

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ПІДБОРУ ГІПЕРПАРАМЕТРІВ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ РОЗПІЗНАВАННЯ РУКОПИСНИХ ЦИФР MNIST

Вінницький національний технічний університет

Анотація

У роботі досліджено ефективність застосування згорткової нейронної мережі для розпізнавання рукописних цифр на основі набору даних MNIST/Digit Recognizer. Для експериментального дослідження використано табличне представлення зображень розміром 28×28 пікселів, де кожен об'єкт належить до одного з десяти класів — цифр від 0 до 9. Виконано завантаження даних, нормалізацію значень пікселів, зміну форми вхідних ознак, one-hot encoding цільових міток, а також аугментацію зображень. Побудовано згорткову нейронну мережу на основі бібліотеки Keras/TensorFlow. Проведено порівняння двох стратегій підбору гіперпараметрів: Random Search та Genetic Algorithm. Встановлено, що Random Search забезпечує швидший підбір параметрів, тоді як Genetic Algorithm дозволяє отримати вищу фінальну точність моделі. Найкраща модель, знайдена за допомогою генетичного алгоритму, досягла валідаційної точності 0,99429, що підтверджує ефективність використання еволюційних методів для оптимізації гіперпараметрів нейронних мереж.

Ключові слова: MNIST, згорткова нейронна мережа, CNN, Random Search, Genetic Algorithm, Keras, TensorFlow, розпізнавання цифр, машинне навчання.

Abstract

The paper investigates the effectiveness of a convolutional neural network for handwritten digit recognition using the MNIST/Digit Recognizer dataset. A tabular representation of 28×28 pixel images was used, where each object belongs to one of ten classes corresponding to digits from 0 to 9. Data loading, pixel normalization, input reshaping, one-hot encoding of target labels, and image augmentation were performed. A convolutional neural network was built using the Keras/TensorFlow framework. Two hyperparameter optimization strategies were compared: Random Search and Genetic Algorithm. The results showed that Random Search provides faster hyperparameter tuning, while Genetic Algorithm achieves higher final model accuracy. The best model found using the genetic algorithm achieved a validation accuracy of 0.99429, confirming the effectiveness of evolutionary methods for optimizing neural network hyperparameters.

Keywords: MNIST, convolutional neural network, CNN, Random Search, Genetic Algorithm, Keras, TensorFlow, digit recognition, machine learning..

Вступ

Сучасні інформаційні технології та методи машинного навчання широко застосовуються для автоматизованого аналізу зображень, розпізнавання символів, комп'ютерного зору та інтелектуальної обробки візуальної інформації. Однією з базових і водночас важливих задач у цій сфері є розпізнавання рукописних цифр, оскільки вона дозволяє дослідити принципи роботи нейронних мереж, методів класифікації та оптимізації параметрів моделей.

Для задач розпізнавання зображень особливо ефективними є згорткові нейронні мережі (Convolutional Neural Networks, CNN), які здатні автоматично виділяти локальні просторові ознаки зображення. На відміну від класичних алгоритмів машинного навчання, CNN не потребує ручного формування ознак, оскільки згорткові шари самостійно виявляють характерні елементи зображення, такі як контури, кути, лінії та складніші візуальні структури [2].

Важливим етапом побудови нейронної мережі є підбір гіперпараметрів, які впливають на швидкість навчання, якість класифікації та здатність моделі до узагальнення. До таких параметрів належать швидкість навчання, розмір пакета даних та кількість нейронів у повнозв'язному шарі. У даній роботі досліджується ефективність двох стратегій пошуку гіперпараметрів: Random Search та Genetic Algorithm [3, 4].

Метою роботи є дослідження ефективності підбору гіперпараметрів згорткової нейронної мережі для задачі розпізнавання рукописних цифр MNIST та порівняння результатів методів Random Search і Genetic Algorithm.

