

УДК 004.85

В. В. ГНАТУШЕНКО, Н. Л. ДОРОШ, Т. М. ФЕНЕНКО  
Національна металургійна академія України

## РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ НАБОРУ FASHION MNIST МЕТОДАМИ ГЛИБОКОГО НАВЧАННЯ

*Огляд сучасних методів розпізнавання об'єктів-зображень показав, що алгоритми глибокого навчання успішно використовуються і дозволяють отримати високу якість. Прикладом можливо привести якість розпізнавання символів набору MNIST, яка наблизилась до 100%. Для другого набору даних, який теж популярний при тестуванні алгоритмів глибокого навчання, а саме, набору елементів одягу та аксесуарів Fashion MNIST, така висока якість розпізнавання ще не отримана.*

*У роботі представлені результати розпізнавання даних Fashion MNIST.*

*Розглянуті моделі нейронної мережі прямого поширення і згорткової мережі.*

*Виконано програмну реалізацію алгоритмів глибокого навчання, а саме, розглянуто багатосарову мережу прямого поширення (FNN) та згорткову нейронну мережу (CNN). Використано мову Python, бібліотеки TensorFlow та Keras. Бібліотека Keras дозволяє спрощувати виклики функцій з TensorFlow. Типовий процес роботи в Python Keras полягає в наступному: завантаження необхідних модулів, завантаження даних, попередня обробка, розбивка їх на навчальну, тестову і валідаційну частини; створення моделі із зазначенням архітектури, та інші. Розроблено модуль для перегляду зображень набору даних, параметрів мережі та інформації з якості розпізнавання.*

*Виконано чисельний експеримент розпізнавання елементів одягу засобами FNN. Проведено автоматичне налаштування гіперпараметрів мережі. Якість розпізнавання на тестових даних дорівнює 0,89. Налаштування гіперпараметрів покращило якість несуттєво. Невисока якість розпізнавання пояснюється також використанням дуже простої моделі нейронної мережі. Поліпшення результатів було досягнуто з використанням згорткової нейронної мережі. Крайній результат розпізнавання, який отримано у роботі, дорівнює 91.26%, але відомий найкращий результат якості розпізнавання дорівнює 94%.*

*Роботу з вдосконалення результатів розпізнавання зображень даних Fashion MNIST доцільно продовжити, а програмний засіб, який розроблено, можливо використовувати щодо розпізнавання інших даних.*

*При творчому підході розпізнавання об'єктів зображень на Python з застосуванням бібліотек Keras Tensorflow і інших є дуже перспективним напрямком для практичного застосування.*

*Ключові слова: розпізнавання зображень, Fashion MNIST, нейронна мережа, FNN, CNN, PYTHON, KERAS, TENSORFLOW, якість розпізнавання, гіперпараметри.*

В. В. ГНАТУШЕНКО, Н. Л. ДОРОШ, Т. М. ФЕНЕНКО  
Национальная металлургическая академия Украины

## РАСПОЗНАВАНИЕ ИЗОБРАЖЕНИЙ НАБОРА FASHION MNIST МЕТОДАМИ ГЛУБОКОГО ОБУЧЕНИЯ

*Обзор современных методов распознавания объектов изображений показал, что алгоритмы глубокого обучения успешно используются и позволяют получить высокое качество. Примером можно привести качество распознавания символов набора MNIST, которое приблизилось к 100%. Для другого набора данных, который тоже популярен в реализации алгоритмов глубокого обучения, а именно, набора элементов одежды и аксессуаров Fashion MNIST, такое высокое качество распознавания еще не достигнуто.*

