

УДК 004.8:624

DOI: 10.30838/J.BPSACEA.2312.250423.65.933

## ІДЕНТИФІКАЦІЯ ТРІЩИН У ВЕРТИКАЛЬНИХ ЕЛЕМЕНТАХ БУДІВЕЛЬНИХ КОНСТРУКЦІЙ ЗА ДОПОМОГОЮ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

САВИЦЬКИЙ М. В.<sup>1</sup>, *докт. техн. наук, проф.*,  
ДАНИШЕВСЬКИЙ В. В.<sup>2\*</sup>, *докт. техн. наук, проф.*,  
ГАЙДАР А. М.<sup>3</sup>, *канд. техн. наук, доц.*

<sup>1</sup> Кафедра залізобетонних і кам'яних конструкцій, Придніпровська державна академія будівництва та архітектури, вул. Архітектора Олега Петрова, 24-а, 49005, Дніпро, Україна, тел. +38 (056) 756-33-00, e-mail: [ms@pdaba.edu.ua](mailto:ms@pdaba.edu.ua), ORCID ID: 0000-0003-4515-2457

<sup>2\*</sup> Кафедра будівельної і теоретичної механіки та опору матеріалів, Придніпровська державна академія будівництва та архітектури, вул. Архітектора Олега Петрова, 24-а, 49005, Дніпро, Україна, тел. +38 (056) 756-33-73, e-mail: [vladyslav.danishevskyy@pdaba.edu.ua](mailto:vladyslav.danishevskyy@pdaba.edu.ua), ORCID ID: 0000-0002-3049-4721

<sup>3</sup> Кафедра технології будівельного виробництва, Придніпровська державна академія будівництва та архітектури, вул. Архітектора Олега Петрова, 24-а, 49005, Дніпро, Україна, тел. +38 (056) 756-34-76, e-mail: [nastuel\\_gaidar@pdaba.edu.ua](mailto:nastuel_gaidar@pdaba.edu.ua), ORCID ID: 0000-0001-8993-1458

**Анотація. Постановка проблеми.** Актуальною проблемою сучасного будівництва стало розроблення нових методів ідентифікації дефектів будівельних конструкцій, які б заощаджували людські ресурси та зменшували залежність результатів обстеження від суб'єктивних людських чинників. **Мета роботи** полягає у розробленні штучних нейронних мереж для виявлення тріщин у вертикальних елементах будівельних конструкцій (наприкладі бетонних та залізобетонних стін) та їх класифікації за напрямом. **Методика.** Хмарний інструмент Teachable Machine застосовувався для створення нейронної мережі із закритою внутрішньою архітектурою. Використання бібліотек програмної платформи TensorFlow дозволило розробити згорткову нейронну мережу з оригінальною архітектурою. Програмний код написано мовою Python. Для створення та навчання моделі використовувалося хмарне середовище розробки програмного забезпечення Colab. **Наукова новизна.** Обчислено нові моделі штучних нейронних мереж для ідентифікації дефектів будівельних конструкцій. Визначено раціональні значення параметрів навчання та раціональну архітектуру згорткової нейронної мережі, які дозволяють досягти найбільшої точності та найменших втрат моделі під час розпізнавання та класифікації тріщин. **Практична значимість.** Розроблені моделі нейронних мереж та одержані за їх допомогою результати свідчать про високу ефективність застосування методів штучного інтелекту для вирішення проблем діагностики будівельних конструкцій. На відміну від традиційних способів обстеження, запропоновані моделі дозволяють проводити діагностику автоматично в режимі реального часу шляхом аналізу фотографічних зображень, зображень з веб-камери комп'ютера, смартфона чи квадрокоптера. Останнє дає можливість виконувати обстеження споруд без фізичної присутності спеціалістів на об'єкті, що особливо важливо під час роботи в небезпечних місцях: на висоті, в аварійних та частково зруйнованих будівлях, на замінованих територіях тощо. **Висновки.** Запропоновані у роботі методи розроблення нейронних мереж у подальшому можуть бути розвинуті для діагностики та класифікації широкого кола дефектів будівельних конструкцій і споруд.

**Ключові слова:** *діагностика будівель і споруд; дефекти будівельних конструкцій; тріщини; нейронні мережі; комп'ютерний зір; Teachable Machine; TensorFlow*

## IDENTIFICATION OF CRACKS IN VERTICAL ELEMENTS OF BUILDING STRUCTURES USING CONVOLUTIONAL NEURAL NETWORKS

SAVYTSKYI M.V.<sup>1</sup>, *Dr. Sc. (Tech.), Prof.*,  
DANISHEVSKYY V.V.<sup>2\*</sup>, *Dr. Sc. (Tech.), Prof.*,  
GAIDAR A.M.<sup>3</sup>, *Cand. Sc. (Tech.), Assoc. Prof.*

<sup>1</sup> Department of Reinforced Concrete and Masonry Structures, Prydniprovsk State Academy of Civil Engineering and Architecture, 24-a, Architect Oleh Petrov St., Dnipro, 49005, Ukraine, tel. +38 (056) 756-33-00, e-mail: [ms@pdaba.edu.ua](mailto:ms@pdaba.edu.ua), ORCID ID: 0000-0003-4515-2457

<sup>2\*</sup> Department of Structural and Theoretical Mechanics and Strength of Materials, Prydniprovsk State Academy of Civil Engineering and Architecture, 24-a, Architect Oleh Petrov St., Dnipro, 49005, Ukraine, tel. +38 (056) 756-33-73, e-mail: [vladyslav.danishevskyy@pdaba.edu.ua](mailto:vladyslav.danishevskyy@pdaba.edu.ua), ORCID ID: 0000-0002-3049-4721

<sup>3</sup> Department of Construction Technologies, Prydniprovsk State Academy of Civil Engineering and Architecture, 24-a, Architect Oleh Petrov St., Dnipro, 49005, Ukraine, tel. +38 (056) 756-34-76, e-mail: [nastuel.gaidar@pdaba.edu.ua](mailto:nastuel.gaidar@pdaba.edu.ua), ORCID ID: 0000-0001-8993-1458

**Abstract. Problem statement.** The relevant issue of modern Civil Engineering is the development of new methods for identifying defects in building structures, which would save human resources and reduce the dependence of survey results on subjective human factors. **The purpose of the research** is to develop artificial neural networks for the identification and classification of cracks in vertical elements of building structures (e. g., concrete and reinforced concrete walls). **Methodology.** The cloud tool Teachable Machine is used to develop a neural network with a pre-defined internal architecture. The libraries of TensorFlow software platform allow us to develop a convolutional neural network with a tunable architecture. The program code is written in Python. The training was performed using the cloud environment Colab. **Scientific novelty.** New models of artificial neural networks for the identification of defects in building structures are developed. The rational magnitudes of the training parameters and the topology of the convolutional neural network are determined allowing to achieve the highest accuracy and the lowest losses of the model. **Practical value.** The developed models of neural networks and the results obtained with their help ensures a high efficiency of the artificial intelligence to solve problems of the health monitoring of building structures. Unlike traditional approaches, the proposed models allow real-time automatic diagnostics by analyzing photographic images, images from computer, smartphone or quadcopter webcams. The latter makes it possible to inspect buildings without the physical presence of humans at the site, which is especially important for working in dangerous places, such as tall buildings, partially destroyed buildings, mined areas, etc. **Conclusions.** The proposed methods can be further extended for the monitoring and classification of a wide range of defects in building structures.

