



СИТНИК ЛАРИСА ГРИГОРІВНА,

кандидатка технічних наук, доцентка,
доцентка кафедри кібернетики та інформатики,
Сумський національний аграрний університет, м. Суми, Україна

Larysa Sytnyk,

Candidate of Technical Sciences, associate professor,
Associate Professor at the Department of Cybernetics and Informatics,
Sumy National Agrarian University, Sumy, Ukraine

E-mail: l.sitnik@snau.edu.ua

ORCID iD: <https://orcid.org/0000-0001-8176-750X>

АЛГОРИТМІЧНА МЕДІАГРАМОТНІСТЬ В УМОВАХ РОЗВИТКУ ГЕНЕРАТИВНОГО ШІ: ДОСВІД НАВЧАННЯ НЕЙРОМЕРЕЖ ЗА ДОПОМОГОЮ GOOGLE TEACHABLE MACHINE

А Стрімкий розвиток генеративного штучного інтелекту вимагає перегляду підходів до формування медіаграмотності сучасних фахівців. Раніше висока ресурсомісткість і потреба у глибоких технічних знаннях були суттєвою перешкодою для масового використання ШІ. Це обмежувало впровадження інтелектуальних систем у вебдодатках і навчальних курсах загального профілю.

Тепер же демократизація технологій дозволяє змінити акцент у навчанні: замість написання складного коду користувачі можуть зосередитися на кураторстві даних. У статті проаналізовано практичний досвід застосування платформи Google Teachable Machine у межах навчальної дисципліни «Основи штучного інтелекту» для підготовки нейронних мереж до розпізнавання образів у режимі реального часу.

Ключові слова: штучний інтелект; машинне навчання; Google Teachable Machine; трансферне навчання; комп'ютерний зір; нейронні мережі; вебтехнології

ALGORITHMIC MEDIA LITERACY IN THE CONTEXT OF GENERATIVE AI DEVELOPMENT: AN EXPERIENCE IN TRAINING NEURAL NETWORKS USING THE GOOGLE TEACHABLE MACHINE

С The rapid development of generative artificial intelligence requires a review of approaches to media literacy training for modern professionals. Previously, high resource intensity and the need for in-depth technical knowledge were significant barriers to the widespread use of AI, which limited the implementation of intelligent systems in web applications and general training courses. Now, however, the democratization of technology allows for a shift in the focus of education: instead of writing complex code, users can concentrate on data curation. This article analyzes the practical experience of using the Google Teachable Machine platform within the academic discipline "Fundamentals of Artificial Intelligence" to train neural networks to recognize images in real time.

Keywords: artificial intelligence; machine learning; Google Teachable Machine; transfer learning; computer vision; neural networks; webtechnologies

Актуальність проблеми у загальному вигляді та її зв'язок із важливими практичними завданнями. Історично склалося так, що розробка систем штучного інтелекту (ШІ) та машинного навчання (ML) була прерогативою вузького кола фахівців, які володіли глибокими знаннями в галузі математичної статистики, лінійної алгебри та програмування на Python або C++, а також мали доступ до значних обчислювальних ресурсів.

Традиційний процес створення моделі включав збір величезних наборів даних, їхнє маркування, налаштування архітектури нейронної мережі та навчання на серверних кластерах із використанням графічних процесорів. Цей бар'єр входу фактично відокремлював технологію від кінцевих користувачів, перетворюючи ШІ на «чорну скриньку», механізми якої залишалися непрозорими для широкого загалу.

Поява Google Teachable Machine ознаменувала фундаментальний зсув у технологічній парадигмі. Відбувся перехід від моделі «чорної скриньки» до концепції «прозорої скриньки». Завдяки візуалізації процесів користувач перестає бути пасивним споживачем і може зрозуміти логіку прийняття рішень. Це вебінструмент, який дозволяє створювати моделі машинного навчання без написання коду, використовуючи лише веббраузер. Це не просто освітній ігровий інструмент, а платформа, що демонструє можливості клієнтських обчислень і трансферного навчання у реальному часі [6].

