє мережі ефективно навчатися, а їх завелика кількість суттєво збільшує час навчання мережі або призводить до перенавчання мережі [8]. Перенавчання (англ. *overfitting*) мережі є однією з головних проблем використання ЗНМ [9] та полягає у тому, що створена модель починає ідентифікувати тільки приклади з навчальної вибірки, адаптуючись до них, але не вчиться класифікувати приклади, які не брали участі в навчанні. Це призводить до втрати мережею здатності узагальнення можливих класифікаційних ситуацій.

Для запобігання перенавчанню в ЗНМ використовують емпіричні та явні методи регуляризації, пов'язані насамперед зі введенням в мережу додаткової інформації. Найбільш поширеним емпіричним методом регуляризації є метод виключення (англ. *dropout*) [10]. Згідно з цим методом на кожному кроці тренування окремі вузли виключаються з мережі з імовірністю  $(1-p)$ , після чого залишається зменшена мережа, а вхідні та вихідні ребра виключених вузлів також усуваються. На наступному етапі на навчальних даних тренується лише зменшена мережа, після чого усунені вузли повторно вставляються до мережі з їхніми первинними вагами (рис. 1).

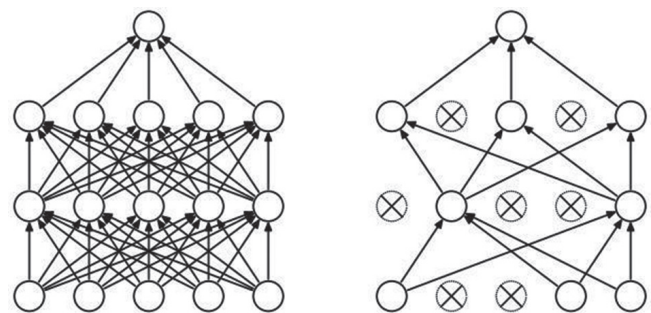


Рис. 1. Ілюстрація до процедури виключення за методом Dropout

Таким чином, ймовірність того, що після застосування процедури виключення нейрон залишиться в мережі, дорівнює  $q = 1 - p$ .

На етапах тренування ймовірність того, що прихований вузол буде виключено, зазвичай дорівнює 0.5, але для вхідних вузлів вона має бути набагато нижчою, адже при ігноруванні вхідних вузлів відбувається безпосередня втрата інформації.

Під час перевірки після завершення тренування в ідеалі бажано би знайти вибіркове середнє всіх можливих  $2^n$  мереж із виключеннями; на жаль, для великих значень  $n$  це є практично нездійсненним. Тим не менше, можна знайти задовільне наближення, використовуючи повну мережу, в якій виходи вузлів зважуються на коефіцієнт  $p$ , після чого математичне очікування значення виходу будь-якого вузла буде таким, як і на етапах тренування. Це є важливою перевагою методу виключення: хоча він породжує  $2^n$  нейронних мереж, але уможливорює поєднання моделей, а під час кожної перевірки розглядати необхідно лише одну мережу.

Таким чином метод виключення (dropout) дозволяє уникати необхідності тренування всіх вузлів на всіх тренувальних даних, що знижує прояв ефекту перенавчання. Цей метод також значно покращує швидкість тренування та дозволяє здійснювати практичне поєднання моделей навіть для згорткових нейронних мереж. Відзначимо, що процедура виключення послаблює взаємодії між вузлами та обмежує навчання урахуванням найбільш надійних ознак, що краще узагальнюється на нові класифікаційні дані. Це дозволяє замість навчання однієї згорткової нейронної мережі навчити набір з декількох мереж, після чого усереднити отримані результати. Нейрони, які були видалені, не впливають на процес навчання ні на одному з етапів алгоритму зворотного поширення помилки, через що виключення хоча б одного з нейронів рівносильно навчанню нової нейронної мережі. Якщо ймовірність виключення кожного нейрона однакова, то функція активації  $a(h)$  залежить від лінійної проекції вхідного  $d_i$  — вимірного вектора  $x$  на  $d_h$  — вимірний простір вихідних значень, тобто від аргументу наступного вигляду:

$$h(x) = xW + b, \quad (1)$$

де  $W$  — вектор ваг;  $b$  — настроюваний коефіцієнт.

Застосування техніки Dropout до даної проекції на етапі навчання можна представити як змінену функцію активації  $a(h)$ :

$$f(h) = D \odot a(h), \quad (2)$$

де  $D = (X_1, \dots, X_{d_h}) - d_h$  — мірний вектор випадкових величин  $X_i$ , що (згідно з законом Бернуллі) має наступний розподіл ймовірностей:

$$f(k; p) = \begin{cases} p, & \text{if } k = 1 \\ 1 - p, & \text{if } k = 0 \end{cases} \quad (3)$$

де  $k$  — можливі вихідні значення.