**Keywords:** structural health monitoring; building defects; cracks; neural networks; computer vision; Teachable Machine; TensorFlow

**Актуальність теми.** Щорічно у світі витрачаються мільйони доларів на технічну діагностику будівель і споруд. Стихійні лиха, такі як повені та землетруси, а також чисельні негативні техногенні впливи завдають серйозної шкоди будівельним конструкціям. Надзвичайної актуальності проблема діагностики будівель і споруд набула після агресії РФ в Україну, яка призвела до масштабних пошкоджень і руйнувань промислових об'єктів, житлового фонду та об'єктів інфраструктури, таких як дороги, мости, тунелі тощо.

Одні з визначальних параметрів технічного стану будівельних конструкцій – це дефекти і пошкодження у вигляді тріщин. Вони з'являються в результаті дії різних факторів і причин. Для визначення технічного стану конструкції оцінити ступінь небезпеки тріщин можливо лише на основі вивчення місця їх розташування, моменту їх виникнення, характеру і ширини розкриття й інших важливих факторів.

У праці [1] запропонована така класифікація тріщин в елементах будівельних конструкцій:

1. Момент виникнення (на стадіях життєвого циклу споруди): на стадії виготовлення, монтажу, експлуатації, знесення.

2. Види за походженням: усадкові, температурні, деформаційні, силові.

3. Причини утворення:

– усадка бетону;

– теплова обробка, дія високих технологічних температур, сезонні коливання температур, нерівномірне охолодження;

– нерівномірні осідання фундаментів або опор, тиск продуктів корозії, набухання матеріалу;

– дія навантажень: транспортних, монтажних, завищених, динамічних.

4. Характерні особливості:

– стабілізовані, нестабілізовані;

– наскрізні, односторонні;

– подовжні, горизонтальні, вертикальні, поперечні;

– одиночні, що перетинаються, паралельні.

5. За розмірами (шириною): волосяні (до 0.1 мм), дрібні (до 0.3 мм), розвинуті

(0.3–0.5 мм), великі (до 1 мм), значні (понад 1 мм).

6. За ступенем небезпеки: безпечні; безпечні, проте знижують довговічність; небезпечні.

Операції з технічної діагностики передбачають візуальний огляд будівельних конструкцій, оцінення їх стану та спроможності забезпечувати необхідну несну здатність та функціональну придатність. Виявлення та оцінення дефектів та пошкоджень конструкцій – це часто дуже трудомістке завдання, якщо виконується шляхом безпосереднього візуального огляду або за допомогою класичних методів діагностики [1–4]. Також існує ймовірність, що через суб'єктивні людські фактори деякі дефекти і пошкодження можуть залишитися непоміченими.

Багато об'єктів критичної інфраструктури потребують регулярних перевірок, що вимагає значних людських ресурсів. Такі роботи можуть бути небезпечними для здоров'я та життя спеціалістів, наприклад, коли вони здійснюються у складних для доступу місцях, на висоті, у частково зруйнованих та аварійних будівлях. Таким чином, існує нагальна потреба автоматизації процесів діагностики будівель і споруд та розроблення нових методів ідентифікації дефектів та пошкоджень будівельних конструкцій, які б заощаджували людські ресурси та зменшували залежність результатів обстеження від суб'єктивних факторів.

**Аналіз публікацій.** Останніми роками в усіх галузях науки і техніки спостерігається швидкий розвиток методів штучного інтелекту, одним із поширених видів якого стали штучні нейронні мережі (ШНМ). Моделюючи роботу людського мозку, після відповідного навчання ШНМ здатні ефективно розв'язувати задачі щодо розпізнавання, класифікації та аналізу різних видів даних та інформації.

Застосуванню ШНМ для виявлення та оцінення дефектів будівельних конструкцій присвячено велику кількість робіт. Так, Perez, Tah, Mosavi [5] за допомогою ШНМ VGG-16 [6] розробили методику діагностики

ушкоджень, спричинених дією вологи. Rajadurai, Kang [7] адаптували ШНМ AlexNet [8] для виявлення тріщин у бетонних поверхнях, а Dorafshan та ін. [9] – для обстеження мостів та будинків із використанням безпілотних літальних апаратів. Chun, Yamane, Maemura [10] за допомогою ШНМ та методів глибокого навчання розробили систему «машинного зору» для діагностики технічного стану мостів. Детальний огляд нових досліджень в цій галузі наведено в статті Munawar та ін. [11].

**Мета роботи** полягає у розробленні ШНМ для виявлення тріщин у вертикальних елементах будівельних конструкцій (на прикладі бетонних та залізобетонних стін) та їх класифікації за напрямом (вертикальні чи горизонтальні). Розв'язання даної задачі ускладнюється тим, що наскрізні тріщини часто можуть бути візуально схожими на дефекти поверхні. Зображення конструкцій, що досліджуються, можуть сильно відрізнятися залежно від фактури поверхонь, фарби, інтенсивності світла, кута фотозйомки тощо. Тріщини також можуть мати нерегулярний характер. Зазначені фактори викликають суттєві труднощі під час навчання та тестування моделей ШНМ.

### 1. Початковий набір зображень.

Для навчання та тестування ШНМ ми використали колекцію цифрових фотографій SDNET2018 [12]. Колекція складається з 56 092 зображень елементів будівельних конструкцій з тріщинами та без них, зроблених фотокамерою Nikon із роздільною здатністю матриці 16 мегапікселів. Для формування колекції використано 230 будівельних об'єктів, які належать до трьох типів: дороги (104 об'єкти), стіни (72 об'єкти) та мости (54 об'єкти). Зображення зменшені до розміру 256 на 256 пікселів і мають три кольорові канали з 256 рівнями яскравості на кожний (24-бітний колір).

Розглянемо модель ШНМ, яка буде діагностувати тріщини у стінах та класифікувати їх як вертикальні чи горизонтальні. Відібраний початковий набір зображень містить 1 086 фотографій, рівномірно поділених на три групи:

вертикальні тріщини, горизонтальні тріщини та неушкоджені конструкції. Відповідно, кожна група складається з 352 фотографій.

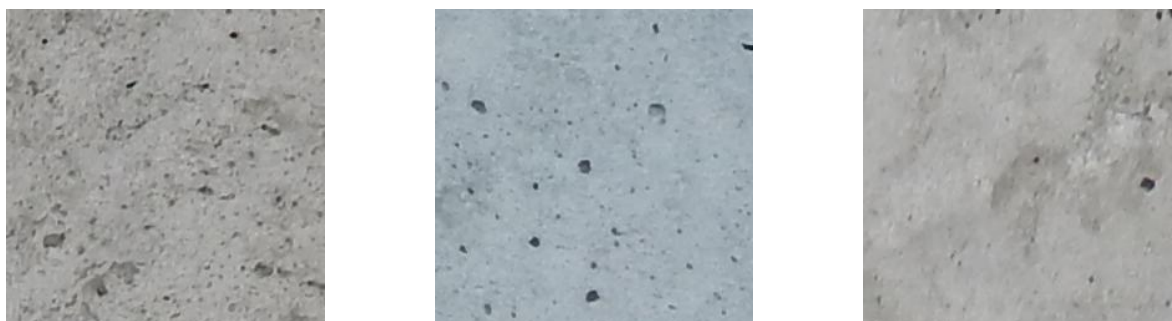
Приклади використаних зображень наведено на рисунках 1–3.



