

УДК 004.932.2:004.85

DOI 10.31651/2076-5886-2023-1-61-68

PACS 07.05.Mh

БИЛИМЕНКО Анна Володимирівна
студентка спеціальності «Прикладна
математика» Черкаського національного
університету імені Богдана Хмельницького
e-mail: bylymenko.anna1619@vu.cdu.edu.ua

КРАСНОШЛИК Наталія Олександрівна
кандидат технічних наук, доцент, доцент
кафедри прикладної математики та
інформатики Черкаського національного
університету імені Богдана Хмельницького
e-mail: krasnoshlyk@vu.cdu.edu.ua
ORCID 0000-0003-4661-6997

ІНТЕЛЕКТУАЛЬНА СИСТЕМА ВИЗНАЧЕННЯ ПОРОДИ СОБАКИ ЗА ДОПОМОГОЮ НЕЙРОННОЇ МЕРЕЖІ

У статті розглядається розробка інтелектуальної системи автоматичного визначення породи собаки на основі згорткових нейронних мереж. Описано теоретичні засади класифікації зображень із застосуванням глибокого навчання: принципи роботи згорткових і пулінгових шарів, функції активації ReLU та Softmax, механізм зворотного поширення помилки. Проведено огляд ключових бібліотек Python, що використовувались у роботі: Pandas – для попередньої обробки даних, TensorFlow – для побудови та навчання нейронної мережі, Streamlit – для розгортання веб-інтерфейсу. Описано практичну реалізацію системи: формування навчальної вибірки з 1888 зображень 21 породи собак, архітектуру послідовної згорткової мережі з чотирма блоками «згортка – підвибірка» та двома повнозв'язними шарами, а також процес навчання протягом 30 epoch. За результатами тестування досягнуто точність класифікації 93.64%. Веб-додаток розгорнуто на платформі Streamlit Cloud з використанням репозиторію GitHub. Виконано серії практичних тестувань, що підтвердили працездатність системи для порід, представлених у навчальній вибірці, та прогнозовану поведінку для відсутніх порід. Запропонований підхід може бути поширений на інші задачі класифікації зображень у галузі комп'ютерного зору.

Ключові слова: згорткова нейронна мережа, машинне навчання, класифікація зображень, розпізнавання порід собак, TensorFlow, Streamlit, глибоке навчання, веб-додаток, штучний інтелект.

Вступ

Штучні нейронні мережі, зокрема згорткові, зарекомендували себе як потужний інструмент для аналізу зображень. Їхні алгоритми, натхненні біологічними процесами, дозволяють виявляти складні візерунки в даних. Згорткові мережі послідовно обробляють зображення, витягуючи все більш абстрактні ознаки.

Мета статті – теоретично обґрунтувати застосування згорткових нейронних мереж для розпізнавання порід собак.

Виклад основного матеріалу

1. Класифікація зображень за допомогою згорткових нейронних мереж

Штучний інтелект, особливо машинного навчання, досяг значних висот завдяки розвитку згорткових нейронних мереж. Ці мережі, натхненні біологічними процесами, виявилися надзвичайно ефективними для аналізу зображень та інших типів даних.

Однією з ключових особливостей згорткових мереж є їх здатність автоматично виявляти важливі ознаки в даних. Завдяки процесу зворотного поширення помилки,

мережа вчиться налаштовувати свої параметри, щоб краще розпізнавати ці ознаки. Це дозволяє їй виконувати різноманітні завдання, від класифікації зображень до розпізнавання мови.[1].

Важливу роль у згорткових мережах відіграють згорткові шари, які застосовують фільтри для виявлення різних ознак. Пулінгові шари, в свою чергу, зменшують розмірність даних і запобігають перенавчанню. Функція активації ReLU додає нелінійність в мережу, що дозволяє їй моделювати складні залежності.

Архітектура згорткових мереж зазвичай складається з чергування згорткових і пулінгових шарів, за якими можуть йти повнозв'язні шари для класифікації. Вибір розміру фільтрів, кількості шарів та інших параметрів мережі є важливим завданням і впливає на її продуктивність.