Відзначимо, що дана випадкова величина відповідає процедурі Dropout, яка застосовується до одного

нейрона. При цьому нейрон відключають з ймовірністю  $p = P(k=1)$ , а в зворотному випадку залишають включеним. Застосування Dropout до конкретного  $i$ -го нейрона можна представити таким чином:

$$O_i = X_i a \left( \sum_{k=1}^{d_i} w_k x_k + b \right) = \begin{cases} a \left( \sum_{k=1}^{d_i} w_k x_k + b \right), & \text{if } X_i = 1 \\ 0, & \text{if } X_i = 0 \end{cases} \quad (4)$$

де  $P(X_i = 0) = p$ .

З урахуванням того, що на етапі навчання нейрон залишається включеним в мережу з ймовірністю  $q$ , то на етапі тестування потрібно відтворити поведінку набору ЗНМ, який використовувався на попередньому етапі [9]. Для цього метод Dropout передбачає на етапі тестування множення функції активації на коефіцієнт  $q$ , тоді процедури навчання та тестування можна відповідно представити таким чином:

$$O_i = X_i a \left( \sum_{k=1}^{d_i} w_k x_k + b \right); \quad (5)$$

$$O_i = qa \left( \sum_{k=1}^{d_i} w_k x_k + b \right). \quad (6)$$

Узагальненням методу виключення є виключення з'єднань (англ. DropConnect), де кожне з'єднання, а не кожен вузол виходу, може бути виключено з ймовірністю  $1 - p$ . При цьому кожен вузол отримує вхід з випадкової підмножини вузлів попереднього шару. Виключення з'єднань є подібним до звичайного виключення тим, що воно вводить до моделі динамічну розрідженість, але відрізняється тим, що ймовірність визначається на вагах, а не на векторах виходу шару. Іншими словами, повноз'єднаний шар з виключенням з'єднань стає розріджено з'єднаним шаром, у якому з'єднання обираються випадково під час етапу тренування.

До явних методів регуляризації ШНМ відносять:

- запобігання перенавчанню шляхом обмеження кількості параметрів (зазвичай кількості прихованих вузлів у кожному шарі, або глибини мережі). Для ЗНМ на кількість параметрів впливає й розмір фільтру. Обмежування кількості параметрів безпосередньо обмежує передбачувальну потужність мережі, знижуючи складність функції, яку вона може виконувати на даних, і відтак обмежує розмір перенавчання;

- ослаблення ваг (англ. weight decay) шляхом додавання до похибки кожного вузла додаткової похибки, що пропорційна сумі ваг (норма L1) або квадратів величини (норма L2) вектора ваг. Рівень прийнятної складності моделі може бути знижено збільшенням

сталої пропорційності, що збільшує штраф за великі вектори ваг. L2-регуляризація (англ. L2 regularization) є найпоширенішим засобом явної регуляризації, що реалізується штрафуванням квадратного ступеня всіх параметрів безпосередньо в цілі. L2-регуляризація здійснює суттєве штрафування пікових вагових векторів та віддає перевагу розсіяним ваговим векторам. У зв'язку з багатократними взаємодіями між вагами та входами це реалізує властивість заохочення мережі використовувати всі її входи замість вибіркового використання лише деяких із них. L1-регуляризація (англ. L1 regularization) є іншим поширеним видом регуляризації, що веде вагові вектори до набуття розрідженості протягом оптимізації. Іншими словами, нейрони з L1-регуляризацією наприкінці процедури використовують лише розріджену підмножину їхніх найважливіших входів і стають практично інваріантними відносно зашумлених входів. Можливо також поєднувати L1 та L2-регуляризації, що відповідає варіанту еластично-сіткової регуляризації (англ. elastic net regularization);

– обмеження абсолютної вхньої межі величини вагового вектору для кожного нейрона і застосування методу проєкційного найшвидшого спуску для забезпечення цього обмеження. На практиці це відповідає попередньому уточненню параметрів з наступним забезпеченням обмеження для вагового вектору кожного нейрона [11].

Попередній аналіз свідчить про доцільність використання для моніторингу зображень ландшафтних ділянок, зокрема ділянок, де відбуваються незаконні дії з вирубки лісів та видобутку бурштину, ЗНМ з процедурою виключення Dropout (далі ця мережа позначається як ЗНМ-Д).

## 2. Побудова та програмна реалізація ЗНМ-Д для завдань екологічного моніторингу

Дані для такого завдання, як навчання нейронної мережі є вибіркою з досить великої кількості зображень. Це може складати кілька сотень тисяч зображень всього для одного з класів, щоб дозволити НМ формувати достатньо точні прогнози на даних, які не входять до навчальної вибірки.

Для деяких завдань задовільну вибірку можна знайти в інтернеті, де вона буде міститися в архіві і мати належний вигляд, проте це стосується далеко не всіх випадків, що відповідають реальним ситуаціям ландшафтного екологічного моніторингу. Зокрема, вибірки з зображеннями супутникових знімків, які б відображували вирубку лісу, а також стан територій, на яких відбувається або відбувався видобуток бурштину в інтернеті відсутні, тому для реалізації проекту побудови та тестування ЗНМ-Д з'явилася необхідність застосування існуючих сучасних додатків попередньої обробки супутникових зображень. Для створення відповідних наборів зображень в роботі

використовувався додаток «Google Earth Pro» (GEP). Особливість його використання полягає в тому, що будь-яку ділянку земної кулі можна побачити з супутника не тільки в останній момент, коли він був над цим місцем, але також в усі попередні рази, коли супутник передавав дані про конкретні території в додаток (рис. 2).