*В работе представлены результаты распознавания данных Fashion MNIST.*

*Рассмотрены модели нейронной сети прямого распространения и сверточной сети.*

*Проведена программная реализация алгоритмов глубокого обучения, а именно, рассмотрены многослойная сеть прямого распространения (FNN) и сверточная нейронная сеть (CNN). Использованы язык Python, библиотеки TensorFlow и Keras. Библиотека Keras позволяет упрощать вызовы функций по TensorFlow. Типичный процесс работы в Python Keras заключается в следующем: загрузка необходимых модулей, загрузка данных, предварительная обработка, разбивка их на учебную, тестовую и валідаційну частини; создание модели с указанием архитектуры, и другое. Разработан модуль для просмотра изображений набора данных, параметров сети и информации по качеству распознавания.*

*Проведен численний експеримент розпізнавання елементів одягу методами FNN. Виконана автоматична настройка гіперпараметрів мережі. Якість розпізнавання на тестових даних дорівнює 0,89. Налаштування гіперпараметрів покращило якість несуттєво. Низька якість розпізнавання пояснюється також використанням дуже простої моделі нейронної мережі. Підвищення результатів було досягнуто за допомогою використання сверточної нейронної мережі. Найкращі результати розпізнавання, отримані в роботі, становлять 91,26%, тоді як відомий найкращий результат якості розпізнавання становить 94%.*

*Роботу з вдосконалення результатів розпізнавання зображень даних Fashion MNIST цілесообразно продовжити, а програмне середовище, розроблене, можливо використати для розпізнавання інших даних.*

*При творчому підході до розпізнавання об'єктів зображень на Python з використанням бібліотек Keras Tensorflow і інших, є дуже перспективним напрямком для практичного застосування.*

*Ключові слова: розпізнавання зображень, Fashion MNIST, нейронна мережа, FNN, CNN, PYTHON, KERAS, TENSORFLOW, якість розпізнавання, гіперпараметри.*

V.V. GNATUSHENKO, N.L. DOROSH, T. M. FENENKO  
National Metallurgical Academy of Ukraine

## **FASHION MNIST IMAGE RECOGNITION BY DEEP LEARNING METHODS**

*A review of modern methods for recognizing image objects has shown that deep learning algorithms are successfully used and provide high quality. An example is the quality of character recognition of the MNIST set, which is close to 100%. For another dataset, which is also popular in the implementation of deep learning algorithms, namely, the Fashion MNIST set of clothing items and accessories, such a high recognition quality has not yet been achieved.*

*The paper presents the results of data recognition Fashion MNIST.*

*Models of a feedforward neural network and convolutional network are considered.*

*The software implementation of deep learning algorithms is carried out, namely, a multilayer feedforward network (FNN) and a convolutional neural network (CNN) are considered. The Python language, the TensorFlow and Keras libraries are used. The Keras library allows you to simplify TensorFlow function calls. A typical workflow in Python Keras is as follows: loading the necessary modules, loading data, preprocessing, breaking them into training, test and validation parts; creating a model with an indication of the architecture, and more.*

*A numerical experiment was carried out to recognize clothing items by means of FNN. Automatic tuning of network hyperparameters has been performed. The recognition quality on test data is 0.89. The hyperparameter tuning did not significantly improve the quality. The low quality of recognition is also explained by the use of a very simple neural network model. Improved results have been achieved using convolutional neural networks. The best recognition results obtained in the work is 91.26%, but the known best recognition quality result is 94%.*

*It is advisable to continue the work on improving the results of recognition of images of Fashion MNIST data, and the software that has been developed can be used to recognize other data.*

*With a creative approach to image object recognition in Python using the Keras Tensorflow libraries and others, it is a very promising direction for practical application.*

*Keywords: image recognition, Fashion MNIST, neural network, FNN, CNN, PYTHON, KERAS, TENSORFLOW, recognition quality, hyperparameters.*

### **Постановка проблеми**

Розпізнавання об'єктів за технологією глибокого навчання є популярним і успішним методом вирішення задач інтелектуального аналізу даних. Вивчення та дослідження алгоритмів глибокого навчання доцільно проводити на широко відомих масивах даних, які набули вже статус класичних. Одним з них є набір даних MNIST [1], який існує з 1999 року і включає 70000 зображень рукописних цифр. Найкращий результат розпізнавання 99,7% отриманий з використанням згорткової мережі, класичні ж алгоритми машинного навчання дозволяють розпізнавати до 97% символів MNIST [2]. Завдяки таким добрим результатам, цей набір все ще не втрачає своєї актуальності і використовується в задачах машинного навчання для тестування алгоритмів та методів

розпізнавання [1,3]. Однак, є й така пропозиція: «MNIST is too easy» [2], то бажано використовувати інший набір даних - Fashion MNIST [4].

