УДК 373.5.091.33:004.8
DOI: 10.24144/2524-0609.2026.58.148-152

Кривонос Олександр Миколайович
кандидат педагогічних наук, доцент
доцент кафедри комп'ютерних наук та інформаційних технологій
Житомирський державний університет імені Івана Франка, м.Житомир, Україна
krypton@zu.edu.ua
https://orcid.org/0000-0002-4211-6541

МЕТОДИКА ФОРМУВАННЯ ШІ-ГРАМОТНОСТІ УЧНІВ ЗАСОБАМИ NO-CODE ІНСТРУМЕНТІВ (НА ПРИКЛАДІ GOOGLE TEACHABLE MACHINE)

Анотація. Стрімка експансія штучного інтелекту (ШІ) зумовлює трансформацію освітньої парадигми, вимагаючи формування ШІ-грамотності як базової компетенції учнів. Проте високий «поріг входження» через складність математичного апарату та синтаксису мов програмування створює методичне протиріччя для учнів середньої ланки. Мета дослідження полягає у теоретичному обґрунтуванні та розробці методичних рекомендацій щодо використання no-code інструментів (на прикладі Google Teachable Machine) для формування базової ШІ-грамотності учнів у процесі навчання інформатики. Застосовано порівняльно-типологічний метод для аналізу підходів до вивчення ІТ та системний аналіз для розгляду процесу формування ШІ-грамотності як цілісної системи. Обґрунтовано ефективність використання методу навчання з перенесенням знань (transfer learning) засобами візуальних ML-платформ. Доведено, що no-code підхід дозволяє змістити акцент із написання коду на концептуальне проєктування інтелектуальних моделей та аналіз даних. Визначено потенціал інтеграції таких моделей у середовище Scratch для створення систем безконтактного керування та екологічного моніторингу. Виявлено обмеження інструменту, зокрема ризик виникнення «ілюзії розуміння» та технічні залежності.

Ключові слова: штучний інтелект, машинне навчання, no-code інструменти, Teachable Machine, навчання інформатики.

Вступ. Стрімкий розвиток та експансія технологій генеративного штучного інтелекту (ШІ) у всі сфери людської життєдіяльності спричинили фундаментальну трансформацію освітньої парадигми. У 2024–2026 роках розуміння принципів функціонування алгоритмів машинного навчання (Machine Learning, ML) остаточно перейшло з категорії вузькоспеціалізованих технічних навичок до базового інваріанту загальної функціональної грамотності особистості. У контексті реалізації концепції Нової української школи (НУШ) та світових стратегій цифровізації освіти (зокрема Digital Education Action Plan ЄС), виникає гостра потреба у підготовці здобувачів освіти, здатних не лише ефективно експлуатувати готові ШІ-сервіси, а й критично аналізувати логіку їхнього «навчання», валідувати вхідні дані та усвідомлювати соціально-етичні наслідки автоматизованих рішень.

Сучасний етап розвитку методики навчання інформатики вимагає переходу від вивчення закритих програмних середовищ до опанування принципів побудови складних систем. Проте впровадження тематики ШІ у шкільний курс стикається з суттєвим методичним протиріччям: високий «поріг входження», зумовлений складністю математичного апарату та специфічним синтаксисом мов програмування (зокрема Python), часто стає перешкодою для учнів середньої ланки. Це породжує ризик формування фрагментарних знань, де ШІ сприймається як «магічна технологія» без зрозумілої внутрішньої структури.

Використання інструментарію класу no-code, яскравим представником якого є Google Teachable Machine, дозволяє ефективно розв'язати зазначену суперечність. Даний підхід актуалізує впровадження стратегії «learning by doing», де освітній акцент зміщується з технічних аспектів написання коду на концептуальне проектування: архітектуру інтелектуальної моделі, формування репрезентативних датасетів та інтерпретацію отриманих результатів. Це не лише демократизує доступ до високотехнологічних знань, забезпечуючи рівні можливості для самореалізації учнів незалежно від їхнього попереднього досвіду програмування, а й створює необхідний когнітивний фундамент для подальшого переходу до професійної розробки систем штучного інтелекту.