Згорткові нейронні мережі знайшли широке застосування в різних галузях, таких як комп'ютерний зір, обробка природної мови, медична діагностика та багато інших. Їхньою перевагою є здатність автоматично витягувати інформативні ознаки з великих обсягів даних.

Ключові переваги згорткових нейронних мереж:

- Автоматичне виявлення ознак.
- Інваріантність до зсувів.
- Ефективність обробки великих обсягів даних.

Застосування:

- Розпізнавання образів (облич, об'єктів)
- Сегментація зображень
- Обробка природної мови
- Медична діагностика
- Автономні автомобілі

Згорткові нейронні мережі є одним з найпотужніших інструментів сучасного машинного навчання і продовжують розвиватися, відкриваючи нові можливості для вирішення складних задач.

2. Задача класифікації

Задача класифікації в контексті машинного навчання полягає в тому, щоб призначити вхідному об'єкту або даним певний "клас" або "мітку" з попередньо визначеного набору можливих класів.

Для виконання задачі класифікації необхідно мати набір прикладів даних, для яких відомі правильні класифікації або мітки. Ці дані використовуються для тренування моделі машинного навчання, яка вивчає зв'язки між вхідними ознаками та відповідними класами.

Після тренування моделі вона може бути застосована до нових, раніше не використаних даних для прогнозування класу, до якого вони належать. Задача класифікації має широке застосування в багатьох галузях, включаючи комп'ютерне зорове сприйняття, медицину, фінанси, рекламу, безпеку, та багато інших областей, де необхідно приймати рішення на основі вхідних даних. [2].

3. Огляд бібліотек для реалізації нейронних мереж і для створення веб-додатків та допоміжних інструментів

На сьогоднішній день, завдяки популярності нейронних мереж на мові Python існує велика кількість бібліотек, які надають можливість конструювання різноманітних нейронних мереж на досить різних рівнях абстракції. Для огляду було обрано три бібліотеки: Pandas, Tensorflow та Streamlit.

Pandas – це високорівнева Python бібліотека для аналізу даних. Вона побудована поверх більш низькорівневої бібліотеки NumPy, що додає їй продуктивності. Ця бібліотека є одною з найбільш просунутих, що швидко розвивається, для обробки та аналізу даних. *Pandas* – одна із дуже зручних бібліотек всередині мови Python, які допомагають працювати з великою кількістю табличних даних (досить часто тренувальні і тестувальні вибірки виглядають, як .csv-таблиці мільйонами рядків і колонок параметрів) є бібліотека *Pandas*. Вона дозволяє дуже швидко завантажувати дані, препроцесити їх (готувати у відповідний формат), щоб у зручному вигляді відправляти на опрацювання алгоритмом, який, наприклад, вибрали з бібліотеки *scikit-learn*.

Бібліотека *Pandas* пропонує дві основні структури даних: *DataFrame* і *Series*. *DataFrame* – це двовимірна структура даних у вигляді таблиці, яка складається зі стовпців із позначеними осями для роботи з табличними даними. Крім того, *Series* є одновимірним позначеним масивом об'єкта, який містить значення з позначеними осями.

Бібліотека *Pandas* може бути корисною в реальних додатках, включаючи фінансове моделювання, наукові обчислення, аналіз даних, візуалізацію даних, та машинне навчання.

Бібліотека *Pandas* надає розробникам простий у використанні інструмент аналізу даних для виконання маніпуляцій з даними, їх фільтрації та обробки в Python. Здатність бібліотеки обробляти величезну кількість даних, повна документація, та велика кількість функцій, роблять її потужним інструментом для аналізу даних.

Дана бібліотека – це *opensource* бібліотека, тобто вихідний код є у відкритому доступі та розміщений на GitHub. Користувачі можуть додавати туди свій код: вносити зміни, доповнювати методи власним кодом, оновлювати розділи, тощо.

TensorFlow – це бібліотека з відкритим кодом, яка дозволяє розробникам і дослідникам створювати та розгортати різні моделі машинного навчання та системи глибокого навчання. Вона розроблена Google і є однією з найпопулярніших бібліотек, що використовуються в програмах машинного навчання та штучного інтелекту.