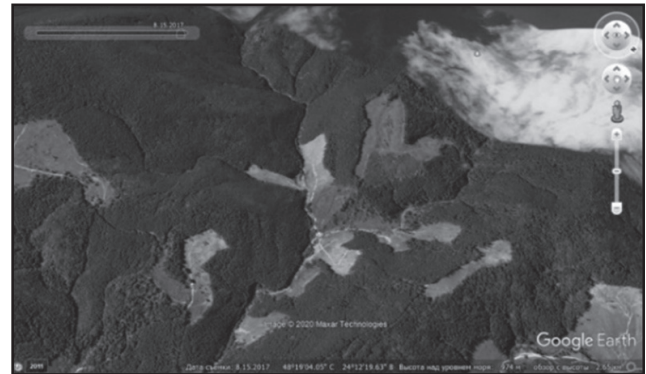


Рис. 2. Приклад зображення з ділянками вирубки лісу, отриманого з використанням «GEP»

Це є важливим для того, щоб використовувати знімки містили в собі, наприклад, саме результати вирубки лісу, а не будь-які інші роботи, через які відбулося знищення дерев. Зображення, отримані за допомогою додатку GEP, містять у верхньому лівому кутку часову шкалу, яка показує дату, коли був зроблений знімок поверхні, зображений на екрані. Аналіз таких зображень дозволяє відстежувати динаміку зміни стану лісових ділянок за будь-який відрізок часу, а послідовний набір цих зображень створює навчальну вибірку для використовуваної нейронної мережі. Результати досліджень, де наводяться дані щодо районів несанкціонованого видобутку бурштину, дозволили з використанням додатку GEP сформувати відповідну вибірку ландшафтних зображень. Приклад такого зображення (для одного з районів Рівненської області) наведено на рис. 3.

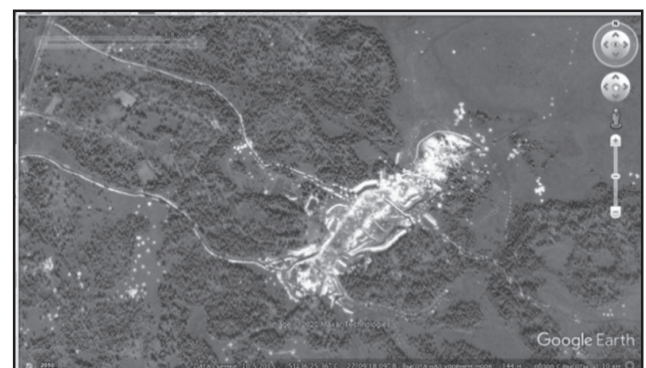


Рис. 3. Приклад зображення з ділянками несанкціонованого видобутку бурштину, отриманого з використанням «GEP»

Зображення ділянок, що передбачається аналізувати в системі екологічного ландшафтного моніторингу, можна віднести до таких чотирьох основних класів:

- клас 1: несанкціонована вирубка лісу;
- клас 2: несанкціонований видобуток бурштину;
- клас 3: хмари (захмареність зображень);
- клас 4: нормальний стан лісу (без вирубок).

Приклади зображень, що належать до класів 1 та 2 наведено вище на рис.2 та рис.3 відповідно.

На рис. 4 та рис. 5 представлено приклади зображень, що належать до класів 3 та 4.

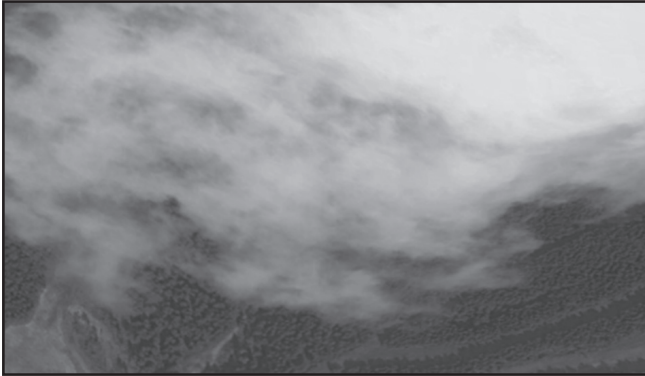


Рис. 4. Приклад захмареного зображення, отриманого з використанням «ГЕР»

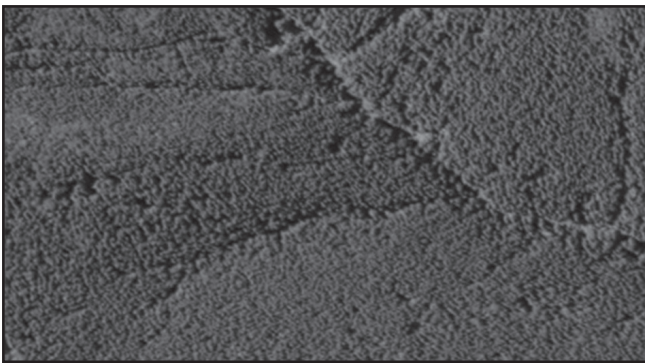


Рис. 5. Приклад зображення ділянки з нормальним станом лісу, отриманого з використанням «ГЕР»

Для створення вибірки примірників кожного з класів були використані супутникові знімки з програми «Google Earth Pro». Кожна категорія містила в собі 11 різних знімків ділянок площиною приблизно 1 кв.км. На рис. 6 наведено приклад структури зображення з цієї вибірки, що відноситься до класу «хмари». Номер зображення в частині вибірки, яка відноситься до класу «хмари», дорівнює 1 (перше зображення цього класу).