Аналіз останніх досліджень і публікацій. Проблема інтеграції штучного інтелекту в освітній простір є об'єктом прискіпливої уваги українських дослідників. Дослідження О.Панухник доповнює теоретичну базу статті аналізом етичних і відповідальних меж використання інтелектуальних систем, що дозволяє обґрунтувати необхідність формування критичного мислення учнів як запобіжника неконтрольованому впливу ШІ в освітньому процесі [8].

Питання загальної методології та адаптації змісту комп'ютерних дисциплін до вимог сучасного ринку праці детально розкрито у дослідженні К.Осадчої [7]. У контексті викладання програмування, В.Величко, О.Ганієв та С.Жадан акцентують увагу на ШІ як інструменті для аналізу та класифікації задач, що дозволяє автоматизувати перевірку знань [5].

Окремий пласт досліджень присвячений генеративним моделям. Так, Ю.Сіциліцин та В.Осадчий аналізують можливості використання ChatGPT у дистанційному навчанні програмування початківців [9]. Логічним продовженням цієї дискусії є наше попереднє дослідження, у якій детально досліджено використання генеративного штучного інтелекту безпосередньо для створення програмного коду. В дослідженні [6] обґрунтовано ефективність інтеграції ШІ-інструментів у процес розробки ПЗ, що дозволяє змістити акцент з рутинного кодування на архітектурне проектування та алгоритмічну логіку.

Важливим складником теоретичного базису дослідження є аналіз міжнародного досвіду формування ШІ-грамотності (AI Literacy). У систематичному огляді літератури M.Laupichler та ін. узагальнено підходи до викладання ШІ у вищій та загальній освіті, що дозволяє визначити універсальні компоненти ШІ-компетентності для різних вікових груп [1]. Концептуальні засади та дизайн-міркування щодо самої структури цієї грамотності розкрито у праці D.Long та B.Magerko, де автори виокремлюють конкретні

© Кривонос О. М.
Creative Commons Attribution NonCommercial ShareAlike License (CC BY-NC-SA 4.0)

компетенції, необхідні для розуміння принципів роботи інтелектуальних систем [2].

Питання вікової адекватності та розробки адаптованих навчальних планів для середньої школи (K-12) висвітлено у дослідженні фахівців MIT Media Lab – R.Williams, H.W.Park та C.Breazeal. Автори доводять ефективність практико-орієнтованого підходу до вивчення ШІ саме в шкільному віці, що безпосередньо корелює з нашою методикою використання візуальних інструментів [4].

Проте, якщо генеративний ШІ (згідно наших попередніх досліджень) допомагає у створенні готового програмного продукту, то використання no-code платформ, таких як Teachable Machine, дозволяє учням зрозуміти «фундамент» цих технологій – принципи збору даних та навчання нейронних мереж, які згодом і стають основою для роботи потужних генеративних моделей [6]. Таким чином, наше дослідження доповнює існуючі напрацювання, пропонуючи методику формування базових ML-компетентностей ще до етапу активного використання генеративного ШІ в програмуванні.

Для глибшого розуміння місця інструментів типу No-code у системі комп'ютерної освіти необхідно провести демаркацію між традиційними підходами до програмування та сучасними методами, що базуються на використанні штучного інтелекту. Якщо у попередніх дослідженнях автора акцент зміщувався на використання генеративного ШІ як допоміжного засобу для автоматизації створення програмного коду, то поточна розробка фокусується на «докодовому» етапі – розумінні природи даних та логіки навчання нейромереж.

Такий підхід дозволяє виокремити зміну ролі учня: від написання чітких детермінованих інструкцій до проектування інтелектуальних моделей через добір релевантних прикладів.

Мета статті: теоретичне обґрунтування та розробка методичних рекомендацій щодо використання no-code інструментів (на прикладі сервісу Google Teachable Machine) для формування базової ШІ-грамотності учнів у процесі навчання інформатики.