Однією з ключових особливостей *TensorFlow* є здатність обробляти великі обсяги даних. Бібліотека підтримує розподілене обчислення, дозволяючи обробляти великі набори даних на кількох машинах, що скорочує час, необхідний для навчання моделей. Крім того, *TensorFlow* має широкий спектр готових моделей, наприклад як модель розпізнавання зображень і мови.

Іншою важливою особливістю *TensorFlow* є те, що вона підтримує кілька платформ, включаючи мобільні та вбудовані пристрої, а також пропонує підтримку різних мов програмування, включаючи Python, C++ і Java. Ця бібліотека також надає широкий набір інструментів для розгортання моделей у різних середовищах.

На відміну від інших бібліотек DL, які в основному орієнтовані на дослідження (наприклад, Theano), *TensorFlow* був розроблений для використання як у системах досліджень, так і у розробці та виробництві програмного забезпечення.

Streamlit – це бібліотека Python з відкритим вихідним кодом, яка дозволяє легко створювати та ділитися чудовими спеціальними веб-програмами для машинного навчання та обробки даних. Лише за кілька хвилин ви можете створити та розгорнути потужну програму для обробки даних. [3].

Три найважливіші функції, які надає *Streamlit*:

1. Використання сценаріїв: можна створити інтерактивну веб-програму за допомогою лише кількох рядків коду, і якщо внести будь-які зміни до свого коду, то побачите автоматичні оновлення у веб-додатку. Це відбувається з API, що надається цим фреймворком.

2. Взаємодія: додати віджети до вашого веб-додатку так само просто, як оголосити змінні в Python.
3. Миттєве розгортання: платформа Streamlit допомагає легко розгорнути програми та керувати ними. [4].

Крім того, Streamlit можна використовувати з популярними бібліотеками Python, такими як NumPy, Pandas і Matplotlib, що робить її універсальним інструментом для проектів з обробки даних.

Streamlit також може легко обробляти великі набори даних і складні обчислення. Її можна легко розгорнути на різних хмарних платформах, включаючи Amazon Web Services (AWS), Google Cloud Platform (GCP) і Microsoft Azure. Streamlit – це чудовий інструмент для розробників, яким потрібен стислий, спрощений та ефективний спосіб створення веб-додатків для аналізу даних, машинного навчання та візуалізації даних.

Основна мета Streamlit – це спростити процес розробки веб-додатків і усунути складність багатьох мов, фреймворків, тощо. За допомогою Streamlit розробники можуть створювати власні веб-інтерфейси зі свого коду Python без використання HTML, CSS або JavaScript.

Однією з важливих особливостей Streamlit є його механізм кешування, який дозволяє користувачам кешувати обчислення та використовувати їх належним чином. Ще однією значною перевагою використання Streamlit є його універсальність у створенні різних типів веб-додатків, керованих даними.

Streamlit є здатність перетворювати статичні блокноти Jupyter в інтерактивні веб-додатки. Ноутбуки Jupyter широко використовуються в машинному навчанні для відтворення коду, спільного використання, та співпраці. Однак Jupyter Notebooks може бути важко розгорнути та поділитися з нетехнічними зацікавленими сторонами. Streamlit вирішує цю проблему, дозволяючи конвертувати Jupyter Notebooks в інтерактивні веб-програми за допомогою кількох простих команд.

Streamlit працює на всіх основних операційних системах, таких як Windows, macOS і Linux, і підтримує Python 3.6 і вище.

Підсумовуючи, Streamlit забезпечує універсальний спосіб створення веб-додатків з мінімальними зусиллями та чудовими варіантами розгортання. Саме бібліотеку Streamlit було обрано для створення веб-додатку.

4. Практична реалізація системи розпізнавання порід собак

4.1. Підготовка набору даних

Першим етапом практичної реалізації є формування навчальної вибірки. Для автоматизованого збору зображень використано бібліотеку Python `bing-image-downloader`, яка здійснює асинхронне завантаження зображень із пошукового сервісу Bing за заданим запитом. Команда запуску має вигляд:

```
python test.py <назва_породи>
```

Для кожної з 21 породи собак було завантажено близько 100 зображень. Загальний обсяг сформованого набору даних склав 1888 файлів. Після завантаження виконано ручну фільтрацію нерелевантних зображень та уніфікацію форматів файлів до вимог Google Colaboratory (JPEG, PNG, GIF, BMP). Підготовлений набір даних завантажено на Google Диск для подальшого використання у середовищі Colaboratory. Підключення Google Діску виконується наступним кодом:

```
from google.colab import drive
drive.mount('/content/drive')
```

Набір даних розподілено на навчальну (90%) та перевірочну (10%) вибірки за допомогою функції `image_dataset_from_directory` з параметром `validation_split=0.1`. Розмір вхідних зображень уніфіковано до 100×100 пікселів.