*Рис. 1. Вертикальні тріщини*



*Рис. 2. Горизонтальні тріщини*



*Рис. 3. Неушкоджені конструкції без тріщин*

У подальшому початковий набір зображень поділено на дві групи; 80–85 % зображень (навчальні зразки) використовуються для навчання ШНМ, тобто для встановлення оптимальних зв'язків між нейронами, які дозволяють моделі приймати правильні рішення щодо класифікації даних. Решта 15–20 % зображень (тестові зразки) не беруть участі у процесі навчання, а використовуються для тестування моделі та перевірки, наскільки ефективно вона може

обробляти дані, з якими зустрічається вперше.

## **2. Розробка ШНМ за допомогою Teachable Machine.**

Teachable Machine [13] – безкоштовний хмарний інструмент компанії Google, який дозволяє легко створювати моделі машинного навчання. Взаємодія з інструментом здійснюється через веб-інтерфейс. За допомогою Teachable Machine можна навчити ШНМ розпізнавати зображення, звуки та пози. Робота Teachable

Machine основана на використанні TensorFlow.js [14] – бібліотеки для машинного навчання JavaScript.

Робота з інструментом зводиться до наступних кроків:

1. Вибір типу проекту: класифікація зображень, звуків або поз. Залежно від цього можуть бути застосовані різні архітектури ШНМ. Зауважимо, що Teachable Machine не дозволяє користувачу самостійно змінювати чи редагувати внутрішню архітектуру ШНМ.

2. Збір початкових даних. Можна завантажувати файли з комп'ютера, записувати зображення та пози через веб-камеру, записувати звуки через мікрофон.

3. Навчання моделі та її тестування. Перевірка, наскільки правильно ШНМ розпізнає та класифікує нові приклади.

4. Експорт моделі. Розроблену модель ШНМ можна зберегти на своєму комп'ютері та використовувати для створення власних програмних додатків або розмістити в Інтернеті.

Необхідно зауважити, що розроблена модель може не завжди працювати так, як потрібно. Навчання нейронної мережі – евристична процедура, ефективність якої залежить від великої кількості параметрів. У більшості випадків оптимальні значення цих параметрів неможливо передбачити наперед і вони мають визначатися експериментальним шляхом для кожної окремої задачі.

Для розроблення моделі використовуємо набір зображень, описаний у розділі 1. Під час навчання ШНМ Teachable Machine дозволяє користувачу задавати значення таких параметрів:

Параметр Epochs (епохи) визначає тривалість навчання моделі. Одна епоха значить, що модель вивчила кожен навчальний зразок один раз. Наприклад, якщо задано Epochs = 50, то у процесі навчання вихідний набір даних буде опрацьовано 50 разів. У цілому, чим більше значення даного параметра, тим краще модель навчиться приймати рішення.

Параметр Batch Size визначає розмір пакета. Пакет – це набір зразків, які використовуються впродовж однієї ітерації у

процесі навчання. Наприклад, задано Batch Size = 16, а вихідний набір навчальних зразків містить 80 зображень. Тоді усі дані буде поділено на 5 пакетів ( $80/16 = 5$ ). Як тільки модель ознайомиться з усіма 5 пакетами, завершиться одна епоха.

Коефіцієнт Learning Rate визначає швидкість навчання моделі,  $0 \leq \text{Learning Rate} \leq 1$ . Він дозволяє керувати величиною корекції вагових коефіцієнтів нейронів на кожній ітерації. Великі значення Learning Rate > 0.7 відповідають великим крокам корекції.

При цьому модель навчається швидше, але також зростають помилки навчання. Малі значення Learning Rate < 0.1 відповідають малим крокам корекції вагових коефіцієнтів. Це збільшує тривалість навчання, але дозволяє досягти кращої точності роботи моделі.

Як зазначалося вище, на практиці оптимальні показники навчальних параметрів визначаються експериментальним шляхом для кожної окремої задачі. На рисунках 4–6 наведено результати навчання запропонованої моделі для різних розмірів пакета Batch Size. При цьому було прийнято Epochs = 50 та Learning Rate = 0.001.

Функція Accuracy (точність) – це частка правильних рішень, прийнятих моделлю. Так, якщо модель правильно розпізнає та класифікує 90 зразків зі 100, то її точність дорівнює  $90/100 = 0.9$ .

Функція Loss (втрати) дозволяє оцінити, наскільки достовірно модель навчилася приймати рішення. Якщо модель дає правильний прогноз із ймовірністю 100 %, втрата дорівнює нулю. В інших випадках втрата буде більшою.

Наприклад, нехай модель А правильно класифікувала зразок із ймовірністю 60 %, а модель Б – із ймовірністю 90 %. Незважаючи на те, що точність моделей однакова, значення втрати у моделі Б буде меншим.

На рисунках 4 та 5 сині криві відповідають навчальним зразкам, а помаранчеві криві – тестовим зразкам.