Методи дослідження: порівняльно-типологічний метод використано для аналізу різних підходів до вивчення інформаційних технологій; системний аналіз дозволяє розглядати процес формування ШІ-грамотності як цілісну систему, а не набір випадкових вправ.

Виклад основного матеріалу. В основі функціонування сервісу Teachable Machine лежить бібліотека відкритого коду TensorFlow.js, що дозволяє інтегрувати складні алгоритми машинного навчання безпосередньо у веб-середовище. З дидактичної точки зору, ключовою перевагою інструменту є реалізація методу навчання з перенесенням знань (transfer learning). Дана технологія передбачає використання претренованих нейронних мереж (зокрема архітектури MobileNet), які вже мають сформовані вагові коефіцієнти для розпізнавання базових ознак об'єктів. Процес «навчання», який здійснює учень, полягає у тонкому налаштуванні (fine-tuning) вихідного шару моделі на специфічному наборі даних, що мінімізує часові витрати на обчислення та робить результат доступним у реальному часі [1].

Для цілей методики навчання інформатики доцільно класифікувати функціональні можливості інструменту за типом вхідних модальностей, кожна з яких має специфічне освітнє навантаження:

– Класифікація візуальних образів (Image Project). Система здійснює виявлення статистичних закономірностей у піксельних масивах. У контексті навчання це дозволяє продемонструвати концепцію формування ознак (feature extraction) та критичну роль варіативності вибірки для запобігання помилковим кореляціям між об'єктом та його оточенням.

– Аналіз акустичних сигналів (Audio Project). Технологічна реалізація базується на трансформації звукових коливань у спектрограми – візуальне представлення частотно-часових характеристик звуку. Даний модуль дозволяє наочно продемонструвати учням процес конвертації аналогового сигналу у цифровий формат, придатний для аналізу нейронною мережею.

– Кінематичне моделювання (Pose Project). Модуль функціонує на базі алгоритму PoseNet, що спеціалізується на ідентифікації ключових антропометричних точок. З методичної точки зору, це дозволяє вивчити принципи абстрагування даних, де складне зображення людського тіла декомпозується до сукупності векторів та геометричних примітивів («скелетної моделі»).

Важливим аспектом безпеки освітнього середовища є архітектурне рішення on-device machine learning. Оскільки всі етапи – від захоплення даних до виведення результату – відбуваються локально в межах браузера, забезпечується повна конфіденційність біометричної інформації учнів. Дана характеристика знімає низку етичних та юридичних бар'єрів при впровадженні технологій штучного інтелекту в навчальних закладах.

Завершальним етапом роботи з інструментом є експорт моделі у форматі TensorFlow.js або TensorFlow Lite, що забезпечує технологічну наступність: від візуального проектування у Teachable Machine до програмної реалізації інтелектуальних систем у професійних середовищах розробки або візуальних мовах програмування.

Функціональна архітектура Teachable Machine базується на принципі абстрагування від складних математичних обчислень та синтаксису програмування, що дозволяє змістити фокус навчання на методологію побудови систем штучного інтелекту. Використання цього інструменту в освітньому процесі реалізує концепцію «конструктивізму», де учень виступає активним творцем інтелектуального продукту. Основна перевага полягає у візуалізації життєвого циклу розробки моделі: від формування навчальної вибірки (dataset) до оцінки точності класифікації (рис.1). Такий підхід сприяє формуванню у школярів алгоритмічного мислення нового типу, де розв'язання задачі досягається не через написання чіткої інструкції, а через добір релевантних прикладів для навчання нейронної мережі.