Основна частина

Для проведення дослідження було використано набір даних Digit Recognizer з платформи Kaggle, який є табличним представленням набору MNIST [1]. Кожен запис містить мітку класу label, що відповідає цифрі від 0 до 9, а також 784 ознаки pixel0–pixel783, які описують інтенсивності пікселів зображення розміром 28×28. Загальний обсяг навчального набору становить 42 000 зображень і 785 стовпців, з яких один стовпець є цільовою змінною, а решта — піксельними ознаками.

На початковому етапі було завантажено файл train.csv у середовищі Kaggle та створено DataFrame за допомогою бібліотеки pandas [5]. Перевірка структури набору даних показала, що кожен об'єкт має числове представлення, придатне для подальшої обробки та використання у згортковій нейронній мережі.

```
Завантажений файл: /kaggle/input/digit-recognizer/train.csv
Розмір датасету: (42000, 785)
```

label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	...	pixel774	pixel775	pixel776	pixel777	pixel778	pixel779	pixel780	pixel781	pixel782	pixel783
0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	4	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 785 columns

Рис. 1. Приклад записів набору даних Digit Recognizer

Рисунок 1 демонструє фрагмент початкового набору даних. У першому стовпці наведено правильну мітку класу, а наступні стовпці містять значення пікселів. Такий формат даних дозволяє виконати перетворення табличного вектора довжиною 784 у зображення розміром 28×28 пікселів.

Після завантаження даних було виконано відокремлення вхідних ознак і цільових міток. Значення пікселів було нормалізовано шляхом ділення на 255, що дозволило привести їх до діапазону [0; 1]. Нормалізація є важливим етапом підготовки даних, оскільки вона стабілізує процес навчання нейронної мережі та пришвидшує збіжність алгоритму оптимізації [2].

Далі вхідні дані було перетворено з векторного вигляду у тензори форми (28, 28, 1), що відповідає формату зображень у відтінках сірого. Цільові мітки було перетворено у формат one-hot encoding, оскільки задача є багатокласовою класифікацією з десятьма класами. Після цього дані було поділено на навчальну та валідаційну вибірки у співвідношенні 80/20 із використанням стратифікації. У результаті було сформовано 33 600 навчальних зображень та 8 400 валідаційних зображень.

```
Форма X_train: (33600, 28, 28, 1)
Форма X_val: (8400, 28, 28, 1)
Форма y_train_raw: (33600,)
Форма y_val_raw: (8400,)
Форма y_train (one-hot): (33600, 10)
Форма y_val (one-hot): (8400, 10)
```

Рис. 2. Результат підготовки даних для навчання CNN-моделі

Рисунок 2 показує форми навчальної та валідаційної вибірок після попередньої обробки. Отримані розміри підтверджують, що дані коректно підготовлені для подання на вхід згорткової нейронної мережі.

Для перевірки правильності перетворення даних було виконано візуалізацію кількох випадкових зображень із навчальної вибірки. Це дозволило переконатися, що початкові числові вектори коректно відновлюються у вигляді зображень рукописних цифр.