У роботі представлені результати розпізнавання зображень елементів одягу набору даних Fashion MNIST [4] з використанням алгоритмів глибокого навчання [5].

### **Аналіз останніх досліджень і публікацій**

Застосування глибокого навчання прискорило прогрес при вирішенні багатьох завдань машинного навчання, в тому числі і в розпізнаванні зображень. У 2009-2010 роках за допомогою згорткових мереж покращилися результати розпізнавання символів.

Більшість завдань машинного навчання зводиться до вирішення задачі оптимізації [6]. У задачі класифікації, яка вирішується у роботі, цільова функція відповідає вибору з кількох класів, тобто має дискретні значення. Кожному класу повинна відповідати область в просторі параметрів. Функція помилки визначається в залежності від числа вірних відповідей. Така метрика (точність класифікації), є кусочно-постійною, а не безперервною функцією, тому оптимізація такої функції не є тривіальною задачею.

Глибоке навчання - підмножина методів машинного навчання, в яких застосовуються штучні нейронні мережі. Слово «глибокий» має на увазі наявність великого числа шарів в штучній нейронній мережі, але його інтерпретація з часом змінювалася. Якщо ще декілька років тому вважалося, що 10 шарів досить, щоб називати мережу глибокою, то тепер глибокою зазвичай називається мережа, яка містить сотні шарів [5, 6].

В даний час застосування глибокого навчання не тільки перспективно, але і практично можливо. З'явилися дуже зручні бібліотеки, які дозволяють швидко побудувати модель нейронної мережі, сформувати для неї граф обчислень, автоматично підрахувати градієнти і виконати процес навчання. З кожним роком ці бібліотеки стають все зручнішими, виходять нові версії існуючих бібліотек, з'являються абсолютно нові програмні продукти в цій області науки. Слід зазначити, що існують і бібліотеки загального призначення, які можуть створити будь-який граф обчислень, і спеціалізовані надбудови, які реалізують різні компоненти нейронних мереж: звичайні шари, згорткові, рекурентні, сучасні алгоритми оптимізації. Такими бібліотеками є TensorFlow і Keras [6-9].

### **Мета дослідження**

Метою роботи є розпізнавання об'єктів набору даних Fashion MNIST методами глибокого навчання з можливістю налаштування гіперпараметрів з використанням програмного засобу, який розроблено.

### **Викладення основного матеріалу дослідження**

Набір даних Fashion MNIST містить 70000 зображень одягу і аксесуарів [4]. З цих 70000 взято 60000 зображень для навчання нейронної мережі, а інші 10000 елементів – для перевірки якості розпізнавання. Набір складається з зображень розміром 28x28 пікселів, кожен піксель представляє собою відтінок сірого. Значення пікселів - це шкала сірого в діапазоні від 0 до 255. У більшості випадків рекомендується виконувати масштабування вхідних значень, наприклад нормалізувати значення пікселів в діапазоні [0,1]. Номери класів - це масив цілих чисел від 0 до 9, вони відповідають класам об'єктів, які зображені на картинках. Кожному зображенню відповідає єдиний клас: 0 T-shirt/top (футболка/топ), 1 Trouser (шорти), 2 Pullover (пуловер), 3 Dress (сукня), 4 Coat (пальто), 5 Sandal (сандали), 6 Shirt (сорочка), 7 Sneaker (кросівки), 8 Bag (сумка), 9 Ankle boot (ботильони).