4.2. Архітектура та навчання нейронної мережі

Для класифікації порід собак побудовано згорткову нейронну мережу послідовної архітектури (Sequential). Мережа складається з чотирьох блоків «згортковий шар – шар підвибірки» з фільтрами розмірністю 5×5 та кількістю карт ознак 16, 32, 64 і 128 відповідно, після яких розташовано два повнозв'язні шари з 1024 та 256 нейронами. У всіх згорткових та повнозв'язних шарах використано функцію активації ReLU; після кожного повнозв'язного шару застосовано Dropout з коефіцієнтом 0.2 для запобігання перенавчанню. Вихідний шар містить 21 нейрон (за кількістю класів) з функцією активації Softmax. Зведена архітектура мережі подана в табл. 1.

Таблиця 1

Архітектура згорткової нейронної мережі

Шар	Тип	Параметри
1	Conv2D	16 фільтрів, 5×5, ReLU
2	MaxPooling2D	2×2
3	Conv2D	32 фільтри, 5×5, ReLU
4	MaxPooling2D	2×2
5	Conv2D	64 фільтри, 5×5, ReLU
6	MaxPooling2D	2×2
7	Conv2D	128 фільтрів, 5×5, ReLU
8	MaxPooling2D	2×2
9	Dense + Dropout	1024 нейрони, p=0.2
10	Dense + Dropout	256 нейронів, p=0.2
11	Dense (вихідний)	21 нейрон, Softmax

Навчання виконувалось протягом 30 епох з використанням навчальної та перевірочної вибірок. Оцінку якості навченої моделі здійснено на тестовому наборі за допомогою методу `model.evaluate`. За результатами тестування точність класифікації склала 93.64%, що підтверджує прийнятну якість навченої моделі для практичного застосування (рис. 1). Динаміка частки правильних відповідей на навчальній та перевірочній вибірках упродовж 30 епох навчання свідчить про стійке зростання точності моделі без ознак суттєвого перенавчання (рис. 2).

Навчену модель збережено у форматі HDF5:

```
model.save("dog_breeds.h5")
```

4.3. Розгортання веб-додатку

Для розгортання веб-інтерфейсу використано бібліотеку Streamlit. Основний файл застосунку `web-for-classification.py` реалізує наступну логіку: завантаження збереженої моделі; попередню обробку завантаженого користувачем зображення (зміна розміру до 100×100 пікселів, нормалізація); передачу оброблених даних до моделі та виведення п'яти найбільш імовірних порід із зазначенням ймовірності кожної з них. Ключовий фрагмент інтерфейсу виглядає наступним чином:

```
st.title('Класифікація зображень')
img = load_image()
```

```
result = st.button('Розпізнати зображення')
if result:
    x = preprocess_image(img)
    preds = model.predict(x)
    print_predictions(preds)
```

```
[ ] # Оцінюємо якість навчання моделі на тестових даних
scores = model.evaluate(test_dataset, verbose=1)
```

```
8/8 [=====] - 17s 30ms/step - loss: 0.3898 - accuracy: 0.9364
```

```
[ ] print("Частка правильних відповідей на тестових даних, у відсотках:", round(scores[1] * 100, 4))
```

```
Частка правильних відповідей на тестових даних, у відсотках: 93.6441
```

Рис. 1. Оцінка якості отриманих результатів

```
plt.plot(history.history['accuracy'],
         label='Частка правильних відповідей на навчальному наборі')
plt.plot(history.history['val_accuracy'],
         label='Частка правильних відповідей на перевірконому наборі')
plt.xlabel('Епоха навчання')
plt.ylabel('Частка правильних відповідей')
plt.legend()
plt.show()
```