У дослідженні представлено досвід організації освітнього процесу із застосуванням інструменту Google Teachable Machine, а також аналіз принципів трансферного навчання в рамках дисципліни «Основи штучного інтелекту».

Аналіз попередніх досліджень і публікацій. Проблема демократизації технологій штучного інтелекту та машинного

навчання є предметом активних дискусій у науковій спільноті останніми роками. Фундаментальні аспекти класичного глибокого навчання, що вимагають значних обчислювальних ресурсів і високопродуктивних графічних процесорів, детально висвітлені у працях таких дослідників, як LeCun, Y., Bengio, Y., & Hinton, G. [4]. Однак висока ресурсоемність і потреба у глибоких технічних знаннях залишаються суттєвою перешкодою для демократизації ШІ, обмежуючи його використання у вебдодатках і навчальних курсах загального профілю.

З появою бібліотеки TensorFlow.js, описаної у роботах D. Smilkov та співавторів, стало можливим перенесення обчислень на бік клієнта (client-side computing), що перетворило веббраузер на повноцінну платформу для навчання нейромереж [8]. Це відкрило шлях до створення інструментів, а саме до Google Teachable Machine. Дослідження, присвячені використанню таких по-соді платформ, акцентують увагу на зміні парадигми від «написання коду» до «кураторства даних». Зокрема, M. Carney та співавтори розглядають Teachable Machine як інструмент, що демонструє принципи трансферного навчання (transfer learning) – методу, який дозволяє адаптувати попередньо навчені моделі (наприклад, MobileNet) для нових задач із мінімальною кількістю даних [2].

Водночас потребують детальнішого вивчення питання практичної імплементації браузерних моделей у реальному часі. Зокрема, це стосується керування динамічними об'єктами в іграх та аналізу технічних обмежень (освітлення, апаратні можливості).

Виділення невіршених раніше частин загальної проблеми, яким присвячується означена стаття. Попри активне обговорення демократизації ШІ, недостатньо вивченими залишаються практичні аспекти використання браузерних моделей для задач реального часу, зокрема керування динамічними об'єктами, аналізу їхніх технічних обмежень. Також потребують систематизації методичні засади впровадження цих технологій в освітній процес саме для розвитку алгоритмічної медіаграмотності фахівців, а не лише технічних навичок.

Мета статті: аналіз ефективності застосування методу трансферного навчання у браузерному середовищі (на базі Google Teachable Machine) для створення систем комп'ютерного зору, що працюють у реальному часі. На відміну від існуючих оглядів інструментарію, робота покликана заповнити прогалину в методиці використання цих платформ саме для розвитку алгоритмічної медіаграмотності фахівців. У статті ставиться завдання визначити оптимальні підходи до інтеграції цих технологій в освітній процес, де акцент зміщується з написання складного коду на розуміння принципів кураторства даних і технічних обмежень штучного інтелекту.

Викладення основного матеріалу. Сучасний етап розвитку цифрового суспільства дослідники визначають як період демократизації штучного інтелекту. Раніше робота з нейромережами вимагала глибоких знань програмування та математики, однак сьогодні завдяки інструментам без коду ці технології стають загальнодоступними [9].

Така трансформація докорінно змінює освітні завдання: відбувається перехід від вивчення синтаксису коду до формування алгоритмічної медіаграмотності. Як зазначають Н. Venbu та співавтори, сучасна цифрова компетентність – це передусім здатність розуміти логіку рішень комп'ютера та ефективно, а головне – критично взаємодіяти з такими системами [1].

Під алгоритмічною медіаграмотністю у цьому дослідженні розуміється не лише навичка користування генеративним ШІ, а й здатність критично оцінювати його роботу, усвідомлюючи пряму залежність між якістю вхідних даних й отриманим результатом.

Технічним проривом, що уможливив реалізацію цього підходу в навчальних аудиторіях, стала поява бібліотек, подібних до TensorFlow.js. Вони дозволяють запускати складні алгоритми машинного навчання безпосередньо у браузері, використовуючи ресурси локального пристрою без потреби у складних налаштуваннях серверів [7]. На цій технологічній базі функціонує Google Teachable Machine – інструмент, що є ключовим для запропонованої методики навчання.