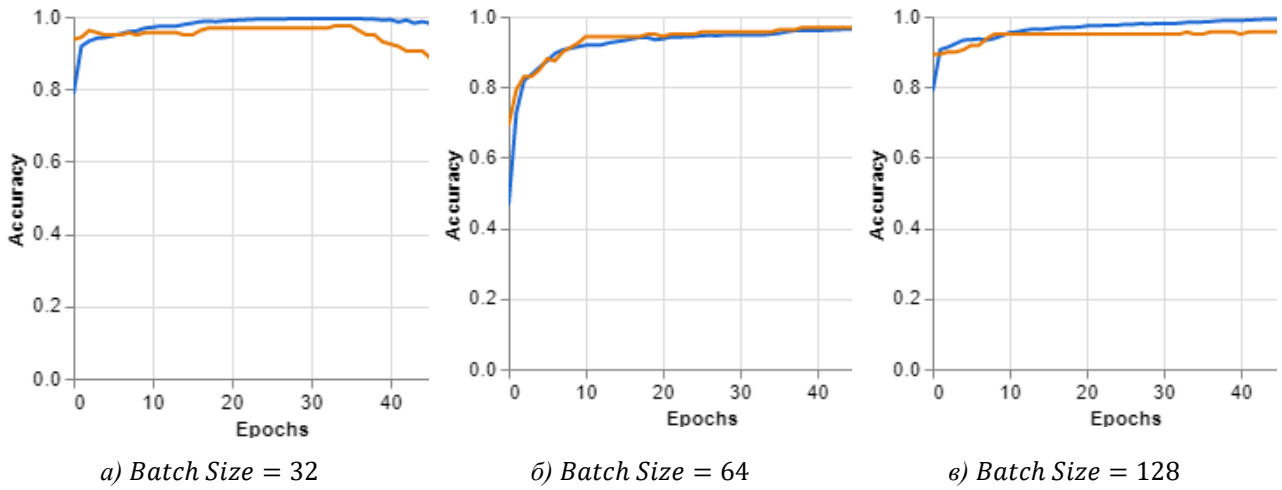


Рис. 4. Точність моделі

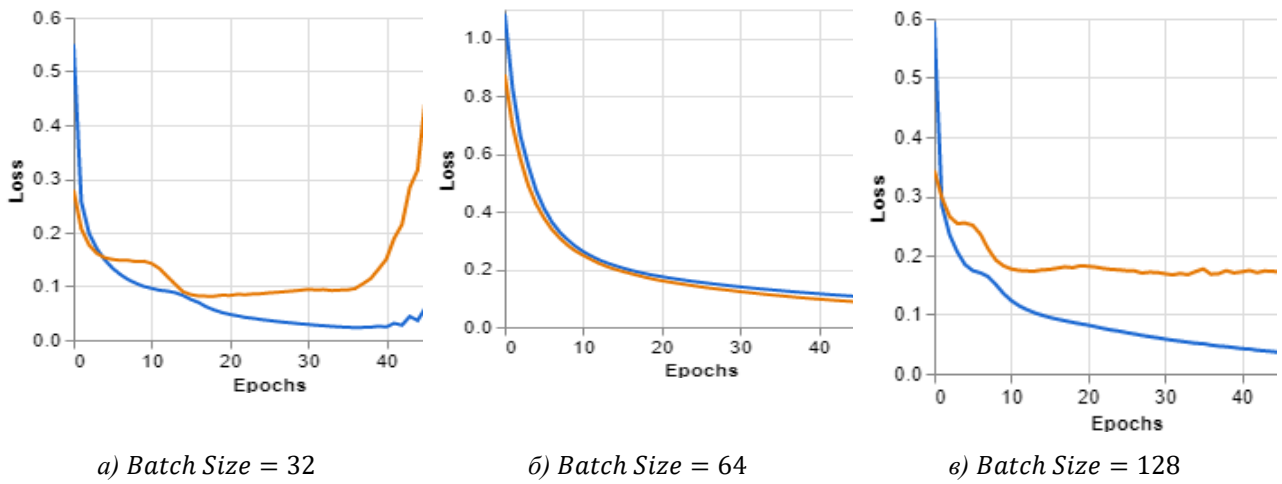


Рис. 5. Втрати моделі

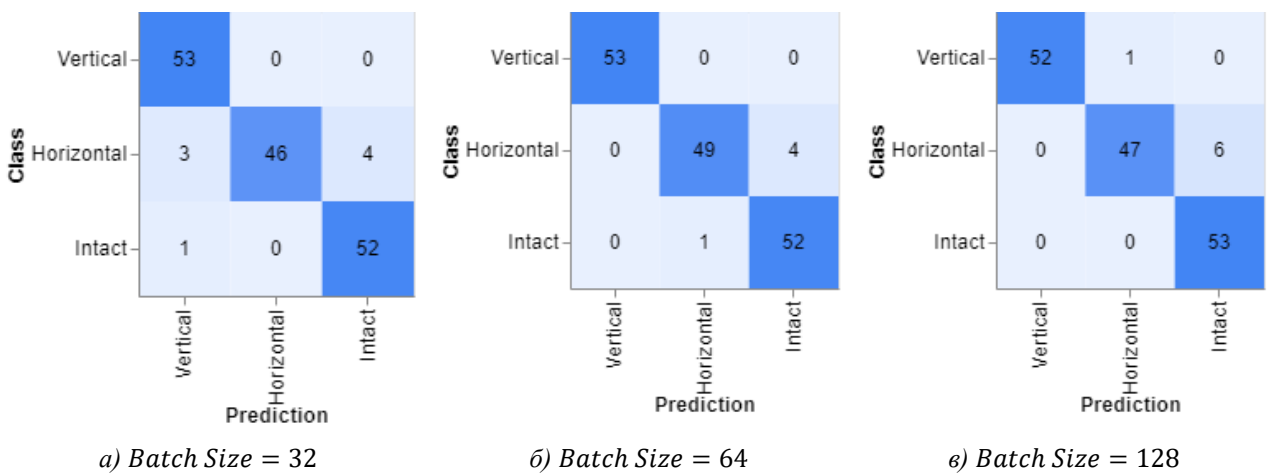


Рис. 6. Матриця помилок

У матриці помилок наведено дані про те, яку кількість тестових зразків розпізнано правильно та в яких класах допущено помилки.

Наприклад, у випадку Batch Size = 32 (рис. 6 а) модель правильно розпізнала 53 вертикальні тріщини, 46 горизонтальних тріщин та 52 неушкоджені стіни. Але при цьому 3 горизонтальні тріщини та 1 неушкоджена стіна інтерпретовані як вертикальні тріщини, а фотографії 4 горизонтальних тріщин інтерпретовано як зображення стін без тріщин.

У розробленні та навчанні нейронних мереж однією з фундаментальних проблем стало перенавчання моделі. Стан перенавчання виникає, коли точність відносно тестових зразків виявляється меншою, а втрати – більшими, ніж точність та втрати відносно навчальних зразків. Це означає, що модель правильно класифікує навчальні дані, але не може точно розпізнати дані, з якими зустрічається вперше. Перенавчання – це результат надмірного прилаштування нейронних зв'язків моделі до інформації, що міститься в навчальному наборі даних. Це означає, що модель прилаштувалася не до загальних, головних ознак зображення (у нашому випадку – до наявності та орієнтації тріщин), а до другорядних деталей (фактура стіни, колір, поверхневі сколи тощо). Така модель стає неспроможною виявляти загальні ознаки і не може застосовуватися на практиці, навіть якщо демонструє високу точність відносно навчальних зразків.

Для запобігання перенавчанню необхідно підібрати оптимальну архітектуру нейронної мережі (зокрема, кількість шарів та нейронів), а також за необхідності видалити частину нейронних зв'язків, зменшивши тим самим кількість вагових коефіцієнтів моделі. Також на ефективність роботи нейронної мережі суттєво впливають значення навчальних параметрів.

Аналіз даних, наведених на рисунках 5, 6 свідчить, що при Batch Size = 32 та Batch Size = 128 модель наближається до стану перенавчання. Натомість найкращі результати досягнуті при Batch Size = 64. В останньому випадку точність і втрати моделі відносно навчальних та тестових зразків дуже близькі (табл. 1). Матриця помилок також свідчить про найменшу

кількість хибних прогнозів при Batch Size = 64.