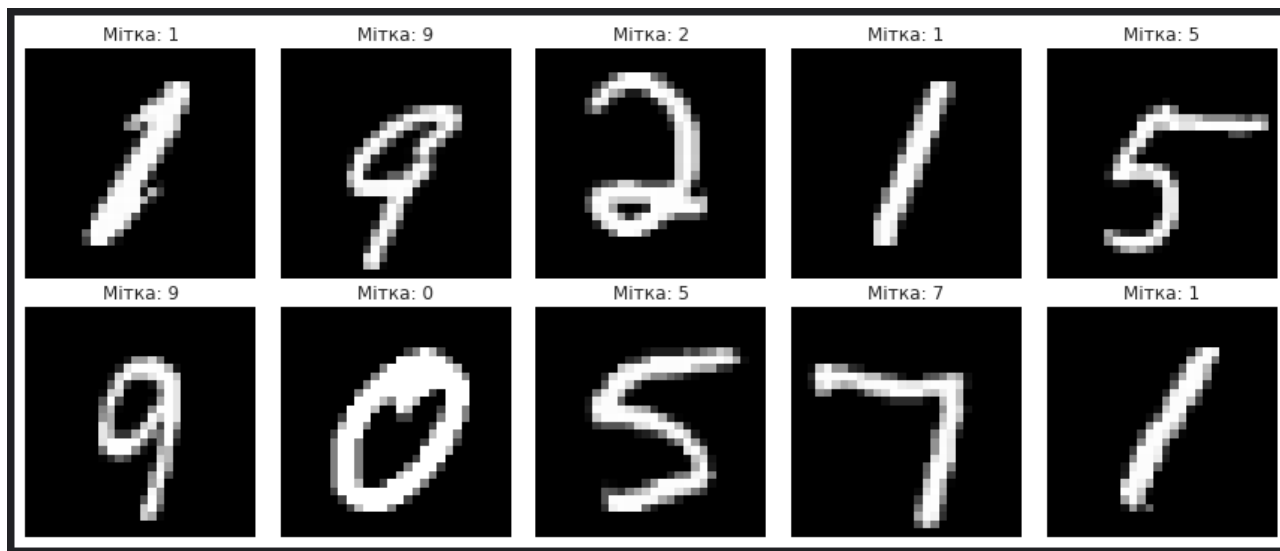


Рис. 3. Приклади рукописних цифр із навчальної вибірки

Рисунок 3 демонструє приклади зображень цифр після перетворення вхідних даних у формат 28×28 . Кожне зображення має відповідну мітку класу. Така візуальна перевірка є важливою, оскільки дозволяє переконатися у відсутності помилок під час нормалізації та зміни форми даних.

Для розв'язання задачі класифікації було побудовано згорткову нейронну мережу засобами бібліотеки Keras/TensorFlow [2]. Архітектура моделі включає два згорткові блоки з шарами Conv2D та Batch Normalization, операції MaxPooling для зменшення розмірності карт ознак, Dropout для зниження ризику перенавчання, а також повнозв'язні Dense-шари для фінальної класифікації. На виході моделі використано шар із 10 нейронами та функцією активації softmax, що дозволяє отримати ймовірності належності зображення до кожного класу.

Для підвищення здатності моделі до узагальнення було застосовано аугментацію зображень за допомогою ImageDataGenerator. Використовувалися невеликі повороти, масштабування та зсуви зображень. Такий підхід дозволяє штучно розширити навчальну вибірку та зробити модель більш стійкою до варіативності рукописного написання цифр.

У роботі досліджено два підходи до підбору гіперпараметрів. Першим підходом є Random Search, який випадково обирає певну кількість комбінацій із загального простору пошуку. У дослідженні розглядалися три гіперпараметри: швидкість навчання, розмір пакета та кількість нейронів у першому Dense-шарі. Загальний простір пошуку складався з 60 можливих комбінацій, з яких Random Search перевіряв 8 варіантів.