Для реалізації розпізнавання одягу розроблено модулі програмного засобу з використанням мови Python [10], бібліотек TensorFlow та Keras [6]. Бібліотека Keras дозволяє спрощувати виклики функцій з TensorFlow. Типовий процес роботи в Python Keras полягає в наступному: 1) завантаження необхідних модулів; 2) завантаження даних, попередня обробка, розбивка їх на навчальну, тестову і валідаційну частини; 3) створення моделі (model) із зазначенням архітектури, виду шарів, функції активації, схеми ініціалізації і регуляризації ваг, зміщення та ін.. Розроблено модуль для перегляду зображень набору даних, параметрів мережі та інформації з якості розпізнавання.

В роботі було розглянуто багатшарову мережу прямого поширення (FNN) з використанням методів автоматичного налаштування гіперпараметрів та архітектуру згорткової нейронної мережі (CNN).

Досліджуємо просту послідовну модель, щоб використовувати її як основу для порівняння з більш складними моделями нейронних мереж. Архітектура мережі містить вхідний шар  $28 \times 28 = 784$ , один прихований шар і вихідний шар. Використано алгоритм оптимізації "adam", функцію втрат "sparse\_categorical\_crossentropy", метрику ['accuracy']. Кількість епох дорівнює 30. В результаті обчислень: точність на навчальних даних дорівнює 0,95, точність на тестових даних дорівнює 0,89.

На рисунку 1 наведено приклад перегляду даних та результатів розпізнавання відповідно простої моделі мережі. Вісь ординат відповідає ймовірності розпізнавання для даного класу. Червоний колір означає, що кількість зображень, які неправильно розпізнані, більше, ніж правильно розпізнаних для даного класу; синій – навпаки.

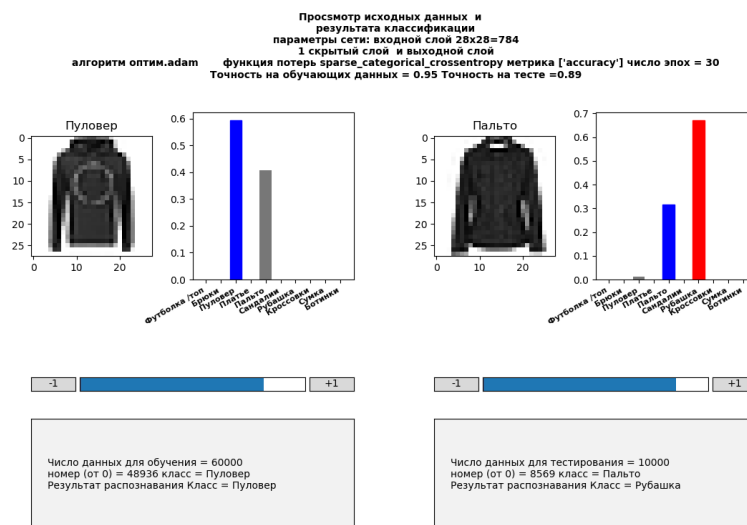


Рис. 1. Приклад перегляду даних та результатів розпізнавання

Проведено дослідження цієї простої моделі при зміні параметрів. У роботі для налаштування гіперпараметрів використано інструмент Keras Tuner. Для цього створено модель мережі з параметрами налаштування для Keras Tuner і його виклик.

Для прихованого шару з ім'ям 'Hidden\_layer' проведено налаштування таких гіперпараметрів: число нейронів в шарі, функції активації (число нейронів в шарі від 50 до 1000 з кроком зміни 50; функції активації з наступного переліку: 'relu', 'tanh', 'sigmoid'). Використовували алгоритми оптимізації ['adam', 'SGD'].

З точки зору найкращої точності визначено 3 варіанти. У всіх трьох наданих варіантах розрахунків метрика якості Score дорівнює приблизно 0,9. Варіанти містять

кількість нейронів в шарі 900, 700, 500; функцію активації 'relu', 'tanh', 'tanh' відповідно варіантам 1, 2,3; алгоритм оптимізації -'adam'.