Таблиця 1

#### Точність та втрати моделі при Batch Size = 64

	Точність	Втрати
Навчальні зразки	0.970	0.099
Тестові зразки	0.971	0.081

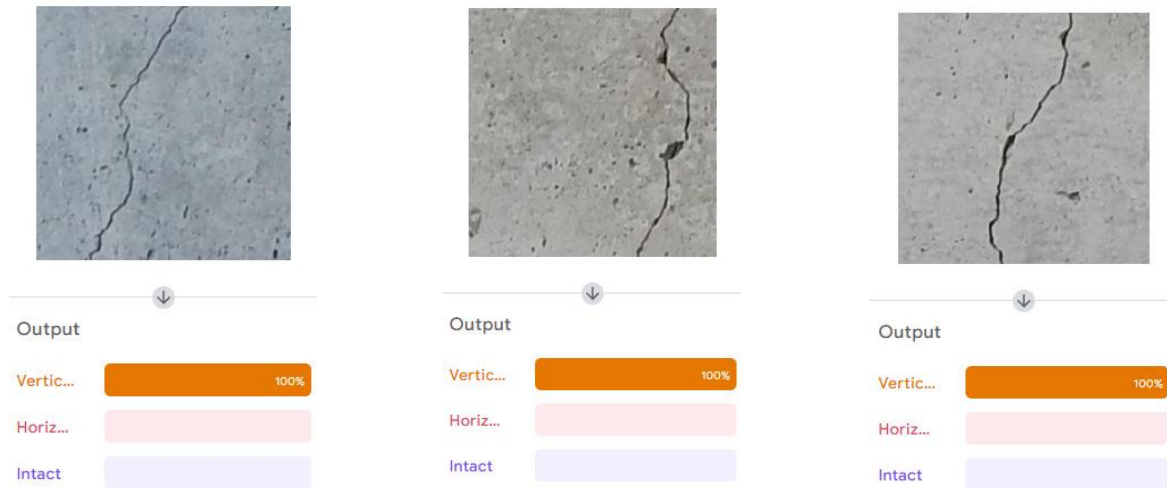
Розроблена модель розміщена в Інтернеті і доступна за посиланням: <https://teachablemachine.withgoogle.com/models/6RsGrbES2/>.

На рисунку 7 наведено результати роботи моделі при класифікації зображень, які не використовувались у процесі навчання. Декілька прикладів неточних прогнозів показано на рисунку 8. Можна зробити загальний висновок при високу практичну точність розробленої ШНМ.

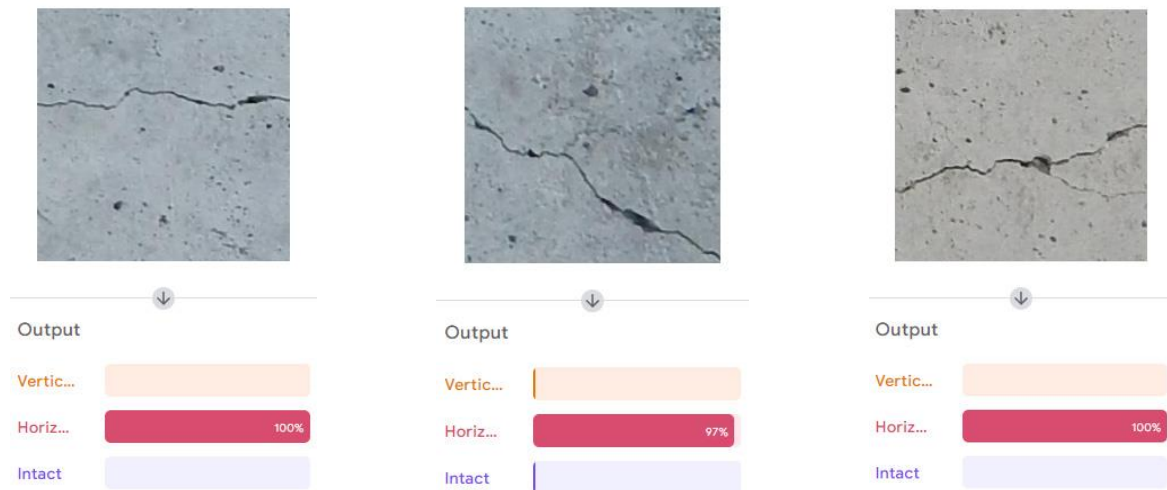
### 3. Розроблення ШНМ за допомогою TensorFlow.

Суттєвим обмеженням інструменту Teachable Machine є неможливість змінювати внутрішню архітектуру ШНМ, яка задається автоматично залежно від типу задачі. Цей розділ присвячено розробленню ШНМ зі своєю власною архітектурою для класифікації тріщин у бетонних стінах. Для цього використовується відкрита програмна платформа TensorFlow [15], розроблена компанією Google для створення та навчання ШНМ.

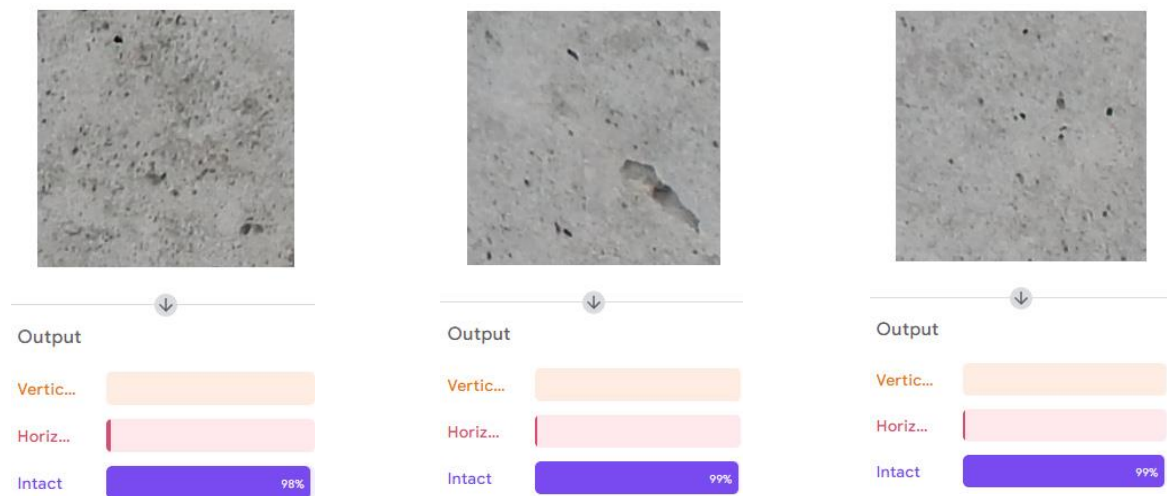
TensorFlow – це наскрізна платформа машинного навчання з відкритим вихідним кодом. Вона має комплексну та гнучку екосистему інструментів, бібліотек та ресурсів, яка дозволяє дослідникам упроваджувати найсучасніші технології машинного навчання, а розробникам – легко створювати та розгортати програми на основі штучного інтелекту. Основний програмний інтерфейс (API) TensorFlow реалізовано для мови програмування Python за допомогою бібліотеки Keras. Зазначимо, що також існують API для інших мов: R, C Sharp, C++, Haskell, Java, Go та Swift. У цій роботі використовується API для Python. Основні переваги TensorFlow зводяться до наступного.