Сутність методики полягає у використанні методу трансферного навчання. Замість тривалого вивчення мов програмування, фахівці одразу залучаються до дослідницької діяльності. Система використовує попередньо навчену потужну нейромережу, здатну розпізнавати базові форми та кольори, і дозволяє користувачеві швидко довести її розрізняти нові об'єкти, наприклад, специфічні жести рук чи типи рослин. У якості показового прикладу доцільно детальніше розглянути методику, застосовану дослідниками для вирішення прикладної задачі класифікації рослин [5]. Цей кейс ідеально ілюструє, як сучасні інструменти по-соді (без написання коду) дозволяють навіть початківцям реалізувати повний цикл машинного навчання: від збору даних до розгортання робочої моделі. У згаданій роботі авторами було поставлено завдання створити модель, здатну автоматично розрізняти листя трьох видів фруктових дерев: гуави, апельсина та манго. Вибір об'єктів не є випадковим – візуальна схожість форми листя робить це завдання нетривіальним для людини без спеціальної ботанічної підготовки, проте воно стає ефективним майданчиком для навчання алгоритмів комп'ютерного зору.

Загальний алгоритм роботи складався з чотирьох послідовних етапів:

1. Збір та підготовка набору даних. Якість роботи штучного інтелекту критично залежить від даних, на яких він навчався. Автори дослідження [5] сформували власний набір зображень, сфотографувавши листя у реальних умовах. Для кожного з трьох класів «Guava Leaf», «Orange Leaf», «Mango Leaf» було зібрано по 100 фотографій (рис. 1).

Це важливий момент: користувачі мають розуміти, що для навчання нейромережі недостатньо одного-двох прикладів. Вибірка зі 100 зображень на клас є оптимальним мінімумом для навчальних проєктів у Teachable Machine, забезпечуючи баланс між швидкістю навчання та точністю розпізнавання.

2. Завантаження даних у середовище Google Teachable Machine. На цьому етапі відбувається безпосередня