Результати машинного навчання з використанням нейронної мережі з прямим розповсюдженням сигналу, архітектура якої є найпростішою (FNN-модель за типом «Sequential», з вхідним шаром типу «Flatten», одним прихованим шаром «Dense» і вихідним шаром «Dense») та налаштуванням гіперпараметрів зведені у таблицю 1.

Таблиця 1

Результати розпізнавання для FNN-моделі

|   | Модель                      | Параметри шару 'Hidden_layer' |   | Оптимізатор ['adam', 'SGD'] | Точність на валідаційних даних | Точність на тестових даних | Загальне число нейронів | Функція втрат на тестових даних |
|---|-----------------------------|-------------------------------|---|-----------------------------|--------------------------------|----------------------------|-------------------------|---------------------------------|
|   |                             | Число нейронів                | Функція активації ['relu', 'tanh', 'sigmoid'] |                             |                                |                            |                         |                                 |
| 1 | Початкова модель            | 784                           | 'relu'  | 'adam'                      | 0.8976                         | 0.8912                     | 623290                  | 0.3791                          |
| 2 | Модель_1 TUNER RandomSearch | 900                           | 'relu'  | 'adam'                      | 0.9002                         | 0.8935                     | 715510                  | 0.3417                          |
| 3 | Модель_2 TUNER RandomSearch | 700                           | 'tanh'  | 'adam'                      | 0.8988                         | 0.8925                     | 556510                  | 0.3294                          |
| 4 | Модель_3 TUNER RandomSearch | 500                           | 'tanh'  | 'adam'                      | 0.8981                         | 0.8884                     | 397510                  | 0.3209                          |
| 5 | Модель_1 TUNER Hyperband    | 800                           | 'relu'  | 'adam'                      | 0.8898                         | 0.8771                     | 636010                  | 0.3341                          |
| 6 | Модель_2 TUNER Hyperband    | 550                           | 'relu'  | 'adam'                      | 0.8893                         | 0.8830                     | 437260                  | 0.3329                          |
| 7 | Модель_3 TUNER Hyperband    | 350                           | 'relu'  | 'adam'                      | 0.8888                         | 0.8795                     | 278260                  | 0.3404                          |

Налаштування гіперпараметрів також не привело до суттєвого покращення результатів.

Розглянемо згорткову модель нейронної мережі з архітектурою:

- перший прихований шар - це шар згортки, Convolution2D, що має 32 карти функцій, розмір яких дорівнює  $3 \times 3$ , і функції активації relu;
- потім визначено шар пулінг, maxPooling2D, з розміром пулу  $2 \times 2$ , який дає максимальні значення;
- наступний рівень - це рівень регуляризації Dropout, він налаштований на випадкове виключення 20% нейронів в шарі, щоб зменшити перенавчання;
- далі - шар, який перетворює дані двовимірної матриці в вектор Flatten, він дозволяє обробляти вихідні дані стандартними повнозв'язними шарами;
- потім - повнозв'язний шар з 64 нейронами і функцією активації relu;
- вихідний шар має 10 нейронів для 10 класів і функцію активації softmax для виведення ймовірнісних результатів розпізнавання для кожного класу.

На рисунку 2 представлені параметри моделі згорткової мережі та її граф.

Результати розпізнавання дали точність на тестових даних 90,78%

Невірні результати розпізнавання наочно показані у матриці неточності моделі, яка наведена на рисунку 3.

Найбільше невірних результатів отримано при розпізнаванні таких об'єктів, як «Пальто» і «Рубашка».

З метою поліпшення результатів розпізнавання були проведені розрахунки згортковою нейромережею з фільтрами (5x5) і (7x7).