Рис. 6. Структура елемента вибірки (приклад)

Вся вибірка складається з файлів типу PNG. Всі зображення вибірки мають розмір 32×32 пікселі. Такий розмір відповідає умовам швидкого та ефективного навчання нейронної мережі. Подальші дії з зображеннями вибірки передбачають визначення каталогів, де будуть зберігатися відповідні дані.

Для каталогів з даними запропоновано таку структуру:

```
# Каталог з набором даних
data_dir = /train;
# Каталог з даними для навчання
train_dir = train;
# Каталог з даними для перевірки
val_dir = val;
# Каталог з даними для тестування
test_dir = test
# Частина набору даних для тестування
test_data_portion = 0.15;
# Частина набору даних для перевірки
val_data_portion = 0.15;
# Кількість елементів даних в одному класі
nb_images = 11.
```

Згідно з наведеною структурою у каталозі «train» зберігаються ті дані, на яких буде навчатися нейронна мережа. У каталозі «val» зберігаються дані для перевірки навченої НМ. У каталозі «test» знаходяться дані для тестування. Для зберігання даних конкретних класів необхідно створити підкаталоги (рис. 7).

```
def create_directory(dir_name):
    if os.path.exists(dir_name):
        shutil.rmtree(dir_name)
    os.makedirs(dir_name)
    os.makedirs(os.path.join(dir_name, "amber"))
    os.makedirs(os.path.join(dir_name, "clouds"))
    os.makedirs(os.path.join(dir_name, "deforestation"))
    os.makedirs(os.path.join(dir_name, "normal"))
create_directory(train_dir)
create_directory(val_dir)
create_directory(test_dir)
```

Рис. 7. Створення розділів для зберігання даних конкретного класу

Створені підкаталоги відповідають кожному з визначених вище класів:

- amber – клас, де зберігаються супутникові знімки із зображенням ділчок, де несанкціоновано видобувають бурштин;
- clouds – клас, де зберігаються супутникові знімки із зображенням захмарених ділянок;
- deforestation – клас, де зберігаються супутникові знімки із зображенням ділянок, де відбувалася несанкціонована вирубка лісу;
- normal – клас, де зберігаються супутникові знімки із зображенням територій, де ліс залишився в нормальному стані.

Наступним кроком є розподілення зображень по цим каталогам та підкаталогам (рис. 8).

```
def copy_images(start_index, end_index, source_dir, dest_dir):
    for i in range(start_index, end_index):
        shutil.copy2(os.path.join(source_dir, "amber." + str(i) + ".png"),
                    os.path.join(dest_dir, "amber"))
        shutil.copy2(os.path.join(source_dir, "clouds." + str(i) + ".png"),
                    os.path.join(dest_dir, "clouds"))
        shutil.copy2(os.path.join(source_dir, "deforestation." + str(i) + ".png"),
                    os.path.join(dest_dir, "deforestation"))
        shutil.copy2(os.path.join(source_dir, "normal." + str(i) + ".png"),
                    os.path.join(dest_dir, "normal"))
    start_val_data_idx = int(nb_images * (1 - val_data_portion - test_data_portion))
    start_test_data_idx = int(nb_images * (1 - test_data_portion))
    copy_images(1, start_val_data_idx, data_dir, train_dir)
    copy_images(start_val_data_idx, start_test_data_idx, data_dir, val_dir)
    copy_images(start_test_data_idx, nb_images, data_dir, test_dir)
```

Рис. 8. Розподіл зображень по каталогах та підкаталогах

Фрагмент коду, наведений на рис. 8, відповідає за розподіл загального набору зображень по каталогах та підкаталогах, які були створені. У підсумку формуються дані, підготовлені до того, щоб навчати, перевіряти і тестувати на них нейронну мережу.

Графічне представлення мережі ЗНМ-Д, побудованої для реалізації поставлених завдань, наведено на рис. 9.

Структура запропонованої ЗНМ, що наведена на рис. 9, містить наступні елементи (зліва направо):

- шар згортки (розмір ядра 3x3, кількість карт ознак – 32 шт., функція активації ReLU);
- шар підвибірки (вибір максимального значення з квадрата 2x2);
- шар згортки (розмір ядра 3x3, кількість карт ознак – 32 шт., функція активації ReLU);
- шар підвибірки (вибір максимального значення з квадрата 2x2);
- шар згортки (розмір ядра 3x3, кількість карт ознак – 64 шт., функція активації ReLU);
- шар підвибірки (вибір максимального значення з квадрата 2x2);
- шар перетворення з двовимірного представлення в одновимірне;
- повнозв’язний шар (64 нейрони, функція активації ReLU);
- шар Dropout;
- вихідний шар (4 нейрони, функція активації sigmoid).