Методична цінність сервісу розкривається через можливість практичного дослідження понять класифікації та розпізнавання образів. Учні отримують змогу експериментувати з трьома типами вхідних даних: візуальними (зображення), акустичними (звуки) та кінематичними (пози тіла). Це дозволяє вчителю інтегрувати елементи машинного навчання в різні розділи курсу інформатики. Наприклад, при вивченні теми «Кодування інформації» учні можуть дослідити, як нейромережа інтерпретує піксельні дані або звукові хвилі, перетворюючи їх на числові вектори для подальшого аналізу. Таким чином, інструмент стає засобом міжпредметної інтеграції, поєднуючи інформатику з біологією (розпізнавання видів рослин) або фізикою (аналіз звукових коливань).

Окремої уваги заслуговує аналіз етапу тестування та валідації моделі, який є критичним для розвитку аналітичних здібностей. Teachable Machine надає інструменти для виявлення системних помилок на-

Рис.1. Приклад реалізації моделі машинного навчання, призначеної для бінарної класифікації об'єктів (наприклад, диференціація зображень робота та собаки).
Джерело: систематизовано автором

вчання, зокрема явища перенавчання (overfitting) або зміщення даних (bias). Коли модель демонструє некоректні результати через недостатню варіативність вибірки, учень змушений проводити рефлексивний аналіз: шукати причини невдачі у вхідних даних та коригувати їх. Це стимулює розвиток критичного ставлення до технологій ШІ та розуміння того, що якість інтелектуальної системи безпосередньо залежить від об'єктивності та репрезентативності інформації, на якій вона базувалася.

Практичне впровадження Teachable Machine у шкільний курс інформатики доцільно реалізовувати через метод проєктів, що дозволяє учням пройти повний цикл розробки інтелектуальної системи. Один із базових сценаріїв передбачає створення моделі екологічного моніторингу, де учні навчають нейромережу класифікувати побутові відходи (пластик, папір, скло) за допомогою веб-камери. На етапі підготовки даних учні стикаються з проблемою варіативності: модель має розпізнавати об'єкт незалежно від освітлення чи ракурсу. Таке завдання не лише формує навички роботи з даними, а й демонструє прикладне значення штучного інтелекту у вирішенні глобальних суспільних викликів, реалізуючи міжпредметні зв'язки з екологією.

Більш складний рівень реалізації передбачає інтеграцію навченої моделі у візуальне середовище програмування Scratch. Завдяки спеціальному розширенню, експортована з Teachable Machine модель стає джерелом подій для програмного коду. Це відкриває можливості для створення інтерфейсів безконтактного керування: наприклад, розробка гри, де рух персонажа контролюється певними жестами рук або положенням голови (використання моделі поз – PoseNet). У такому контексті ШІ перестає бути самоціллю і стає інструментом введення даних, що дозволяє учням глибше зрозуміти архітектуру сучасних людино-машинних інтерфейсів.

Особливу методичну цінність має впровадження лабораторних робіт на дослідження аудіо-класифікації. Учні можуть створювати системи «розумного дому», що реагують на специфічні звукові команди (плескіт, свист або ключові слова). Важливим дидактичним моментом тут є аналіз фонового шуму: учні навчають модель окремому класу «Тиша/Фон», щоб мінімізувати хибні спрацьовування. Це дозволяє на практиці розібрати поняття співвідношення сигналу та шуму, а також важливість очищення даних (data cleaning) перед початком навчання нейромережі.

Завершальним етапом практичної реалізації має стати рефлексивний аналіз створених моделей. Вчитель може запропонувати учням «зламати» моделі один одного, підбираючи вхідні дані, які можуть ввести алгоритм в оману (наприклад, фотографію об'єкта замість самого об'єкта). Такий ігровий елемент критичного тестування дозволяє усвідомити межі застосування ШІ та зрозуміти, що комп'ютер не «розуміє» сутність предмета, а лише ідентифікує статистичні закономірності у вхідному сигналі.

Практична реалізація: Розробка інтелектуальної системи моніторингу присутності учнів.

Для демонстрації прикладного потенціалу no-code інструментарію доцільно розглянути проєктне завдання з автоматизації обліку відвідуваності занять. Метою кейсу є побудова моделі бінарної класифікації, здатної диференціювати стан робочого місця учня («присутній» / «відсутній»).