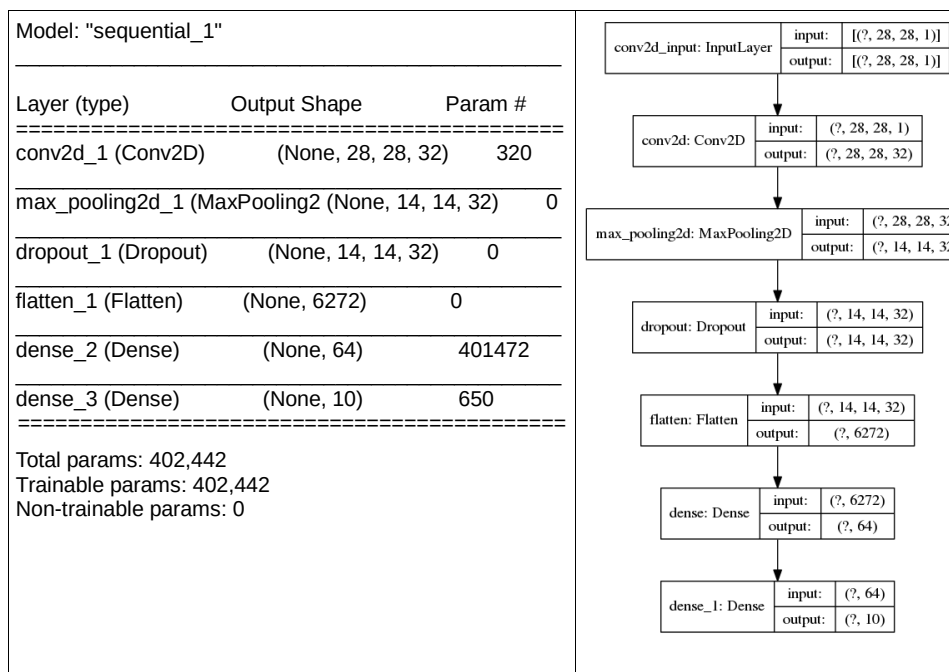


Рис. 2. Параметри моделі і граф CNN

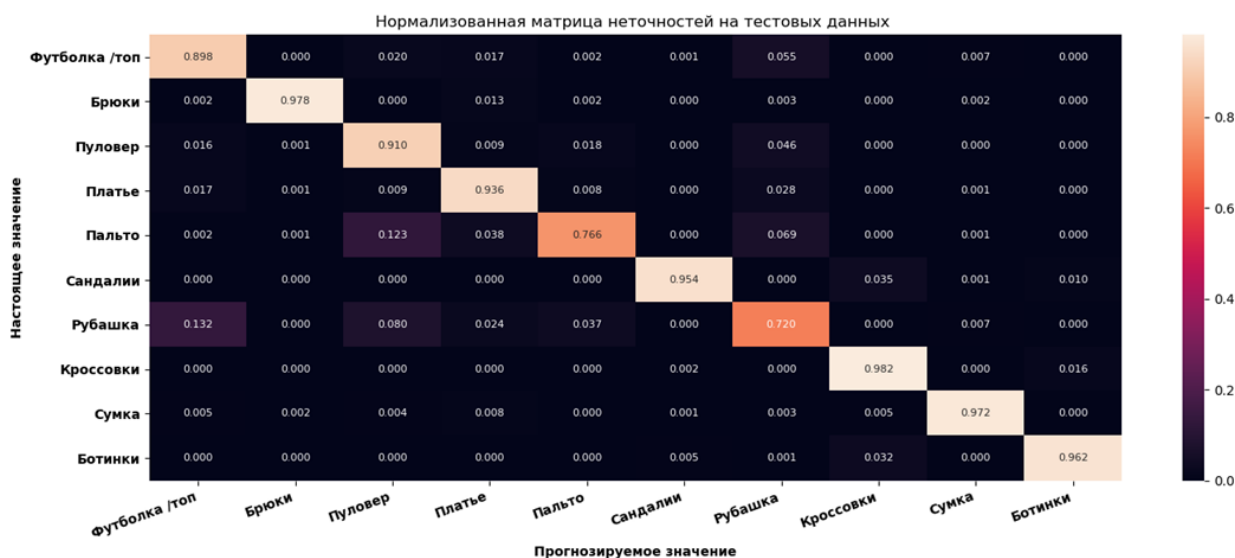


Рис. 3. Матрица неточності моделі CNN

У таблиці 2 показані результати розпізнавання з використанням згорткової нейронної мережі. Кращі результати відповідають моделі з розміром фільтру (7x7), точність розпізнавання дорівнює 91.26%.