Дана нейронна мережа дозволяє класифікувати зображення ландшафтних ділянок розміру 32x32 пікселя по 4 визначеним вище класам.

Для активації усіх шарів згортки ЗНМ-Д, а також для повнозв’язного шару, обрана функція ReLU, що повертає  $X$ , якщо  $X$  позитивно, а в зворотному випадку повертає 0:

$$A(x) = \max(0, x). \tag{7}$$

Функція ReLU нелінійна за своєю природою, а набори шарів, створені з її застосуванням, теж нелінійні. Діапазон значень ReLU знаходиться між 0 і  $+\infty$ . ReLU дозволяє уникати щільної активації, яка є дуже витратною. Завдяки створенню розрідженої активації, ReLU дозволяє вирішити поставлене завдання при менших витратах. Слід однак відзначити, що функція ReLU частково представлена горизонтальною лінією, а при негативних значеннях  $X$  ця частина має від’ємне значення градієнта. Через це відсутнім є коригування ваги частки, а це, в свою чергу, призводить до того, що нейрони, які перебувають в цьому стані, не реагують на зміни у вхідних даних, що робить частину мережі пасивною.

Втім функція ReLU підходить для створення шарів ЗНМ завдяки тому, що використовує більш прості математичні операції, ніж, наприклад, сигмоїда або гіперболічний тангенс [11].

Для вихідного шару, що містить 4 нейрони, кожен з яких відповідає одному з класів, доцільно використовувати функцію активації sigmoid, що має такий вигляд:

$$A = \frac{1}{1 + e^{-x}}. \tag{8}$$

Важливими перевагами цієї функції є те, що вона не бінарна та має гладкий градієнт. Сигмоїда добре підходить для вихідного шару класифікаційних нейронних мереж, так як вона прагне привести значення до однієї зі сторін кривої. Це дозволяє виділити чіткі межі при отриманні відповіді з приводу віднесення об’єкта до того чи іншого класу.

Також однією з переваг даної функції активації є те, що її значення лежить в діапазоні від 0 до 1 (наприклад, лінійна функція може приймати значення від  $-\infty$  до  $+\infty$ ), що дозволяє уникнути помилок, пов’язаних з великими значеннями активації.

Програмну реалізацію розглянутої структури ЗНМ-Д було здійснено з використанням мови Python та середовища розробки Colab від Google, що дало можливість не тільки зручно формувати НМ, але й додавати допоміжні коментарі в якості заголовків, які дозволяють підкреслити призначення різних частин коду.

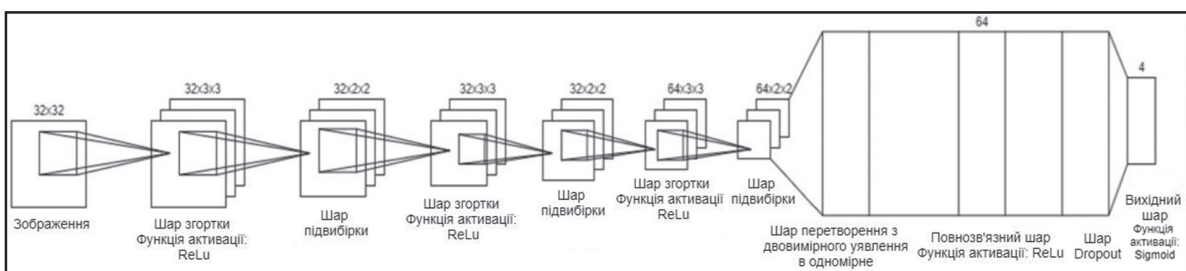


Рис. 9. Структура ЗНМ-Д

На початковому етапі створення ЗНМ-Д були визначені змінні, які в подальшому мають передаватися як параметри (для створення НМ, її навчання, перевірки та тестування) в процедури і функції, а саме:

```
# Розміри зображення
img_width, img_height = 32, 32;
# Розмірність тензору на основі зображення для
вхідних даних ЗНМ-Д
backend Tensorflow, channels_last
input_shape = (img_width, img_height, 3);
# Кількість епох
epochs = 30;
# Розмір мінівибірки
batch_size = 4;
# Кількість класів
num_classes = 4;
# Кількість зображень для навчання
nb_train_samples = 24;
# Кількість зображень для перевірки
nb_validation_samples = 12;
# Кількість зображень для тестування
nb_test_samples = 12
```

Крім змінних, що визначають каталоги з даними, які мають використовуватися для навчання, перевірки та тестування, задаються: розміри зображення (в даному випадку це зображення розміром 32x32 пікселя); розмірність тензора на основі зображення для вхідних даних ЗНМ-Д, де вказуються розміри зображення та кількість каналів. Далі задається кількість епох навчання нейронної мережі на одному і тому ж (навчальному) наборі даних. Крім цього вказуються розмір міні-вибірки, кількість класів і розміри кожної з вибірок.