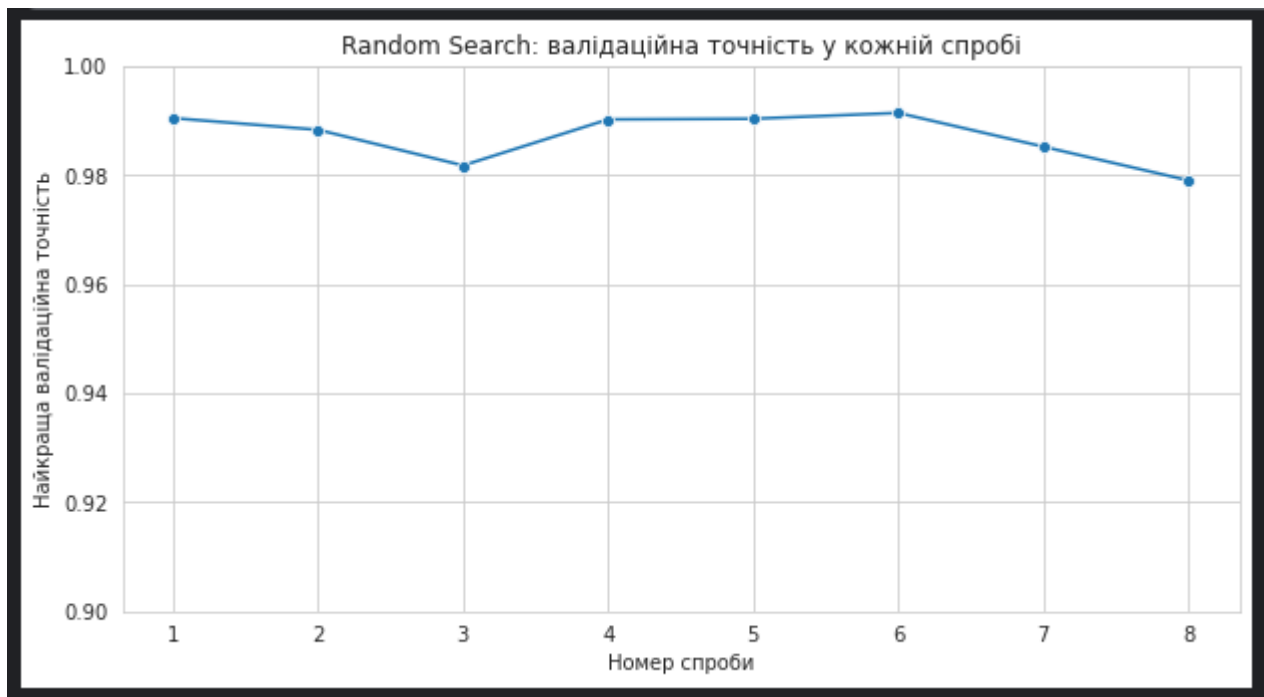


Рис. 4. Результати Random Search за валідаційною точністю

Рисунок 4 показує зміну найкращої валідаційної точності для кожної спроби Random Search. За результатами пошуку найкращими гіперпараметрами стали: `learning_rate = 0.005`, `batch_size = 128`, `dense_units = 64`. Перевагою цього методу є простота реалізації та відносно невеликі часові витрати.

Другим підходом було використано Genetic Algorithm — генетичний алгоритм без зовнішніх бібліотек. У цьому випадку кожна особина кодувала один набір гіперпараметрів. Генетичний алгоритм використовував популяцію з 6 особин, 4 покоління, 3 батьківські особини та ймовірність мутації 0,30. На відміну від випадкового пошуку, генетичний алгоритм поступово покращує популяцію за рахунок відбору, кросоверу та мутації [4].