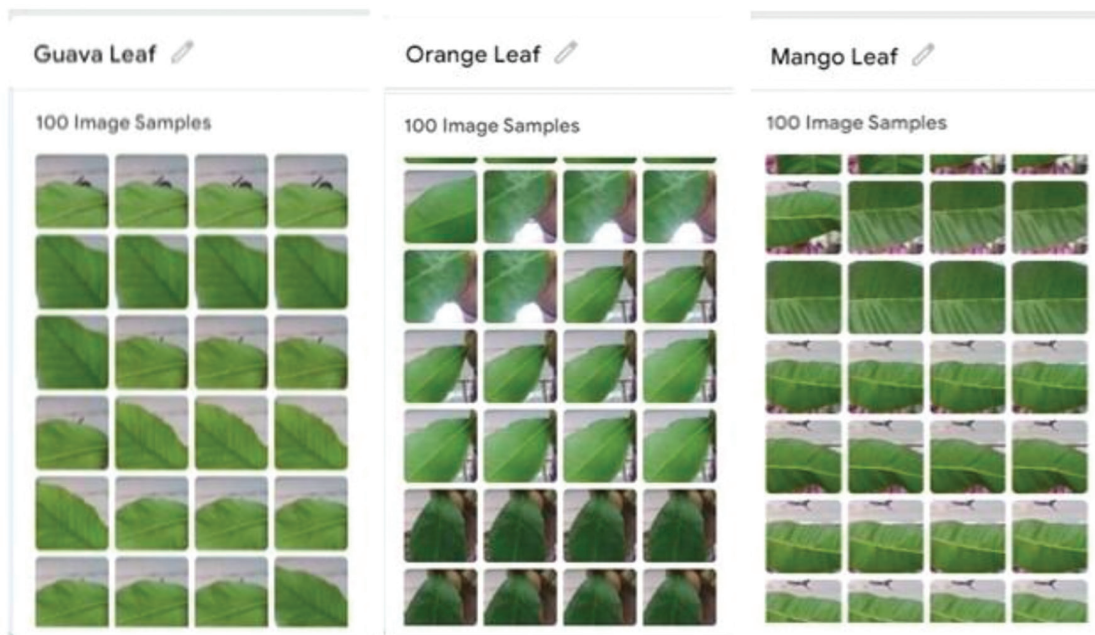


Рис. 1. Приклади зображень з навчального набору даних для трьох класів

взаємодія з інтерфейсом GTM. Методика передбачає створення трьох окремих класів. Інтуїтивно зрозумілий інтерфейс дозволяє користувачеві завантажити підготовлені файли у відповідні зони або використати вебкамеру для миттєвого захоплення зображень. Автори [5] зазначають, що платформа автоматично використовує методи трансферного навчання. Це означає, що система не вчить розпізнавати зображення із самого початку, а використовує вже існуючу потужну нейромережу, донавчаючи лише її фінальні шари для розрізнення конкретних типів листя.

3. Налаштування гіперпараметрів і тренування моделі. Хоча Teachable Machine позиціонується як доступна no-code платформа, вона дозволяє керувати ключовими параметрами процесу навчання, що є критично важливим для розуміння принципів роботи ШІ.

У розглянутому дослідженні [5] було використано налаштування, які забезпечили високу точність:

- кількість епох – 50. Це означає, що нейромережа пройшла повний цикл навчання на всьому наборі даних 50 разів. Збільшення кількості епох дозволяє системі краще вивчити ознаки, але й надмірна їхня кількість може призвести до перенавчання (коли модель просто запам'ятовує картинку, а не розуміє закономірності);

- швидкість навчання – 0.001. Стандартне значення, яке визначає, наскільки сильно коригуються ваги нейромережі на кожному кроці.

4. Тестування та валідація результатів. Фінальний етап експерименту – перевірка працездатності моделі. Для цього дослідники використали окремий набір зображень (по 15 штук для кожного класу), які не брали участі у процесі навчання (рис. 2). Це фундаментальне правило машинного навчання, тестувати систему потрібно лише на нових, незнайомих їй даних. Результати, отримані авторами, демонструють вражаючу ефективність інструменту: точність розпізнавання (Assiguacy) склала 1.00, тобто 100%.

Матриця помилок (Confusion Matrix), отримана авторами [5], показує, що жодне листя гуави не було сплутане з листями манго чи апельсина (рис. 3).

У цьому контексті алгоритмічна медіаграмотність визначається як здатність користувача не лише застосовувати інструменти штучного інтелекту, а й розуміти принципи їх функціонування. Ключовим стає усвідомлення прямої залежності кінцевого результату генерації від якості та збалансованості вхідних даних. Практична реалізація цього підходу можлива через безпосереднє залучення користувача до процесу тренування нейромережі. Робота з інструментами, подібними до Google Teachable Machine, дозволяє наочно побачити механіку навчання системи. Коли користувач самостійно збирає та класифікує дані, він починає розуміти природу помилок розпізнавання та причини виникнення алгоритмічної упередженості. Наприклад, стає очевидним, що модель може втратити точність при зміні освітлення або фону, оскільки система шукає статистичні збіги у пікселях, а не розуміє об'єкт як людина [3].

Отже, формується розуміння того, що ефективність штучного інтелекту визначається насамперед коректністю підібраної людиною інформації. Замість ідеалізації технології, розвивається вміння виявляти її вразливі місця. Це й є гуманістичний аспект технічної освіти – розуміння того, що за кожним автоматизованим алгоритмом стоїть людина, яка його навчала.

До появи Teachable Machine запуск нейронних мереж у браузері був пов'язаний із значними труднощами. JavaScript спочатку не призначався для виконання складних операцій із матрицями, на яких базуються алгоритми глибокого навчання. Більшість рішень поклалися на хмарні API, де дані (зображення чи аудіо) відправлялися на сервер, оброблялися, й результат повертався користувачеві. Це створювало проблеми із затримкою, залежністю від інтернет-з'єднання та, що найважливіше, з конфіденційністю даних [10].