Таблиця 2

Результати розпізнавання для CNN-моделі з різними фільтрами

| Розмір ядра фільтра | Точність розпізнавання на тестових даних | Значення функції втрат при розпізнаванні тестових даних |
|---------------------|--|---|
| 3 x 3               | 90.78%                                   | 0.2796  |
| 5 x 5               | 90.89%                                   | 0.2799  |
| 7 x 7               | 91.26%                                   | 0.2630  |

Застосування методів автоматичного налаштування гіперпараметрів для такої згорткової мережі є трудомісткою справою, доцільніше буде використовувати другу архітектуру мережі, наприклад [3], де отримана точність трохи більша і дорівнює 93,56%.

Архітектура мережі для конкретного завдання шукається в ході експериментів, напрям яких визначається помилкою на конкретному наборі даних.

### Висновки

Проведено розпізнавання зображень Fashion MNIST з використанням програмного засобу, розробленого мовою Python з функціями бібліотек Keras і Tensorflow. Розглянуті моделі нейронної мережі прямого поширення і згорткової мережі.

Розпізнавання зображень Fashion MNIST за моделлю FNN показало якість розпізнавання на навчальних даних 0,95, на тестових - 0,89. Проведено дослідження моделі FNN, яку отримано в результаті налаштування гіперпараметрів, але якість розпізнавання поліпшилась несуттєво. У даній роботі найкращий результат отримано для згорткової мережі, точність розпізнавання дорівнює 91.26%.

Роботу з вдосконалення результатів розпізнавання зображень даних Fashion MNIST доцільно продовжити, а розроблений програмний засіб можливо використовувати щодо розпізнавання інших наборів даних.

При творчому підході розпізнавання об'єктів зображень на Python з застосуванням бібліотек Keras Tensorflow і інших є дуже перспективним напрямком для практичного застосування.

### Список використаної літератури

1. Digit Recognizer [Електронний ресурс] – Режим доступу. — URL: <https://www.kaggle.com/c/digit-recognizer> (дата звернення 20.05.2021).
2. To Serious Machine Learning Researchers [Електронний ресурс] – Режим доступу. — URL: <https://github.com/zalandoresearch/fashion-mnist> (дата звернення 23.04.2021).  
[https://www.researchgate.net/publication/343173734\\_CNN\\_Model\\_for\\_Image\\_Classification\\_on\\_MNIST\\_and\\_Fashion-MNIST\\_Dataset](https://www.researchgate.net/publication/343173734_CNN_Model_for_Image_Classification_on_MNIST_and_Fashion-MNIST_Dataset)
3. Shivam S. Kadam, Amol C. Adamuthe, Ashwini B. Patil. CNN Model for Image Classification on MNIST and Fashion-MNIST Dataset. *Journal of Scientific Research* 64(02):374-384. DOI:10.37398/JSR.2020.640251– Режим доступу. — URL: <https://www.researchgate.net/publication/343173734>
4. Digit Recognizer [Електронний ресурс] – Режим доступу. — URL: <https://www.kaggle.com/zalando-research/fashionmnist>: (дата звернення 20.05.2021).
5. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение/пер. с англ. Слинкина А.А. 2-е изд., испр. М.: ДМК Пресс, 2018. 652 с.
6. Николенко С., Кадурын А., Архангельская Е. Глубокое обучение. СПб.: Питер, 2018. 480 с.
7. Антонио Джулли, Суджит Пал. Библиотека Keras – инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow/Пер. с англ.Слинкин А. А. М.: ДМК Пресс, 2018. 294с.
8. Жерон, Орельен. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем./Пер. с англ. СПб.: ООО "Альфа-книга, 2018. 688 с.
9. Плас Дж. Вандер. Python для сложных задач: наука о данных и машинное обучение. СПб.: Питер, 2018. 576 с.