*а) вертикальні тріщини*



*б) горизонтальні тріщини*



*в) неушкоджені стіни*

*Рис. 7. Приклади роботи моделі (Teachable Machine) та ступінь ймовірності прийнятих рішень*



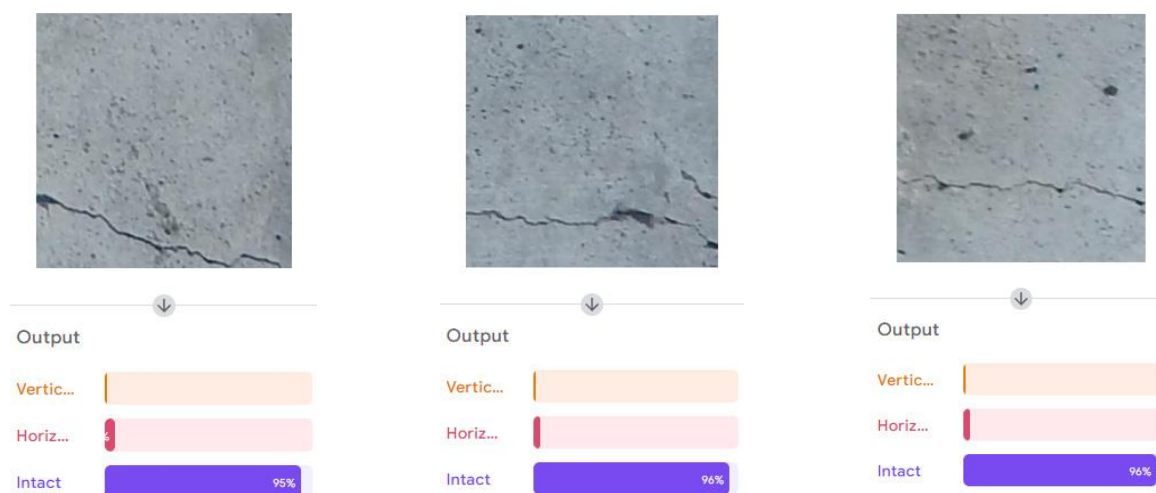


Рис. 8. Приклади неправильної класифікації

*Проста побудова моделей.* TensorFlow пропонує кілька рівнів абстракції, тому розробник може вибрати той, що відповідає його індивідуальним потребам. Для взаємодії з TensorFlow використовується API високого рівня Keras, який суттєво спрощує початок роботи з машинним навчанням. Для більшої гнучкості, режим Eager Execution (активне виконання) дозволяє негайно бачити результати виконання програмного коду та легко його налагоджувати. Для великих проектів можна використовувати API паралельних обчислень для виконання процесу машинного навчання на різних конфігураціях обладнання без зміни визначеної моделі.

*Надійність машинного навчання у будь-якому місці.* TensorFlow забезпечує прямий шлях виробництва: на серверах, периферійних пристроях або в Інтернеті. TensorFlow дозволяє легко навчити та розгорнути модель незалежно від того, які мови або пристрої використовуються. Платформа TensorFlow Extended призначена для створення великих виробничих середовищ машинного навчання. Бібліотека TensorFlow.js робить можливим навчання та розгортання нейронних мереж у браузері. Бібліотека TensorFlow Lite забезпечує роботу моделей машинного навчання на мобільних пристроях, мікроконтролерах та різних периферійних приладах.

*Простір для експериментальних досліджень.* TensorFlow надає широкі

можливості для створення та навчання нових моделей без обмежень у швидкості та продуктивності. Функції Keras Functional API та Model Subclass API забезпечують високу гнучкість і глибокий контроль під час розроблення ШНМ зі складною архітектурою. TensorFlow також підтримує екосистему потужних бібліотек надбудов і моделей для експериментів, включаючи Ragged Tensors, TensorFlow Probability, Tensor2Tensor і BERT.

Сьогодні провідні компанії світу широко застосовують TensorFlow для розв'язання своїх прикладних задач у різних галузях (Airbnb, Airbus, Coca-Cola, PayPal, Spotify, Twitter та багато інших).

Для створення та навчання ШНМ на Python з використанням бібліотек TensorFlow у нашій роботі використовується хмарне середовище Colaboratory (скорочено Colab) [16], розроблене науково-дослідним підрозділом Google Research. Colab дозволяє писати та виконувати програми Python через веб-інтерфейс браузера. З технічної точки зору, Colab являє собою хмарний хостинг для блокнотів Jupyter [17]. Розроблений програмний код виконується на віртуальних машинах із потужними графічними процесорами. Для науково-дослідницьких робіт та освітніх цілей використання Colab безкоштовне.

Оптимальна архітектура ШНМ має визначатися залежно від особливостей кожної конкретної задачі та початкового набору даних. Для задачі розпізнавання та

класифікації тріщин у бетонних стінах ми провели серію експериментів і знайшли наступну раціональну архітектуру мережі, яка дозволяє досягнути високої точності та мінімізувати втрати моделі.

Запропонована модель являє собою багат шарову згорткову ШНМ із послідовним з'єднанням шарів. Архітектуру

ШНМ наведено у таблиці 2. Перший шар є вхідним шаром зображення з розміром  $256 \times 256 \times 3$ , де  $256 \times 256$  – розмір у пікселях зображень, що аналізуються, а 3 – кількість кольорових каналів для одного пікселя. У цьому шарі також здійснюється нормалізація вхідних даних, що приводить усі зображення до одного рівня яскравості.

Таблиця 2

Архітектура ШНМ

Номер шару	Тип шару	Розмір даних на виході	Кількість параметрів
1	Input Rescaling	256, 256, 3	0
2	Convolution 2D	254, 254, 32	896
3	MaxPooling 2D	127, 127, 32	0
4	Dropout	127, 127, 32	0
5	Convolution 2D	125, 125, 32	9 248
6	MaxPooling 2D	62, 62, 32	0
7	Dropout	62, 62, 32	0
8	Convolution 2D	60, 60, 32	9 248
9	MaxPooling 2D	30, 30, 32	0
10	Dropout	30, 30, 32	0
11	Convolution 2D	28, 28, 32	9 248
12	MaxPooling 2D	14, 14, 32	0
13	Dropout	14, 14, 32	0
14	Convolution 2D	12, 12, 32	9 248
15	MaxPooling 2D	6, 6, 32	0
16	Dropout	6, 6, 32	0
17	Convolution 2D	4, 4, 32	9 248
18	MaxPooling 2D	2, 2, 32	0
19	Dropout	2, 2, 32	0
20	Flatten	128	0
21	Dense	128	16 512
22	Dense	32	4 128
23	Dense	3	99

Далі послідовно розташовано 6 блоків, кожен з яких включає:

- шар 2D згортки `tf.keras.layers.Conv2D`, який виявляє головні ознаки зображення;
- шар субдискретизації `tf.keras.layers.MaxPooling2D`, який зменшує розміри попереднього шару шляхом ущільнення зображення та відкидання другорядних деталей;
- шар виключення `tf.keras.layers.Dropout`, який випадковим чином виключає з роботи певну частку (у нашому проєкті 20 %) нейронів

попереднього шару. Це зменшує загальну кількість вагових коефіцієнтів та запобігає перенавчанню моделі.

Після цього шар `tf.keras.layers.Flatten` зменшує розмірність даних, конвертуючи 2D-матрицю, одержану на вході, у 1D-вектор на виході.

Далі розташовано три звичайні повнозв'язані нейронні шари `tf.keras.layers.Dense`, які здійснюють класифікацію одержаних даних. Розмір цих шарів поступово зменшується. Останній з них має розмір 3, що дорівнює кількості

класів, які розпізнає модель (вертикальні тріщини, горизонтальні тріщини, неушкоджені стіни). Загальна кількість параметрів моделі, показники яких визначаються в процесі навчання, складає 67 875.