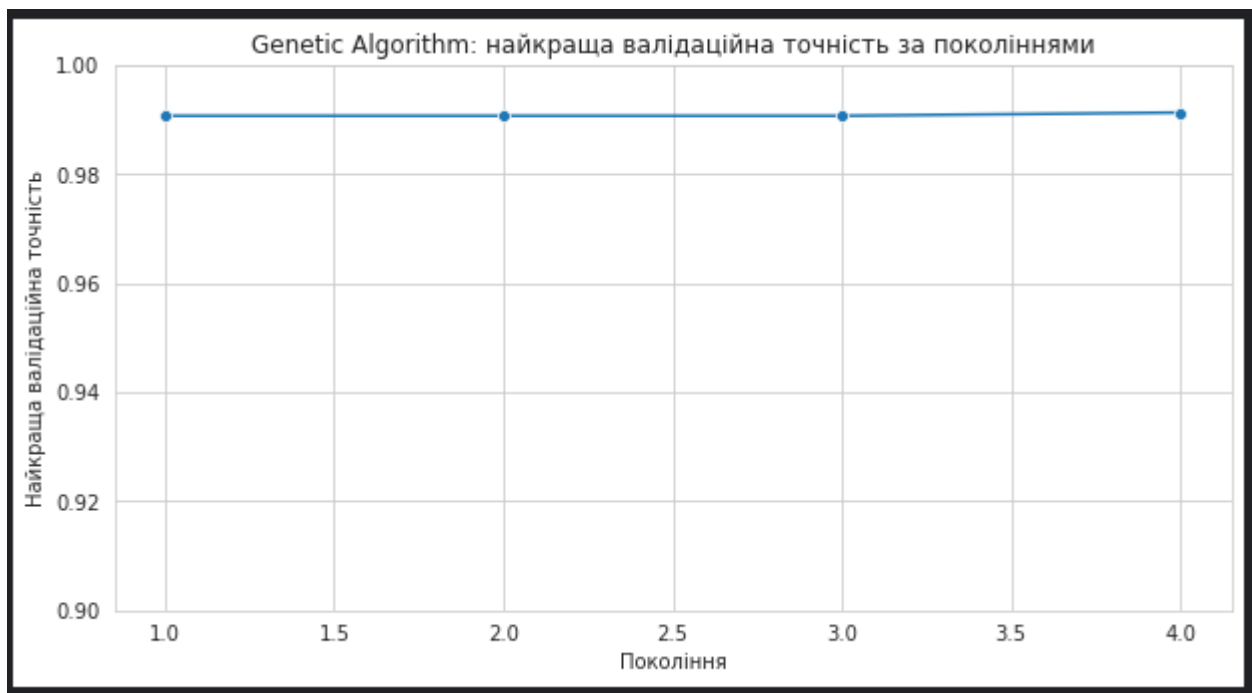


Рис. 5. Зміна найкращої валідаційної точності за поколіннями генетичного алгоритму

Рисунок 5 демонструє, як змінювалася найкраща валідаційна точність у процесі роботи генетичного алгоритму. Найкраща конфігурація, знайдена цим методом, мала параметри `learning_rate = 0.001`, `batch_size = 64`, `dense_units = 128`. Отримані результати показують, що еволюційний пошук здатний знаходити якісніші конфігурації за рахунок поступового вдосконалення популяції.

Після завершення короткого етапу підбору гіперпараметрів найкращі конфігурації Random Search та Genetic Algorithm було навчено повторно протягом більшої кількості епох. Це дозволило перевірити, чи зберігається перевага знайдених параметрів після повнішого навчання моделі. За результатами порівняння Random Search витратив 357,47 секунди на підбір гіперпараметрів і досяг фінальної валідаційної точності 0,99143. Genetic Algorithm витратив 578,80 секунди, однак забезпечив вищу фінальну точність — 0,99333.

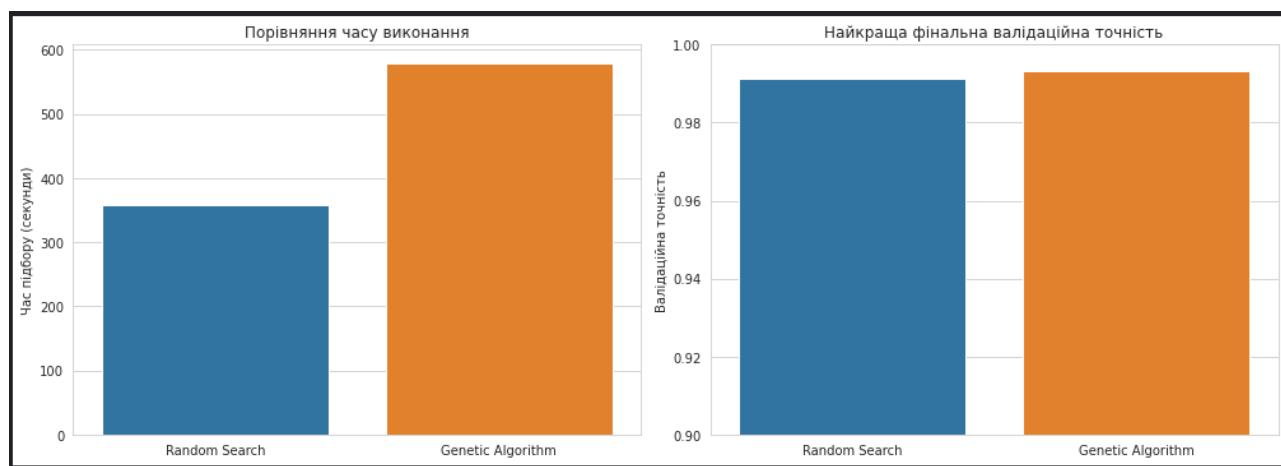


Рис. 6. Порівняння Random Search та Genetic Algorithm за часом і фінальною точністю