10. Лутц М. Программирование на Python, том I, 4-е издание. Пер. с англ. СПб.: Символ-Плюс, 2011. 992 с.

#### References

1. Digit Recognizer [Elektronnyi resurs] – Rezhym dostupu. — URL: <https://www.kaggle.com/c/digit-recognizer>
2. To Serious Machine Learning Researchers [Elektronnyi resurs] – Rezhym dostupu. — URL: <https://github.com/zalandoresearch/fashion-mnist>.  
[https://www.researchgate.net/publication/343173734\\_CNN\\_Model\\_for\\_Image\\_Classification\\_on\\_MNIST\\_and\\_Fashion-MNIST\\_Dataset](https://www.researchgate.net/publication/343173734_CNN_Model_for_Image_Classification_on_MNIST_and_Fashion-MNIST_Dataset)
3. Shivam, S. Kadam, Amol, C. Adamuthe & Ashwini, B. Patil. CNN Model for Image Classification on MNIST and Fashion-MNIST Dataset. *Journal of Scientific Research* 64(02):374-384. DOI:[10.37398/JSR.2020.640251](https://doi.org/10.37398/JSR.2020.640251)– Rezhym dostupu. — URL: <https://www.researchgate.net/publication/343173734>
4. Digit Recognizer [Elektronnyi resurs] – Rezhym dostupu. — URL: <https://www.kaggle.com/zalando-research/fashionmnist>
5. Gudfellou, Ya., Bendzhio, I. & Kurvill, A. (2018). Glubokoe obuchenie/per. s ang. Slinkina A. A. 2-e izd., ispr. M.: DMK Press.
6. Nikolenko, S., Kadurin, A. & Arhangelskaya, E. (2018). Glubokoe obuchenie. SPb.: Piter.
7. Antonio, Dzhulli & Sudzhit, Pal. (2018). Biblioteka Keras – instrument glubokogo obucheniya. Realizatsiya neyronnykh setey s pomoschyu bibliotek Theano i TensorFlow/Per. s angl.Slinkin A. A. M.: DMK Press.
8. Zheron, Orelen. (2018). Prikladnoe mashinnoe obuchenie s pomoschyu Scikit-Learn i TensorFlow: kontseptsii, instrumentyi i tehniki dlya sozdaniya intellektualnykh sistem./Per. s angl. SPb.: OOO "Alfa-kniga.
9. Plas, Dzh. Vander. (2018). Python dlya slozhnykh zadach: nauka o dannykh i mashinnoe obuchenie. SPb.: Piter.
10. Lutts, M. (2011). Programirovanie na Python, tom I, 4-e izdanie. / Per. s angl. SPb.: Simvol-Plyus.

Гнатушенко Вікторія Володимирівна – д.т.н., професор кафедри інформаційних технологій та систем Національної металургійної академії України. E-mail: [vvitagnat@gmail.com](mailto:vvitagnat@gmail.com), <https://orcid.org/0000-0001-5304-4144>.

Дорош Наталія Леонідівна – к.т.н. доцент кафедри інформаційних технологій та систем Національної металургійної академії України, e-mail: [natalidorosh53@gmail.com](mailto:natalidorosh53@gmail.com) <https://orcid.org/0000-0003-4184-3648>

Фененко Тетяна Михайлівна – старший викладач кафедри інформаційних технологій та систем Національної металургійної академії України, e-mail: [fenenkot@gmail.com](mailto:fenenkot@gmail.com) <https://orcid.org/0000-0002-7631-3148>.