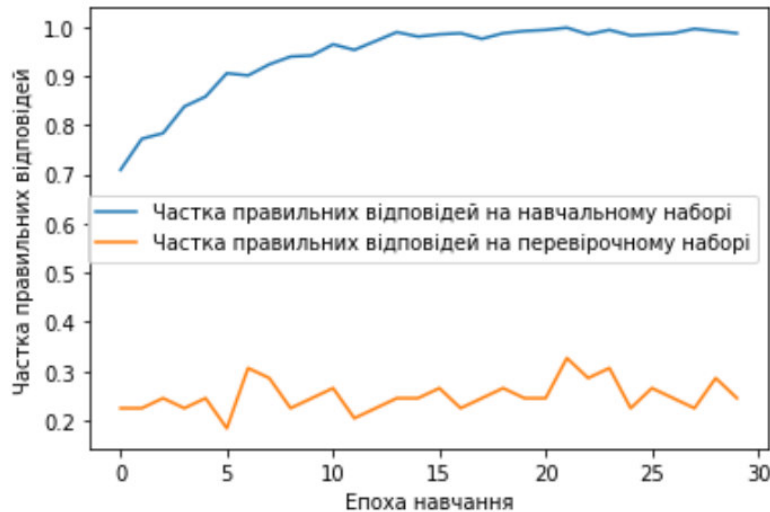


Рис. 2. Динаміка зміни частки правильних відповідей

Публікацію застосунку виконано через платформу Streamlit Cloud з використанням репозиторію GitHub. До репозиторію завантажено файли: `web-for-classification.py` (код застосунку), `dog_breeds.h5` (навчена модель) та `requirements.txt` (перелік залежностей). Розгортання здійснюється через вебінтерфейс Streamlit Cloud після підключення облікового запису GitHub та вказівки шляху до основного файлу. Зовнішній вигляд розгорнутого веб-додатку наведено на рис. 3.

Для перевірки практичної ефективності системи було виконано три серії тестувань: розпізнавання зображень із навчального набору (точність наближається до 100%), розпізнавання нових зображень порід, що входять до набору даних (хаскі розпізнано з імовірністю 94.08%; джек-расел-тер'єр – з імовірністю 99.98%), та розпізнавання породи, відсутньої в наборі даних (бультер'єр). В останньому випадку модель розподілила ймовірність між візуально схожими породами (бульдог – 41.33%, такса – 40.67%), що є прогнозованою поведінкою системи з фіксованим набором класів.

Висновки

Згорткові нейронні мережі є ефективним інструментом для розпізнавання порід собак. Завдяки здатності автоматично виявляти характерні риси об'єктів на зображеннях, ці мережі досягають високої точності класифікації. Для реалізації такого розпізнавання було використано сучасні бібліотеки Python, такі як TensorFlow та Streamlit, які забезпечили зручність розробки та гнучкість у налаштуванні моделей. Запропонований підхід може бути застосований для створення інших веб-додатків, що базуються на технологіях глибокого навчання, та відкриває широкі перспективи для розвитку інноваційних рішень.

Список використаної літератури:

1. Згорткова нейронна мережа. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Згорткова_нейронна_мережа (дата звернення: 01.10.2023).
2. Patterson J., Gibson A. Deep Learning: A Practitioner's Approach. Sebastopol : O'Reilly Media, 2017. 520 p.
3. Streamlit Documentation. URL: <https://docs.streamlit.io/> (дата звернення: 01.10.2023).
4. LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998. Vol. 86, No. 11. P. 2278–2324. DOI: 10.1109/5.726791.
5. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems. 2012. Vol. 25. P. 1097-1105.
6. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556. 2014. URL: <https://arxiv.org/abs/1409.1556>.
7. Khosla A., Jayadevaprakash N., Yao B., Fei-Fei L. Novel Dataset for Fine-Grained Image Categorization. Proceedings of the 1st Workshop on Fine-Grained Visual Categorization (FGVC), CVPR. Colorado Springs, 2011.
8. Parkhi O. M., Vedaldi A., Zisserman A., Jawahar C. V. Cats and Dogs. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Providence, 2012. P. 3498-3505. DOI: 10.1109/CVPR.2012.6248092.