Методологія моделювання задачі передбачає наступні етапи:

1. Учні створюють два класи візуальних даних. Перший клас («Empty Seat») містить зображення порожніх учнівських місць за різних умов освітлення та ракурсів. Другий клас («Student Present») включає репрезентативну вибірку зображень учнів. На цьому етапі вчитель наголошує на важливості об'єктивності даних. Учні мають зрозуміти, що результати роботи ШІ залежать від якості завантажених фотографій. Якщо набір прикладів буде одноманітним (наприклад, тільки діти в однаковій формі), система може «звикнути» до однієї ознаки й помилятися в інших ситуаціях. Тому завдання учнів – зробити вибірку максимально різноманітною.

2. Процес навчання нейронної мережі в середовищі Teachable Machine базується на архітектурі MobileNet. У межах дослідження учням пропонується експериментально встановити оптимальну кількість епох (циклів) навчання (рекомендовано 80–100) для досягнення високої точності (Accuracy) та мінімізації втрат (Loss) при роботі зі схожими за кольоровою гамою фоновими об'єктами.

3. Отримана модель експортується у середовище візуального програмування Scratch. Програмна логіка проєкту передбачає використання розширення Teachable Machine для створення циклічного алгоритму перевірки: якщо ймовірність класу «Empty Seat» перевищує встановлений поріг (наприклад,

0.9) протягом визначеного часу, система автоматично інкрементує лічильник відсутніх або подає сигнал сповіщення.

Реалізація такого проєкту дозволяє учням не лише опанувати технічні навички роботи з ML-платформами, а й усвідомити етичні аспекти використання систем відеоспостереження, конфіденційності біометричних даних та критично оцінити ймовірність похибки ШІ в реальних умовах (наприклад, розпізнавання залишеного на стільці одягу як «присутності» учня).

Попри високу дидактичну цінність, використання Teachable Machine має низку технічних та методичних обмежень, які вчителю слід враховувати при плануванні занять. По-перше, це проблема «ілюзії розуміння». Оскільки інструмент максимально спрощує процес, в учнів може сформуватися хибне уявлення, що машинне навчання – це лише «натискання кнопок». Без належного теоретичного супроводу школярі можуть не усвідомити внутрішню логіку роботи нейронних мереж, сприймаючи результат як магічне вгадування, а не як складний статистичний аналіз вагових коефіцієнтів.

По-друге, існують суттєві технічні залежності. Сервіс критично залежить від стабільного високошвидкісного інтернет-з'єднання та обчислювальних потужностей браузера. Хоча саме навчання відбувається на серверах Google, збір даних (особливо відео високої чіткості) та подальше тестування моделі в реальному часі створюють значне навантаження на апаратне забезпечення шкільних комп'ютерів. Крім того, виникає питання конфіденційності даних: використання веб-камер та мікрофонів у класі потребує чіткого дотримання етичних протоколів та отримання дозволів на обробку біометричних даних учнів, навіть якщо обробка відбувається локально в браузері.

Нарешті, серйозним методичним викликом є вузька спеціалізація моделей. Teachable Machine обмежена лише завданнями класифікації (зображень, звуків або поз). Вона не дозволяє продемонструвати інші фундаментальні напрями ШІ, такі як регресійний аналіз, генеративні моделі (на кшталт ChatGPT чи Midjourney) або навчання з підкріпленням (reinforcement learning). Це створює ризик формування однобокого уявлення про машинне навчання. Тому вчителю важливо позиціювати цей інструмент лише як «вхідну точку», за якою має слідувати ознайомлення з більш комплексними системами та алгоритмами.