Після того, як всі параметри визначені, створюються та компілюються шари ЗНМ-Д (рис. 10).

```
model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=input_shape))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(5))
model.add(Activation('sigmoid'))
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

Рис. 10. Компіляція ЗНМ-Д

Слід зауважити, що для створення ЗНМ-Д використовувався фреймворк TensorFlow (TF) з високорівневим API-інтерфейсом Keras, що забезпечує можливість ітеративного налагодження моделі [12].

Для завантаження зображень і їх перетворення було створено генератор зображень (рис. 11).

```
datagen = ImageDataGenerator(rescale=1. / 255)
```

Рис. 11. Створення генератора зображень

Відзначимо, що кожен піксель зображення ділиться на 255, щоб інтенсивність кожного пікселя була в діапазоні від 0 до 1. Це набагато спрощує роботу з зображенням і виключає появу занадто великих чисел і помилок, пов'язаних з цим.

Для навчання на основі зображень з каталогу було створено генератор даних, де зберігаються зображення для навчання (рис. 12). В цьому генераторі використано метод `flow_from_directory`, який завантажує зображення з відповідного каталогу. Так як розмір міні-вибірки дорівнює 4, то кожен раз при зверненні до каталогу «train» ця функція буде завантажувати 4 зображення розміром 32x32 і готувати для них відповідні передбачення, які вказуються в назві кожного з файлів-зображень типу PNG.

```
train_generator = datagen.flow_from_directory(
    train_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical')
```

Рис. 12. Генератор даних для навчання на основі зображень з каталогу генератора зображень

Далі було створено генератор даних для перевірки на основі зображень з каталогу. Цей генератор працює за аналогією з попереднім (за методом `flow_from_directory`), а його основною відмінністю є те, що зображення завантажуються з каталогу «val», де зберігаються дані для перевірки (рис. 13).

Слід відзначити, що в останньому аргументі `class_mode` вказано значення «categorical», так як зображення можуть бути розподілені по декількох класах.

```
val_generator = datagen.flow_from_directory(
    val_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical')
```

Рис. 13. Генератор даних для перевірки на основі зображень з каталогу генератора зображень

Далі було створено генератор даних для тестування на основі зображень з каталогу «test», де зберігаються зображення для тестування (рис. 14).

```
test_generator = datagen.flow_from_directory(
    test_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical')
```

**Рис. 14.** Генератор даних для тестування на основі зображень з каталогу генератора зображень

Наступною дією створення ЗНМ-Д було її навчання за допомогою даних, які надають генератори, з використанням методу `fit_generator`. Основою цього методу є генератор, що використовується для навчання, де в аргументі `steps_per_epoch` задається кількість проходів, яке дорівнює загальній кількості зображень, поділений на розмір міні-вибірки (тобто на 4). Далі задається кількість епох навчання ЗНМ-Д на одному і тому ж наборі даних, що впливає на точність прогнозів, що аналізується на етапі тестування. Наступним аргументом методу `fit_generator` є генератор для перевірки, який був створений раніше. Для нього так само потрібно вказати кількість проходів, яке дорівнює кількості зображень в папці «val», поділений на число зображень міні-вибірки (рис. 15).

```
model.fit_generator(
    train_generator,
    steps_per_epoch=nb_train_samples // batch_size,
    epochs=epochs,
    validation_data=val_generator,
    validation_steps=nb_validation_samples // batch_size)
```

**Рис. 15.** Навчання ЗНМ-Д

Таким чином було побудовано згорткову НМ, здатну розподілити зображення ландшафтних ділянок за чотирма визначеними вище класами.

### 3. Тестування та перспективи розвитку ЗНМ-Д для завдань екологічного моніторингу

Після створення ЗНМ-Д було здійснено аналіз ефективності її навчання на 30 епохах. В табл. 1 наведено результати точності навчання та перевірки досліджуваної мережі в залежності від номера епохи (прийняті позначення: ВН – втрата навчання; ТН – точність навчання, ВП – втрата перевірки; ТП – точність перевірки).

Згідно з результатами аналізу (табл. 1) максимальна точність на навчальній вибірці доходить до 79,17%, в той час як мінімальна втрата становить 66,30%. На перевіркою вибірці максимальна точність досягає 83,33%, а мінімальна втрата – 54,71%.

Результати можна вважати задовільними тому, що вибірки були малі, так як потрібні вибірки в мережі відсутні, а створення навіть невеликої вибірки всього на 11 примірників вичагає чимало часу.

Після цього була розрахована точність нейронної мережі для тестової вибірки:

```
scores = model.evaluate_generator(test_generator,
    nb_test_sample // bath_size)
print («Точність на тестових даних: %. 2f%%»
(scores ) [1] *100)).
```

Слід відзначити, що на тестовій вибірці точність прогнозу показала 91.67%, що є цілком задовільним результатом для згорткових регуляризованих нейронних мереж (з урахуванням малої кількості елементів вибірки).

Відповідь нейронної мережі представлено на зображенні в лівій верхній частині екрана (рис. 16). Завдяки використанню функції активації ReLu, можна визначити ймовірність належності довідповідного класу значення між 0 і 1. Значення, найближче до одиниці, буде означати клас, до якого належить зображення. В даному прикладі зображення належить до класу «Вирубка лісу» («deforestation») і можна помітити, що ЗНМ-Д визначила це згідно з пошуком для цього класу найбільшої ймовірності.

Це підтверджують результати перевірки ЗНМ-Д на навчальній вибірці, приклад якої наведено на рис. 17.

**Таблиця 1**

Точність ЗНМ-Д в залежності від епох навчання

№ епохи	Час, мс	ВН	ТН	ВП	ТП
1	2	3	4	5	6
1	28	1.61	0.25	1.59	0.25
2	14	1.58	0.33	1.56	0.25
3	14	1.55	0.37	1.51	0.33
4	13	1.47	0.25	1.42	0.33
5	14	1.46	0.17	1.36	0.25
6	15	1.24	0.45	1.33	0.25
7	16	1.60	0.21	1.32	0.33
8	13	1.32	0.37	1.32	0.25
9	13	1.25	0.46	1.30	0.42
10	13	1.36	0.46	1.29	0.50
11	14	1.19	0.46	1.24	0.42
12	14	1.27	0.29	1.18	0.50
13	14	1.29	0.50	1.11	0.67
14	14	1.16	0.37	1.05	0.67
15	14	1.09	0.54	0.94	0.67
16	14	0.94	0.67	0.89	0.83
17	15	1.04	0.50	0.78	0.75
18	14	0.88	0.62	0.85	0.75
19	14	0.83	0.54	0.95	0.50
20	13	0.83	0.62	0.79	0.75
21	14	0.76	0.67	0.64	0.75
22	14	0.97	0.62	0.68	0.67
23	14	0.89	0.67	1.03	0.67
24	14	1.22	0.54	0.80	0.83
25	14	0.76	0.58	0.77	0.58
26	14	0.85	0.46	0.73	0.75
27	14	0.78	0.54	0.66	0.75
28	13	0.66	0.71	0.63	0.83
29	14	0.66	0.79	0.57	0.83
30	14	0.76	0.71	0.55	0.83



На рис. 16 наведено приклад розпізнавання мережею ЗНМ-Д зображення ландшафтної ділянки, отриманого з використанням GEP.

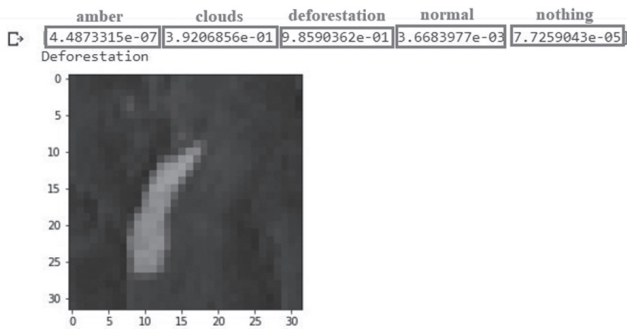


Рис. 16. Приклад розпізнавання зображення ландшафтної ділянки

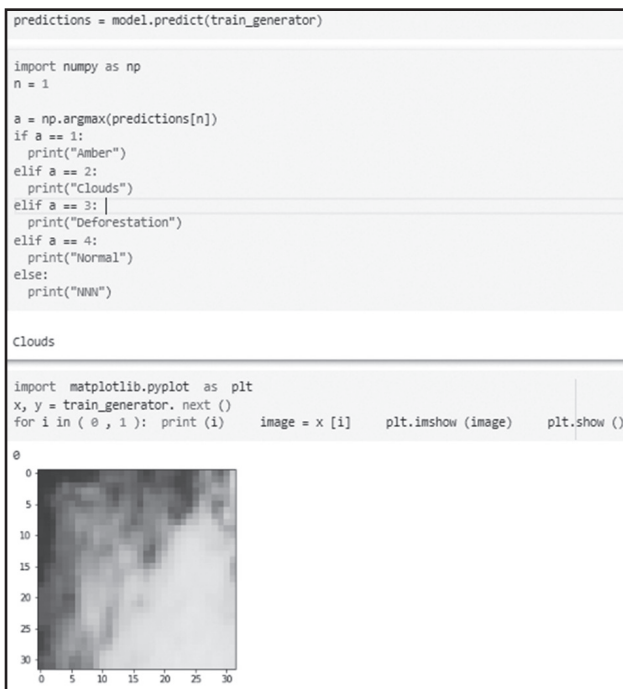


Рис. 17. Розпізнавання зображення, що належить до навчальної вибірки

Аналіз зображень з навчальної вибірки здійснювалося в два етапи, перший з яких формує відповідь ЗНМ-Д, а другий надає зображення, на яке було отримано цю відповідь. З прикладу, наведеному на рис. 17, випливає, що програма коректно віднесла аналізоване зображення до класу «хмари».

Інші результати проведеного тестування запропонованої системи розпізнавання зображень ландшафтних ділянок також підтвердили її працездатність.

Перспективним продовженням досліджень є удосконалення характеристик запропонованої ЗНМ-Д шляхом застосування комбінованих засобів регуляризації мережі та сучасних графічних прискорювачів. Крім того, можуть бути розглянуті інші завдання екологічного моніторингу ландшафтних об'єктів (наприклад, моніторинг лісових пожеж та визначення хворих ділянок лісу), де доцільно впровадження запропонованої ЗНМ-Д.

## Висновки

До важливих завдань екологічного моніторингу за результатами супутникових спостережень слід, зокрема, віднести відстеження зміни ландшафту внаслідок вирубки лісів та видобутку бурштину. Перспективним напрямом вирішення цих завдань є застосування згорткових нейронних мереж (ЗНМ). Однією з головних проблем використання ЗНМ є перенавчання мережі, яке призводить до втрати мережею здатності узагальнення можливих класифікаційних ситуацій. У статті здійснено аналіз методів регуляризації, що сприяють запобіганню ефекту перенавчання в ЗНМ, та визначено переваги методу виключення (dropout). Запропоновано архітектуру ЗНМ, що використовує метод виключення для запобігання її перенавчання. Наводяться результати побудови, навчання та застосування