References:

1. Convolutional neural network. Wikipedia. URL: https://en.wikipedia.org/wiki/Convolutional_neural_network (accessed: 01.10.2023).
2. Patterson J., Gibson A. Deep Learning: A Practitioner's Approach. Sebastopol: O'Reilly Media, 2017. 520 p.
3. Streamlit Documentation. URL: <https://docs.streamlit.io/> (accessed: 01.10.2023).
4. LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998. Vol. 86, No. 11. P. 2278–2324. DOI: 10.1109/5.726791.
5. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems. 2012. Vol. 25. P. 1097-1105.
6. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556. 2014. URL: <https://arxiv.org/abs/1409.1556>.
7. Khosla A., Jayadevaprakash N., Yao B., Fei-Fei L. Novel Dataset for Fine-Grained Image Categorization. Proceedings of the 1st Workshop on Fine-Grained Visual Categorization (FGVC), CVPR. Colorado Springs, 2011.
8. Parkhi O. M., Vedaldi A., Zisserman A., Jawahar C. V. Cats and Dogs. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Providence, 2012. P. 3498–3505. DOI: 10.1109/CVPR.2012.6248092.

BYLYMENKO Anna,

Student, Department of Informatics and Applied Mathematics, The Bohdan Khmelnytsky National

University of Cherkasy, Ukraine

KRASNOSHLYK Nataliya,

Candidate of Technical Sciences, Associate Professor, Department of Applied Mathematics and Informatics, The Bohdan Khmelnytsky National University of Cherkasy, Ukraine

AN INTELLIGENT SYSTEM FOR DOG BREED RECOGNITION USING CONVOLUTIONAL NEURAL NETWORKS

Summary. Introduction. Convolutional neural networks (CNNs) have established themselves as a powerful tool for image analysis. Inspired by biological visual processing, they automatically detect complex patterns in data by sequentially extracting increasingly abstract features. The growing demand for AI-powered interactive applications makes automated image recognition systems particularly relevant across a wide range of domains, including veterinary practice, animal shelters, and mobile applications.

The Purpose of the article is to theoretically justify and practically demonstrate the use of convolutional neural networks for dog breed identification, and to describe the full development cycle of a CNN-based web application – from dataset preparation to deployment.

Results. The article presents a review of image classification methods based on deep learning, with a focus on the architecture and training principles of CNNs. The roles of convolutional layers, pooling layers, the ReLU activation function, and the backpropagation algorithm are discussed. Three Python libraries central to the project are reviewed: Pandas for data preprocessing, TensorFlow for building and training the neural network model, and Streamlit for rapid development and deployment of the web interface.

The practical implementation covers all stages of system development. A dataset of 1,888 images across 21 dog breeds was collected using the `bing-image-downloader` library and manually filtered. The images were split into training (90%) and validation (10%) sets. The CNN architecture follows a sequential structure consisting of four convolutional-pooling blocks with 16, 32, 64, and 128 filters respectively, followed by two fully connected layers (1,024 and 256 neurons) with Dropout regularization ($p = 0.2$), and a Softmax output layer with 21 neurons. The model was trained for 30 epochs and achieved a classification accuracy of 93.64% on the test set. Practical testing confirmed high accuracy for breeds present in the training data (e.g., Husky – 94.08%, Jack Russell Terrier – 99.98%) and predictable probabilistic distribution for breeds outside the dataset.

The web application was deployed via Streamlit Cloud integrated with a GitHub repository, providing a user-friendly interface for uploading images and displaying the top five predicted breeds with corresponding probabilities.

Conclusion. Convolutional neural networks are an effective tool for dog breed classification tasks. The combination of TensorFlow for model development and Streamlit for interface deployment provides an efficient and accessible pipeline for building machine learning-based web applications. The proposed approach can be readily extended to other image recognition tasks and illustrates the practical potential of deep learning in real-world automated recognition systems.

Keywords: convolutional neural network, machine learning, image classification, dog breed recognition, TensorFlow, Streamlit, deep learning, web application, artificial intelligence.

Одержано редакцією 03.11.2023 р.
Прийнято до публікації 06.12.2023 р.