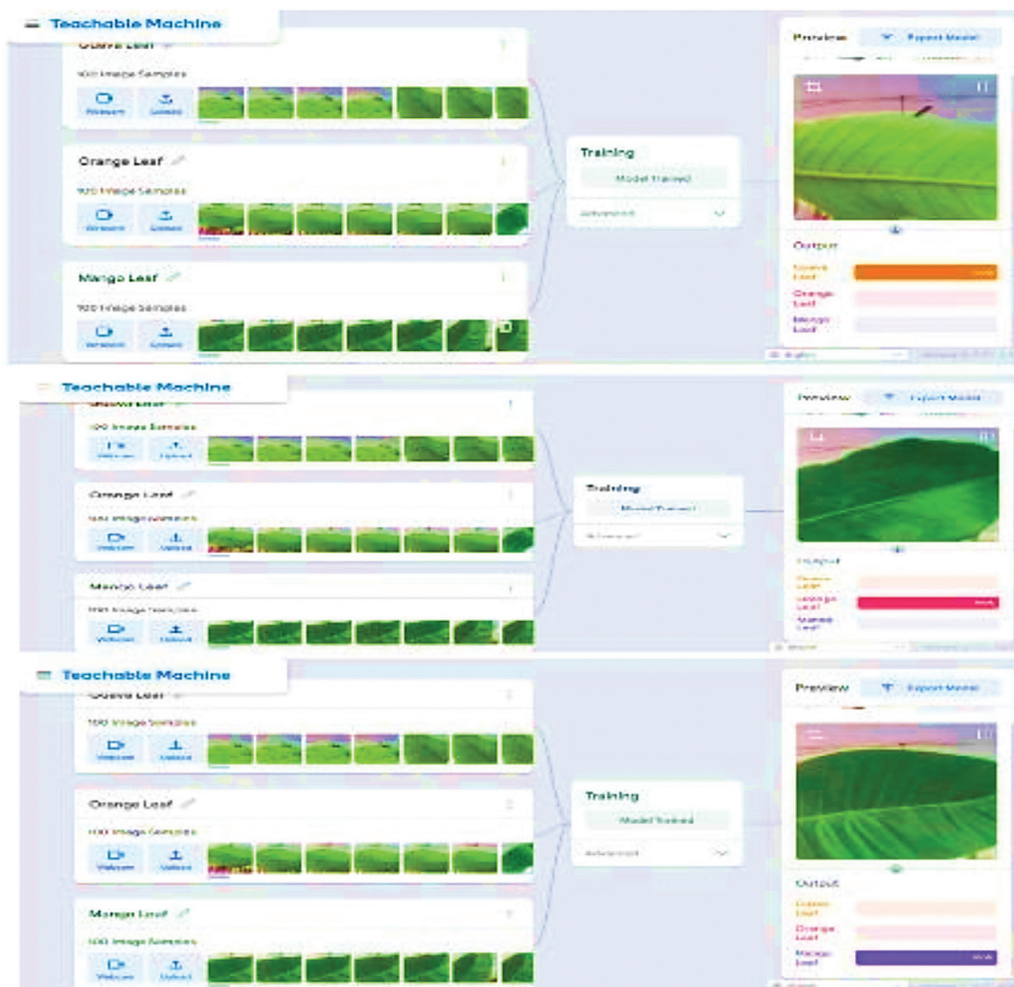


Рис. 2. Тестування моделі за класами типів листя

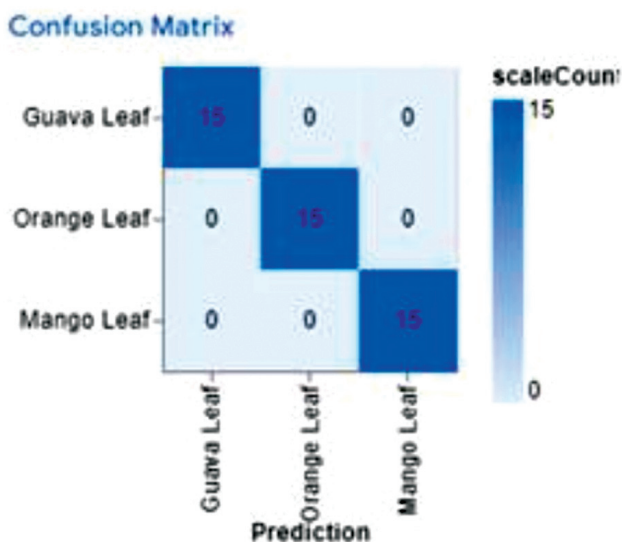


Рис. 3. Матриця помилок

Розглянемо приклад практичного завдання, запропонованого при вивченні теми «Прототипування системи класифікації зображень за допомогою Google Teachable Machine».