Важливим параметром ШНМ є функція активації. Вона визначає, яким буде значення сигналу нейрона на виході залежно від середньозваженої суми сигналів на його вході. У цій роботі для кожного шару використовується нелінійна функція ReLU виду  $f(x) = \max(0, x)$ .

Навчання моделі виконувалось для наступних значень параметрів: Epochs = 50, Batch Size = 64, Learning Rate = 0.001. Точність та втрати моделі наведено на рисунках 9 та 10. Сині криві відповідають навчальним зразкам, а помаранчеві – тестовим зразкам. Одержані результати навчання свідчать про високу точність та низькі втрати моделі, що підтверджує здатність правильно розпізнавати дані, з якими модель зустрічається вперше.

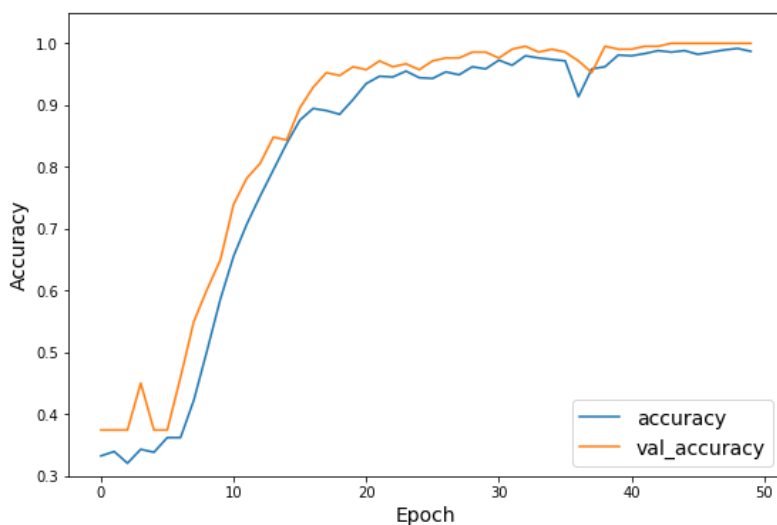


Рис. 9. Точність моделі

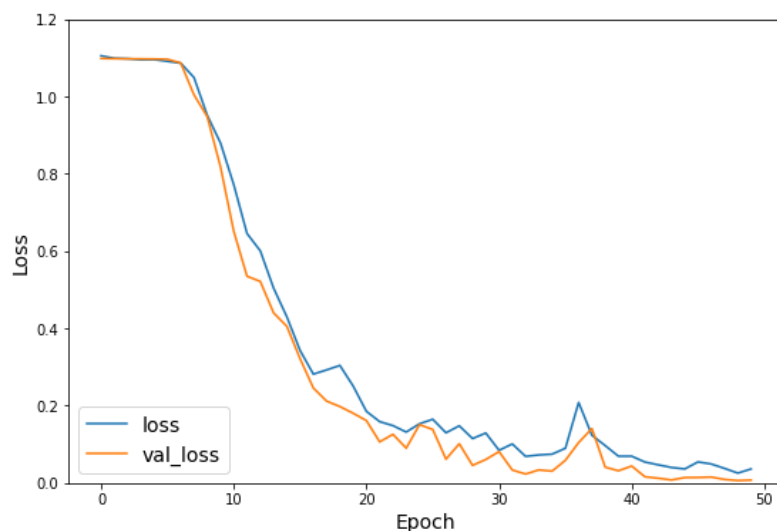


Рис. 10. Втрати моделі

На рисунці 11 наведено приклади роботи моделі для класифікації зображень, які не використовувались у процесі навчання. Тут розглянуто ті самі фотографії, що були задіяні під час тестування роботи моделі,

створеної за допомогою Teachable Machine. У даних прикладах ШНМ, розроблена за допомогою TensorFlow, демонструє приблизно таку саму високу точність, як і модель Teachable Machine.



vertical with a 100.0 percent confidence



vertical with a 100.0 percent confidence



vertical with a 100.0 percent confidence

*а) вертикальні тріщини*



horizontal with a 100.0 percent confidence



horizontal with a 88.68 percent confidence



horizontal with a 100.0 percent confidence

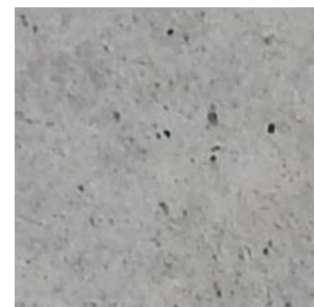
*б) горизонтальні тріщини*



intact with a 99.59 percent confidence



intact with a 99.84 percent confidence



intact with a 99.94 percent confidence

*в) неушкоджені стіни*

*Рис. 11. Приклади роботи моделі (TensorFlow) та ступінь ймовірності прийнятих рішень*



horizontal with a 99.02 percent confidence



horizontal with a 100.0 percent confidence



horizontal with a 100.0 percent confidence

*Рис. 12. Правильна класифікація зображень, які не були розпізнані моделлю Teachable Machine*

Тепер розглянемо приклади, у яких модель Teachable Machine припустилася помилок (див. рис. 8). Результати роботи моделі, розробленої за допомогою TensorFlow, наведено на рисунці 12. Можна спостерігати, що дана модель правильно та з високою точністю класифікувала зображення, які не змогла розпізнати модель Teachable Machine. Це дозволяє зробити висновок, що індивідуальний підбір раціональної архітектури ШНМ для кожної окремої задачі дозволяє досягти суттєво кращих результатів, ніж використання загальних універсальних рішень.

### Висновки

Розроблено дві моделі ШНМ для виявлення та класифікації тріщин у вертикальних елементах будівельних конструкцій (наприкладі бетонних та залізобетонних стін). Для цього використовувались хмарний інструмент Teachable Machine та бібліотеки програмної платформи TensorFlow. Досліджено, як значення навчальних параметрів впливають на точність моделей та обчислено оптимальні значення цих параметрів. Для розроблення ШНМ за допомогою TensorFlow проведено серію обчислювальних експериментів і досліджено, як архітектура мережі (кількість та типи шарів, кількість нейронів тощо) впливає на результати роботи моделі. В результаті знайдено раціональну внутрішню архітектуру ШНМ, яка дозволяє досягнути найбільшої точності та найменших втрат.

Проведено порівняльний аналіз практичної ефективності моделей, розроблених за допомогою Teachable Machine та TensorFlow. Обидві моделі демонструють приблизно однакову точність. Але при цьому модель, створена на основі TensorFlow, уникає помилок класифікації, яких припускається модель Teachable Machine. Це дозволяє зробити висновок, що індивідуальний підбір раціональної архітектури ШНМ для кожної окремої задачі дозволяє досягти суттєво кращих результатів, ніж використання загальних універсальних рішень.

Розроблені моделі ШНМ та одержані за їх допомогою результати свідчать про високу ефективність застосування методів штучного інтелекту для вирішення проблем діагностики будівельних конструкцій. На відміну від традиційних способів обстеження, запропоновані моделі дозволяють проводити діагностику автоматично в режимі реального часу шляхом аналізу фотографічних зображень, зображень із веб-камери комп'ютера, смартфона чи квадрокоптера. Останнє дає можливість виконувати обстеження споруд без фізичної присутності спеціалістів на об'єкті, що особливо важливо під час роботи у небезпечних місцях: на висоті, в аварійних та частково зруйнованих будівлях, на замінованих територіях тощо.