Висновки. Впровадження no-code інструментів, зокрема Google Teachable Machine, дозволяє ефективно формувати ШІ-грамотність учнів шляхом зниження «порогу входження» та зміщення акценту з написання коду на концептуальне проєктування інтелектуальних моделей. Такий підхід мінімізує когнітивне навантаження, стимулює розвиток критичного мислення та аналітичних навичок, перетворюючи школярів із пасивних споживачів на активних творців цифрових продуктів у межах STEM-освіти. Водночас успішна реалізація методики потребує збалансованого поєднання візуального моделювання з опануванням теоретичних основ для запобігання «ілюзії розуміння» технологій. Перспективи подальших досліджень полягають у розробці методик плавного переходу від візуальних моделей до професійних мов програмування (Python) та бібліотек (TensorFlow, Keras), а також в інтеграції інструментів ШІ в курси медіаграмотності для навчання учнів протидії дезінформації та критичного аналізу алгоритмічних систем.

Конфлікт інтересів. Автор підтверджує відсутність фінансових, особистих чи інших інтересів, що можуть розглядатися як потенційний конфлікт інтересів щодо публікації цієї статті.
Фінансування. Робота виконана за відсутності фінансової підтримки з боку будь-яких організацій.
Доступність даних. Це теоретичне дослідження не передбачає використання додаткових наборів даних.
Використання штучного інтелекту. Інструменти штучного інтелекту не використовувались при написанні цієї статті.

Список використаної літератури

1. Laupichler M.C., Asteria F., Schiefer J., Herzig B. Artificial intelligence literacy in higher and general education: A systematic literature review. *Computers and Education: Artificial Intelligence*. 2022. Vol.3. Article 100101. https://doi.org/10.1016/j.caeai.2022.100101
2. Long D., Magerko B. What is AI Literacy? Competencies and Design Considerations. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI 20). New York, USA: Association for Computing Machinery, 2020. P.1–16. https://doi.org/10.1145/3313831.3376727
3. Carney M., Webster B., Alvarado I., Phillips K., Howell N., Griffith J., Jongejan J., Pitaru A., Chen A. Teachable machine: approachable web-based tool for exploring machine learning classification. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA 20). New York, USA: Association for Computing Machinery, 2020. P.1–8. https://doi.org/10.1145/3334480.3382839
4. Williams R., Park H.W., Breazeal C. A is for Artificial Intelligence. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). New York, USA: Association for Computing Machinery, 2019. Paper 447, P.1–11. https://doi.org/10.1145/3290605.3300677
5. Величко В., Ганієв О., Жадан С. Штучний інтелект як інструмент аналізу і класифікації задач з програмування в освітньому процесі. Технології електронного навчання. 2024. Вип.8. С.66–73. https://doi.org/10.31865/2709-840082024316949.
6. Кривонос О.М. Використання генеративного AI для створення програмного коду. Наука і техніка сьогодні. 2024. Вип.12 (40). С.1314–1325. https://doi.org/10.52058/2786-6025-2024-12(40)-1314-1325.
7. Осадча К.П., Бармак О.В. Використання технологій штучного інтелекту для підвищення якості комп'ютерної освіти. Матеріали Міжнародної науково-практичної конференції «Сучасні стратегії інноваційного розвитку економіки» (MSIE-2023). Київ: КНУТД, 2023. С.265–267. URL: https://test.knutd.edu.ua/bitstream/123456789/26086/1/MSIE_2023_P265-267.pdf
8. Панухник О.В. Штучний інтелект в освітньому процесі та наукових дослідженнях здобувачів вищої освіти: відповідальні межі вмісту ШІ. Галицький економічний вісник. 2023. Вип.83 (4). С.202–211. https://doi.org/10.33108/galicianvisnyk_tntu2023.04.202
9. Сіциліцин Ю.О., Осадчий В.В. Можливості використання ChatGPT у дистанційному навчанні програмування початківців. Інформаційні технології і засоби навчання. 2023. Вип.97 (5). С.167–180. https://doi.org/10.33407/itlt.v97i5.5277.