Робота над інтелектуальним інтерфейсом розпочалася з підготовки набору даних. На відміну від розпізнавання

статичних об'єктів, основну увагу було зосереджено на рухах і жестах, що слугують командами для керування ігровим персонажем.

Для цього було визначено систему класів, що відповідають ігровій логіці.

Клас 1. Спокій: базовий стан, коли користувач не виконує активних дій. Це критично важливо для запобігання помилковим висновкам.

Клас 2. Дія А: візуальний тригер для активації основної механіки (наприклад, стрибок або рух уперед).

Клас 3. Дія Б: тригер для альтернативної механіки (наприклад, атака або взаємодія з об'єктом).

Для кожного класу було зібрано від 40 до 50 зображень за допомогою вебкамери. Особлива увага приділялася варіативності вибірки (data augmentation): зйомка проводилася під різними кутами, зі зміною відстані до камери та при різному освітленні. Це дозволило мінімізувати ефект перенавчання (overfitting), коли модель запам'ятовує конкретний фон, а не об'єкт інтересу.

Розроблення моделі здійснювалося на базі платформи Teachable Machine, яка використовує бібліотеку TensorFlow.js. У якості базової архітектури використано метод transfer learning, що дозволяє донавчати попередньо треновані моделі на нових специфічних даних без необхідності у великих обчислювальних ресурсах.

Саме через набуття практичних навичок конструювання наборів даних розкривається сутність алгоритмічної медіаграмотності. Процес самостійного тренування моделі трансформує пасивне сприйняття технології в усвідомлене розуміння її внутрішньої логіки та обмежень. Користувач перестає сприймати штучний інтелект як безпомилковий інструмент і починає бачити в ньому статистичну систему, залежну від вхідної інформації.

Експериментальна робота з класами даних наочно демонструє механізми виникнення алгоритмічної упередженості. Коли користувач власноруч виправляє помилки розпізнавання шляхом додавання нових зразків або зміни умов освітлення, він отримує докази вразливості нейромереж. Такий емпіричний досвід формує стійку навичку критичної оцінки будь-яких рішень, згенерованих автоматизованими системами, що є ключовим показником сформованої медіаграмотності.

У ході експерименту було налаштовано наступні гіперпараметри:

1. Кількість епох: експериментально встановлено, що збільшення кількості епох до 70–80 забезпечує стабілізацію функції втрат і підвищує точність розпізнавання до 95–98% на тренувальній вибірці.

2. Швидкість навчання: використання значення 0.001, що є стандартним для оптимізатора Adam у бібліотеці TensorFlow.js, дозволило досягти оптимального балансу між швидкістю збіжності моделі та точністю градієнтного спуску.

Наступний етап передбачав інтеграцію навченої моделі у вебдодаток. Використання формату TensorFlow.js забезпечило виконання обчислень на стороні браузера, що дало змогу отримати швидкість відгуку, достатню для ігрового процесу в реальному часі.

Представимо алгоритм взаємодії ML-моделі з ігровим персонажем:

1. Захоплення кадру: функція `loop()` постійно оновлює зображення з вебкамери.

2. Предикція: метод `model.predict()` аналізує вхідний кадр і повертає масив ймовірностей для кожного класу.

3. Фільтрація шумів: для уникнення хаотичних рухів персонажа впроваджено порогове значення ймовірності. Дія ініціюється лише за умови, що ймовірність

прогнозованого класу за функцією Softmax перевищує 0,85 (85%), що забезпечує стабільність керування персонажем у динамічному середовищі.

Процес тестування показав, що система залишається чутливою до рівня освітлення та іноді некоректно розпізнає схожі жести. З метою корекції цих помилок до навчального набору даних було додано більше зразків із «помилковими» станами, що суттєво покращило здатність моделі диференціювати команди. Тестування розробленої системи продемонструвало високу ефективність керування персонажем у реальному часі.