Рисунок 6 ілюструє компроміс між обчислювальними витратами та якістю моделі. Random Search виявився швидшим на 221,32 секунди, тобто приблизно на 38,24% менш витратним за часом. Водночас Genetic Algorithm забезпечив вищу фінальну точність після повнішого навчання. Це свідчить про те, що Random Search доцільно використовувати для швидкого отримання базового результату, тоді як Genetic Algorithm є ефективнішим для пошуку якіснішої конфігурації при наявності більшого часу на обчислення.

Для більш детального аналізу було побудовано криві валідаційної точності фінальних моделей. Вони дозволяють оцінити не лише кінцевий результат, а й характер збіжності моделей під час навчання.

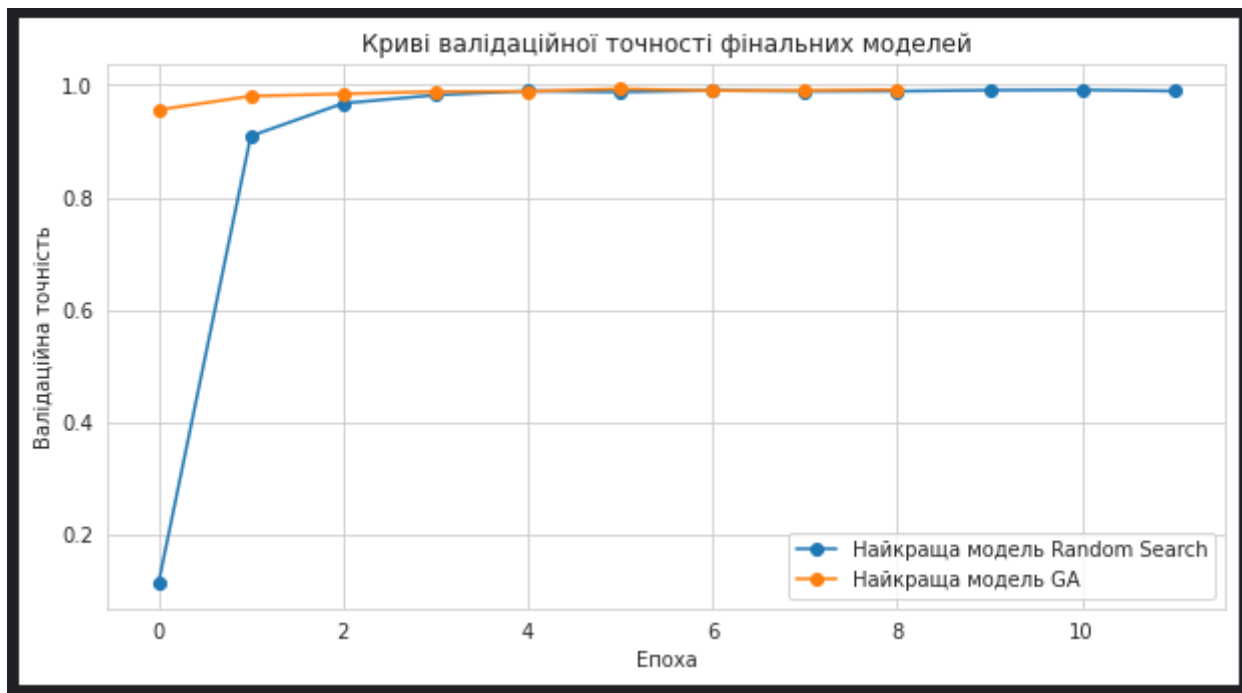


Рис. 7. Криві валідаційної точності фінальних моделей

Рисунок 7 показує зміну валідаційної точності моделей, отриманих після застосування Random Search та Genetic Algorithm. Аналіз кривих навчання дозволяє оцінити стабільність моделі, швидкість досягнення високої точності та наявність можливих ознак перенавчання.