References

1. Laupichler, M.C., Asteria, F., Schiefer, J., & Herzig, B. (2022). Artificial intelligence literacy in higher and general education: A systematic literature review. *Computers and Education: Artificial Intelligence*, *3*, Article 100101. https://doi.org/10.1016/j.caeai.2022.100101
2. Long, D., & Magerko, B. (2020). What is AI Literacy? Competencies and Design Considerations. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI 20)* (pp.1–16). Association for Computing Machinery. https://doi.org/10.1145/3313831.3376727
3. 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. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA 20)* (pp.1–8). Association for Computing Machinery. https://doi.org/10.1145/3334480.3382839
4. Williams, R., Park, H.W., & Breazeal, C. (2019). A is for Artificial Intelligence. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)* (pp.1–11. Paper 447). Association for Computing Machinery. https://doi.org/10.1145/3290605.3300677
5. Velychko, V., Ganiiev, O., & Zhadan, S. (2024). Artificial intelligence as a tool for analysis and classification of programming tasks in the educational process. *E-learning Technologies*, *8*, 66–73. https://doi.org/10.31865/2709-840082024316949
6. Kryvonos, O.M. (2024). Using generative AI to create program code. *Science and Technology Today*, *12* (40), 1314–1325. https://doi.org/10.52058/2786-6025-2024-12(40)-1314-1325
7. Osadcha, K.P., & Barmak, O.V. (2023). Using artificial intelligence technologies to improve the quality of computer education. *Modern Strategies for Innovative Economic Development – Proceedings of the International Scientific and Practical Conference (MSIE-2023)* (pp.265–267). https://test.knutd.edu.ua/bitstream/123456789/26086/1/MSIE_2023_P265-267.pdf
8. Panukhnyk, O.V. (2023). Artificial intelligence in the educational process and scientific research of higher education seekers: Responsible limits of AI content. *Galician Economic Bulletin*, *83*(4), 202–211. https://doi.org/10.33108/galicianvisnyk_tntu2023.04.202
9. Sytsylitsyn, Yu O., & Osadchyi, V.V. (2023). Possibilities of using ChatGPT in distance learning for beginning programmers. *Information Technologies and Learning Tools*, *97* (5), 167–180. https://doi.org/10.33407/itlt.v97i5.5277

Рукопис надійшов: 09.03.2026
Перше рішення: 08.04.2026
Доопрацьовано (раундів: 1): 14.04.2026
Прийнято до друку: 21.04.2026
Опубліковано онлайн: 30.04.2026

Kryvonos Oleksandr
PhD in Pedagogical Sciences, Associate Professor
Department of Computer Science and Information Technologies
Zhytomyr Ivan Franko State University, Zhytomyr, Ukraine

METHODOLOGY FOR DEVELOPING AI LITERACY IN STUDENTS USING NO-CODE TOOLS (BASED ON THE EXAMPLE OF GOOGLE TEACHABLE MACHINE)

Abstract. The rapid expansion of artificial intelligence (AI) necessitates a transformation of the educational paradigm, requiring the development of AI literacy as a basic competence for students. However, the high «entry threshold» caused by the complexity of the mathematical apparatus and programming language syntax creates a methodological contradiction for secondary school students. The study aims to theoretically substantiate and develop methodological recommendations for using no-code tools (specifically Google Teachable Machine) to build students' basic AI literacy in computer science education. The study employs a comparative-typological method to analyze approaches to learning IT and a systems analysis to consider the process of AI literacy development as a holistic system. The effectiveness of transfer learning on visual ML platforms is substantiated. It is proven that the no-code approach allows shifting the focus from code writing to the conceptual design of intelligent models and data analysis. The potential for integrating such models into the Scratch environment for creating contactless control systems and environmental monitoring is identified. The limitations of the tool, including the risk of an «illusion of understanding» and technical dependencies, are revealed. Successful implementation of the methodology requires a balanced combination of visual modeling with mastery of theoretical foundations to prevent the «illusion of understanding» of technologies. Prospects for further research include developing methods for a smooth transition from visual models to professional programming languages (Python) and libraries (TensorFlow, Keras), as well as integrating AI tools into media literacy courses to teach students how to counter disinformation and critically analyze algorithmic systems.

Keywords: artificial intelligence, machine learning, no-code tools, Teachable Machine, computer science education.