Отже, використання браузерних нейромереж дозволяє створювати інтерактивні ігрові інтерфейси без використання спеціалізованого апаратного забезпечення (контролерів, сенсорів), спираючись виключно на можливості комп'ютерного зору.

Висновки з даного дослідження. Використання Google Teachable Machine суттєво спрощує знайомство з основами машинного навчання, роблячи його доступним для широкої аудиторії. Завдяки бібліотеці TensorFlow.js оброблення даних відбувається безпосередньо на пристрої користувача, що забезпечує швидку роботу системи та захист персональної інформації.

Практичне застосування платформи підтвердило її ефективність в освітньому процесі. Головним результатом є не лише опанування технічних навичок, а й розвиток критичного мислення. Самостійна підготовка наборів даних дозволяє користувачам наочно побачити, як якість вхідної інформації впливає на точність роботи алгоритмів. Це допомагає усвідомити, що результати роботи ШІ залежать від того, на яких прикладах його навчали, та запобігає беззастережній довірі до автоматизованих рішень.

Перспективи подальших розвідок доцільно присвятити вивченню апаратного прискорення у браузері на основі технологій WebGPU та WASM. Це дозволить запускати архітектури генеративного ШІ та малі мовні моделі безпосередньо на пристроях користувачів. Цей підхід допоможе усунути технічні перешкоди та створити безпечне автономне середовище для опанування алгоритмів штучного інтелекту, що стане важливим кроком у розвитку сучасної цифрової освіти.

References

- Benbya, H., Pachidi, S., & Jarvenpaa, S. L. (2021). Special Issue Editorial: Artificial Intelligence in Organizations: Implications for Information Systems Research. *Journal of the Association for Information Systems*, 22 (2), 281-303. DOI: <https://doi.org/10.17705/1jais.00662>.
- Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J., Jongejan, J., Pitaru, A., & Chen, A. (2020). Teachable Machine: Approachable web-based tool for exploring machine learning classification. *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 1-8. DOI: <https://doi.org/10.1145/3334480.3382839>.
- Google. (2026). *Teachable Machine FAQ*. Teachable Machine. Retrieved from <https://teachablemachine.withgoogle.com/faq>.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521 (7553), 436-444. DOI: <https://doi.org/10.1038/nature14539>.
- Parate, R. K., Dhole, K. M., & Sharma, S. J. (2023). Classification of Leaf using Teachable Machine. *International Journal for Research in Applied Science and Engineering Technology*, 11 (9), 307-311. DOI: <https://doi.org/10.22214/ijraset.2023.55629>.
- Ridlo, Z. R., Dafik, Ningsih, S. P. A., & Anggaraini, A. L. (2025). Computer Vision on Education: Fostering AI Literacy using RBL-STEM with Google Teachable Machine. *Jurnal Penelitian & Pengembangan Pendidikan Fisika*, 11 (2), 197-210. DOI: <https://doi.org/10.21009/1.11205>.
- Sanjeevkumar B., Varun S. P., & Yokes Babu S. (2025). Assessing Google's Teachable Machine by deploying American Sign Language detection system. *International Journal for Multidisciplinary Research (IJFMR)*, 7 (1).
- Smilkov, D., Thorat, N., Assogba, Y., Yuan, A., Nickoloff, N., Nielsen, E., ... & Corrado, G. S. (2020). TensorFlow.js: Machine learning for the web and beyond. *Communications of the ACM*, 63 (5), 79-88. DOI: <https://doi.org/10.1145/3343431>.
- Strahringer, S., & Westner, M. (2024). Low-Code/No-Code - Demokratisierung der IT? *HMD Praxis der Wirtschaftsinformatik*, 61 (5), 1067-1069. DOI: <https://doi.org/10.1365/s40702-024-01108-w>.
- Vishakha, A. (2022). An Overview of Web-Based Machine Learning Frameworks. *Journal of Advances in Developmental Research*, 13 (2), 1-5. DOI: <https://doi.org/10.5281/zenodo.14684704>.

Дата надходження до редакції авторського оригіналу: 17.02.2026