Запропоновані методи розроблення ШНМ у подальшому можуть бути розвинуті для діагностики та класифікації широкого кола дефектів та пошкоджень будівельних конструкцій і споруд.

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Березюк А. Н., Савицкий Н. В., Шимон Н. И. и др. Диагностика и оценка технического состояния строительных конструкций и оснований зданий и сооружений. Днепропетровск, 1996. 176 с.
2. Лучко І. Й., Коваль П. М., Дем'ян М. Л. Методи дослідження та випробування будівельних матеріалів і конструкцій. НАН України : Фіз.-мех. ін-т ім. Г.В. Карпенка. Львів : Каменяр, 2001. 425 с.
3. Суханов В. Т., Коробко О. О., Лисенко В. А. Діагностика, оцінка та методи обстеження. Одеса : Optimum, 2005. 190 с.
4. Бліхарський З. Я. Реконструкція та підсилення будівель і споруд. Львів : Вид-во НУ «Львівська політехніка», 2008. 106 с.
5. Perez H., Tah J. H. M., Mosavi A. Deep learning for detecting building defects using convolutional neural networks. *Sensors*. 2019. № 19. P. 3556. (22 pages).
6. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition. *ArXiv*. 2014. Vol. 1409. P. 1556.
7. Rajadurai R. S., Kang S-T. Automated vision-based crack detection on concrete surfaces using deep learning. *Appl. Sci*. 2021. № 11. P. 5229 (14 pages).

8. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 2017. № 60. Pp. 84–90.
9. Dorafshan S., Coopmans C., Thomas R. J., Maguire M. Deep learning neural networks for sUAS-assisted structural inspections : feasibility and application. *Proceedings of 2018 International Conference on Unmanned Aircraft Systems* (Dallas, USA). 2018. Pp. 874–882.
10. Chun P.-J., Yamane T., Maemura Yu. A deep learning-based image captioning method to automatically generate comprehensive explanations of bridge damage. *Comput. Aided Civ. Inf.* 2022. № 37. Pp. 1387–1401.
11. Munawar H. S., Hammad A. W. A., Haddad A., Soares C. A. P., Waller S. T. Image-based crack detection methods: a review. *Infrastructures*. 2021. № 6. P. 115 (20 pages).
12. Maguire M., Dorafshan S., Thomas R. J. SDNET2018 : a concrete crack image dataset for machine learning applications. Utah State University, 2018. URL : <https://doi.org/10.15142/T3TD19>
13. Веб-сайт хмарного інструменту Teachable Machine. URL : <https://teachablemachine.withgoogle.com>
14. Веб-сайт бібліотеки машинного навчання TensorFlow на JavaScript. URL : <https://www.tensorflow.org/js>
15. Веб-сайт платформи машинного навчання TensorFlow. URL : <https://www.tensorflow.org>
16. Веб-сайт хмарного середовища Colaboratory. URL : <https://colab.research.google.com>
17. Веб-сайт інтерактивного середовища розробки програмного забезпечення Jupyter. URL: <https://jupyter.org>

## REFERENCES

1. Berezyuk A.N., Savitsky N.V., Shimon N.I. et al. *Diagnostika i otsenka tekhnicheskogo sostoyaniya stroitel'nykh konstruksiy i osnovaniy zdaniy i sooruzheniy* [Diagnostics and evaluation of the technical condition of building structures and foundations of buildings and structures]. Dnipropetrovsk, 1996, 176 p. (in Russian).
2. Luchko I.Y., Koval P.M. and Demyan M.L. *Metody doslidzhennya ta vyprobuvannya budivel'nykh materialiv i konstruksiy* [Methods of research and testing of building materials and structures]. NAS of Ukraine: Phys.-Mech. Institute named after H.V. Karpenko. Lviv : Kamenyar Publ., 2001, 425 p. (in Ukrainian).
3. Sukhanov V.T., Korobko O.O. and Lysenko V.A. *Diahnostyka, otsinka ta metody obstezhennya* [Diagnosis, assessment and examination methods]. Odesa : Optimum Publ., 2005, 190 p. (in Ukrainian).
4. Bliarsky Z.Ya. *Rekonstruksiya ta pidsylennya budivel' i sporud* [Reconstruction and strengthening of buildings and structures]. Lviv : Lviv Polytechnic University Publishing House, 2008, 106 p. (in Ukrainian).
5. Perez H., Tah J.H.M. and Mosavi A. Deep learning for detecting building defects using convolutional neural networks. *Sensors*. 2019, no. 19, p. 3556 (22 pages).
6. Simonyan K. and Zisserman A. Very deep convolutional networks for large-scale image recognition. *ArXiv*. 2014, vol. 1409, p. 1556.
7. Rajadurai R.S. and Kang S-T. Automated vision-based crack detection on concrete surfaces using deep learning. *Appl. Sci*. 2021, no. 11, p. 5229 (14 pages).
8. Krizhevsky A., Sutskever I. and Hinton G.E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 2017, no. 60, pp. 84–90.
9. Dorafshan S., Coopmans C., Thomas R.J. and Maguire M. Deep learning neural networks for sUAS-assisted structural inspections : feasibility and application. *Proceedings of 2018 International Conference on Unmanned Aircraft Systems* (Dallas, USA). 2018, pp. 874–882.
10. Chun P.-J., Yamane T. and Maemura Yu. A deep learning-based image captioning method to automatically generate comprehensive explanations of bridge damage. *Comput. Aided Civ. Inf.* 2022, no. 37, pp. 1387–1401.
11. Munawar H.S., Hammad A.W.A., Haddad A., Soares C.A.P. and Waller S.T. Image-based crack detection methods: a review. *Infrastructures*. 2021, no. 6, p. 115 (20 pages).
12. Maguire M., Dorafshan S. and Thomas R.J. SDNET2018 : a concrete crack image dataset for machine learning applications. Utah State University, 2018. URL : <https://doi.org/10.15142/T3TD19>
13. *Veb-sayt khmarnoho instrumentu Teachable Machine* [Cloud tool Teachable Machine]. URL : <https://teachablemachine.withgoogle.com> (in Ukrainian).
14. *Veb-sayt biblioteki mashynnoho navchannya TensorFlow na JavaScript* [JavaScript library of the machine learning platform TensorFlow]. URL : <https://www.tensorflow.org/js> (in Ukrainian).
15. *Veb-sayt platformy mashynnoho navchannya TensorFlow* [Machine learning platform TensorFlow]. URL : <https://www.tensorflow.org> (in Ukrainian).
16. *Veb-sayt khmarnoho seredovyshcha Colaboratory* [Cloud environment Colaboratory]. URL : <https://colab.research.google.com> (in Ukrainian).
17. *Veb-sayt interaktyvnoho seredovyshcha rozrobky prohramnoho zabezpechennya Jupyter* [Development environment Jupyter]. URL : <https://jupyter.org> (in Ukrainian).

Надійшла до редакції: 22.02.2023.