такої мережі для завдань екологічного ландшафтного моніторингу. Детально розглянуто етапи побудови генераторів зображень та компіляції елементів ЗНМ. Побудову ЗНМ здійснено з частковим використанням бібліотеки Keras для мови програмування Python. Зображення ділянок, що передбачається аналізувати в системі екологічного ландшафтного моніторингу, пропонується віднести до чотирьох основних класів (несанкціонована вирубка лісу; несанкціонований видобуток бурштину; захмареність зображень; нормальний стан лісу (без вирубок)). Для формування наборів зображень в роботі використовувався додаток «Google Earth Pro».

Результати проведеного тестування запропонованої системи розпізнавання зображень ландшафтних ділянок підтвердили її працездатність (зокрема, на тестовій вибірці точність прогнозу дорівнює 91.67%, що є цілком задовільним результатом для згорткових регуляризованих нейронних мереж з урахуванням малої кількості елементів вибірки).

Розроблена система розпізнавання зображень ландшафтних ділянок є гнучкою та розширюваною, що робить можливим її подальше удосконалення у напрямку підвищення точності розпізнавання та розширення функцій.

## Список літератури:

- [1] Зацерковний В. І. Геоінформаційні системи і бази даних: монографія. – Кн. 2 / В. І. Зацерковний, В. Г. Бурачек, О. О. Железняк, А. О. Терещенко. – Ніжин : НДУ ім. М. Гоголя, 2017. – 237 с.
- [2] Луньова О.В. Методологічні засади застосування орто-трансформованих космічних знімків для оцінки стану навколишнього середовища та програмне забезпечення досліджень / О.В. Луньова, О.В. Буглак // Український журнал дистанційного зондування Землі – 2019. № 22. – С. 27–34.
- [3] Бондар О.І. та ін. Моніторинг стану навколишнього середовища засобами ГІС / О.І. Бондар та ін. // К. : ДЕА (Державна екологічна академія), 2018. – 52 с.

- [4] *Hnatushenko V.V.* Satellite Monitoring of Consequences of Illegal Extraction of Amber in Ukraine / V.V. Hnatushenko, D.K. Mozgovyi, V.V. Vasyliiev, O.O. Kavats // Scientific bulletin of National Mining University. State Higher Educational Institution «National Mining University». Dnipropetrovsk. – 2017. No 2 (158). – P. 99–105.
- [5] *Schmidhuber J.* Deep learning in neural networks: An overview / J. Schmidhuber // *Neural networks*. – 2015. – V. 61. – P. 85–117.
- [6] *Bezsonov O.* Processing of noisy digital images with use of evolving autoencoders / O. Bezsonov, O. Rudenko, S. Udovenko, O. Dudinova // *Eastern-European Journal of Enterprise Technologies*. – 6/9(90). – 2017. – P. 63–69.
- [7] *McCann M.* Convolutional neural networks for inverse problems in imaging / M. McCann, K. Jin., M Unser // *A review //IEEE Signal Processing Magazine*. – 2017. – V. 34. – № 6. – P. 85–95.
- [8] *Sikorskiy O.* Convolutional neural networks in image classification / O. Sikorskiy // *Information Innovative Technologies* – 2017. – №1. – P. 397–401.
- [9] *Gatys L.* Image style transfer using convolutional neural networks / L. Gatys., A. Ecker, M. Bethge // *Proceedings of the IEEE conference on computer vision and pattern recognition*. – 2016. – P. 2414–2423.
- [10] *Srivastava N.* Dropout: A Simple Way to Prevent Neural Networks from Overfitting / N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov // *Journal of Machine Learning Research*. – 15(56). – 2014 – P. 1929–1958.
- [11] *Miikkulainen R.* et al. Evolving deep neural networks // *Artificial Intelligence in the Age of Neural Networks and Brain Computing*. – Academic Press, 2019. – P. 293–312.
- [12] *Білашенко С.В.* Розпізнавання зображень за допомогою згорткових нейронних мереж з використанням бібліотеки Keras / С.В. Білашенко, Н. Н. Шаповалова, О.Г. Рибальченко // *Гірничий вісник*. – 2018. – Вип. 103. – С. 148–154.

*Надійшла до редколегії 18.03.2020*