Остаточне оцінювання було виконано для найкращої моделі, знайденої за допомогою Genetic Algorithm. Повторне навчання цієї конфігурації забезпечило валідаційну точність 0,99429. Додатково було побудовано classification report і матрицю помилок, що дозволило оцінити якість класифікації для кожного з десяти класів окремо.

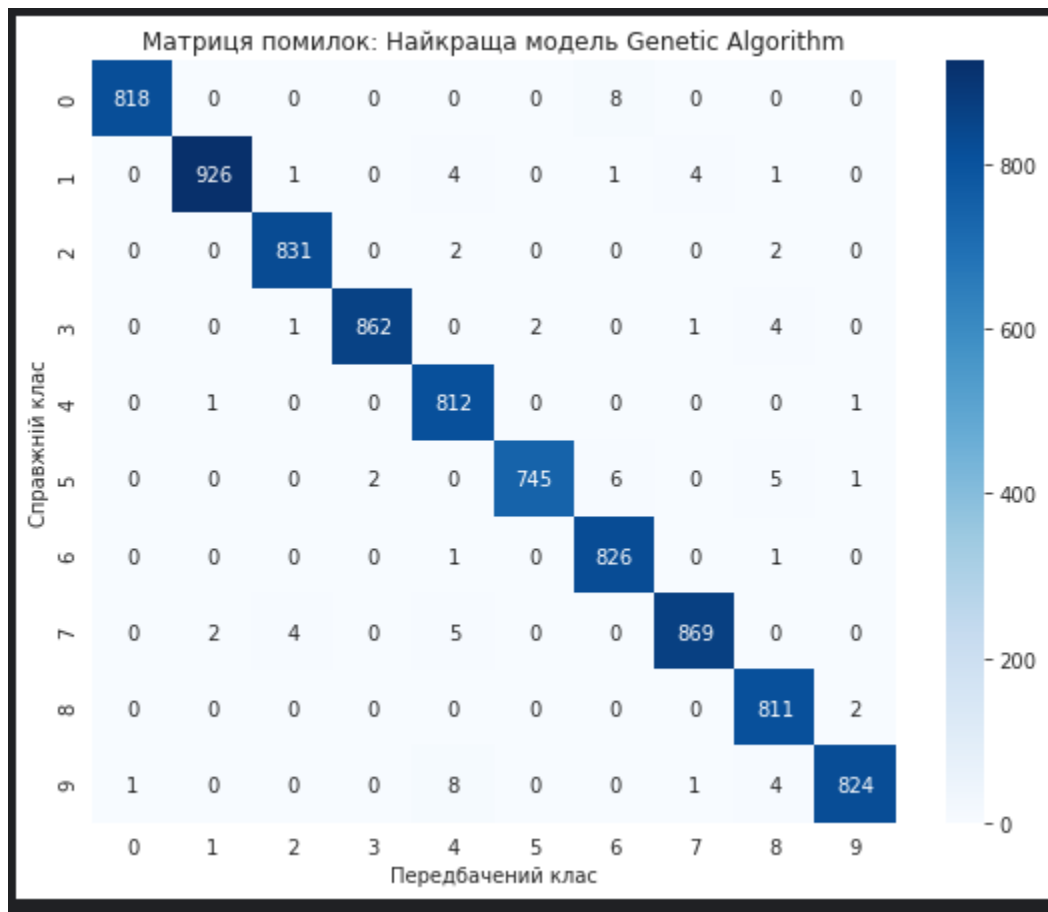


Рис. 8. Матриця помилок найкращої CNN-моделі

Рисунок 8 демонструє кількість правильних і неправильних класифікацій для кожної цифри. Більшість зображень класифіковано правильно, а залишкові помилки виникають переважно між візуально схожими цифрами. Це є очікуваним для задачі розпізнавання рукописних символів і підтверджує високу якість узагальнення побудованої моделі.

Отримані результати свідчать, що згортова нейронна мережа є ефективним інструментом для розпізнавання рукописних цифр. При цьому стратегія підбору гіперпараметрів суттєво впливає на кінцеву якість моделі та часові витрати. Random Search забезпечує швидкий і достатньо якісний результат, тоді як Genetic Algorithm дозволяє отримати кращу фінальну точність за рахунок більш складного процесу оптимізації.

Висновки

У роботі досліджено задачу розпізнавання рукописних цифр MNIST із використанням згорткової нейронної мережі. Для експерименту використано набір даних Digit Recognizer, який містить 42 000 зображень цифр у табличному форматі. Було виконано нормалізацію пікселів, перетворення вхідних векторів у тензори форми $28 \times 28 \times 1$, one-hot encoding міток класів, стратифікований поділ на навчальну та валідаційну вибірки, а також аугментацію зображень.

Побудована CNN-модель включала згорткові шари, Batch Normalization, MaxPooling, Dropout та повнозв'язні Dense-шари. Для підбору гіперпараметрів було порівняно два підходи: Random Search та Genetic Algorithm. Random Search перевіряв 8 конфігурацій із 60 можливих і витратив 357,47 секунди, досягнувши фінальної валідаційної точності 0,99143. Genetic Algorithm оцінив 12 унікальних конфігурацій за 4 покоління, витратив 578,80 секунди та досяг фінальної точності 0,99333.

Підсумкове повторне навчання найкращої конфігурації, знайденої генетичним алгоритмом, забезпечило валідаційну точність 0,99429. Звіт класифікації показав, що для всіх десяти класів значення precision, recall та F1-score перебувають приблизно в межах 0,98–1,00. Матриця помилок

підтвердила, що більшість зображень класифікуються правильно, а помилки виникають переважно між візуально схожими цифрами.

Отже, мету роботи досягнуто. Встановлено, що обидві стратегії підбору гіперпараметрів дозволяють отримати високу якість класифікації, однак мають різні переваги. Random Search є швидшим і може використовуватися як базовий підхід за обмеженого часу, тоді як Genetic Algorithm є більш обчислювально витратним, але дозволяє отримати вищу фінальну точність. Отримані результати підтверджують доцільність застосування еволюційних алгоритмів для оптимізації гіперпараметрів згорткових нейронних мереж у задачах комп'ютерного зору.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Digit Recognizer. Kaggle [Електронний ресурс]. – Режим доступу: <https://www.kaggle.com/competitions/digit-recognizer>
2. TensorFlow Documentation. Keras API [Електронний ресурс]. – Режим доступу: https://www.tensorflow.org/api_docs/python/tf/keras
3. Bergstra J., Bengio Y. Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research, 2012.
4. Genetic Algorithm. Towards Data Science / Machine Learning Mastery [Електронний ресурс]. – Режим доступу: <https://machinelearningmastery.com/simple-genetic-algorithm-from-scratch-in-python/>
5. Pandas Documentation [Електронний ресурс]. – Режим доступу: <https://pandas.pydata.org>
6. NumPy Documentation [Електронний ресурс]. – Режим доступу: <https://numpy.org>
7. Matplotlib Documentation [Електронний ресурс]. – Режим доступу: <https://matplotlib.org>
8. Seaborn Documentation [Електронний ресурс]. – Режим доступу: <https://seaborn.pydata.org>
9. Scikit-learn Documentation. Metrics and train_test_split [Електронний ресурс]. – Режим доступу: <https://scikit-learn.org>

Лавренюк Євгеній Михайлович – студент групи СА-236, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця. e-mail: lavrenyuk0629@gmail.com

Варчук Ілона Вячеславівна – к.т.н., доцент кафедри системного аналізу та інформаційних технологій, Вінницький національний технічний університет, e-mail: ivvarchuk@vntu.edu.ua

Lavreniuk Yevhemii – student of Faculty of Intellectual Information Technologies and Automation, SA-23b, Vinnytsia National Technical University, Vinnytsia, e-mail lavrenyuk0629@gmail.com

Varchuk Iona V. - Ph.D., Assistant Professor of the Department of System Analysis and Information Technologies, Vinnytsia National Technical University, Vinnytsia, e-mail: ivvarchuk@vntu.edu.ua