

© Балалаєва О. Ю.<sup>1</sup>, Чичкар'юв Є.А.<sup>2</sup>, Зінченко О.В.<sup>3</sup>, Сергієнко А.В.<sup>4</sup>,  
Ковальов О.О.<sup>5</sup>

### ДОСЛІДЖЕННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ МЕТОДОЛОГІЇ РОЗПІЗНАВАННЯ РУКОПИСНИХ СИМВОЛІВ З ВИКОРИСТАННЯМ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

У роботі проведено аналіз можливостей використання нейронних мереж глибокого навчання для реалізації інформаційних систем обробки текстів, обґрунтовано перспективність даної технології та можливості її удосконалення. Розглянуто можливість використання бази рукописних цифр MNIST, а також бази рукописних букв і цифр EMNIST, а саме набору EMNIST Letters. Проведено моделювання об'єкта дослідження в нотаціях IDEF0 та IDEF3 для варіанту «AS-IS». Виявлено, що найбільше питань викликають «Побудова набору даних для навчання моделі» та «Вибір архітектури згорткової нейронної мережі». Запропоновано заходи з реінжинірингу, а саме показано доцільність використання крім відомих наборів EMNIST та MNIST додатково створеного в рамках роботи набору рукописних і курсивних шрифтів, що мають українські гліфи. Для цього обрані сучасні IT-інструменти, такі як бібліотека Pillow, Image Data Generator та пакет Scikit-Learn використано для виділення навчальних та тестових вибірок. Також крім початково запропонованих найпростіших архітектур CNN типу Lenet запропоновано використання більш складних архітектур типу AlexNet та VGG-16. З урахуванням запропонованих заходів з реінжинірингу побудовано діаграми IDEF0 та IDEF3 для варіанту «TO-BE». У роботі докладно проаналізовано результати розпізнавання рукописних українських літер та арабських цифр з використанням різних архітектур CNN з використанням для навчання синтетичного набору даних. Дослідження, представлені в роботі, проведено з використанням програмного додатку, розробленого на мові програмування Python з використанням пакету Scikit Learn, що надає користувачеві можливість розпізнати рукописний текст за допомогою багатозарового перцептрона, що проведено саме з використання розробленої програми. Обґрунтовано, що на протизагу простим архітектурам типу Lenet доцільніше використовувати більш складні варіанти, а саме архітектуру типу VGG-16. Проведено експериментальні дослідження впливу кількості параметрів CNN різних архітектур на точність розпізнавання та час навчання нейронної мережі. Також проаналізовано результати розпізнавання символів при розпізнаванні зображень, що не належать до навчальної або тестової вибірки.

**Ключові слова:** розпізнавання рукописного тексту, розпізнавання символів, OCR, згорткові нейронні мережі, CNN, глибоке навчання, обробка зображень.

<sup>1</sup> канд. техн. наук, доцент, ДВНЗ «Приазовський державний технічний університет», м. Дніпро/Маріуполь, ORCID: 0000-0003-1461-4399, [balalaevaev@gmail.com](mailto:balalaevaev@gmail.com)

<sup>2</sup> д-р техн. наук, професор, Державний університет інформаційно-комунікаційних технологій, м. Київ, ORCID: 0000-0002-4362-5129, [influence@meta.ua](mailto:influence@meta.ua)

<sup>3</sup> д-р техн. наук, професор, Державний університет інформаційно-комунікаційних технологій, м. Київ, ORCID: 0000-0002-3973-7814, [zinchenkoov@gmail.com](mailto:zinchenkoov@gmail.com)

<sup>4</sup> канд. техн. наук, ДВНЗ «Приазовський державний технічний університет», м. Дніпро/Маріуполь, ORCID: 0000-0003-1328-2572, [sergienko\\_a\\_v@pstu.edu](mailto:sergienko_a_v@pstu.edu)

<sup>5</sup> магістрант, ДВНЗ «Приазовський державний технічний університет», м. Дніпро/Маріуполь, [ronin732@gmail.com](mailto:ronin732@gmail.com)

*O. Balalaieva, Ye. Chychkarov, O. Zinchenko, A. Serhiienko, O. Kovalov. Research and evaluation of the efficiency of handwritten character recognition methods using convolutional neural networks. The paper analyzes the possibilities of using deep learning neural networks for the implementation of text processing information systems, substantiates the prospects of this technology and the possibilities of its improvement. The possibility of using the MNIST database of handwritten digits on, as well as the EMNIST database of handwritten letters and numbers, namely the EMNIST Letters set, was considered. The research object was modelled in IDEF0 and IDEF3 notations for the «AS-IS» option. It was found that «Building a dataset for model training» and «Choosing a convolutional neural network architecture» are the most frequently asked questions. Reengineering measures are proposed, namely, the expediency of using, in addition to the well-known EMNIST and MNIST sets, a set of handwritten and italic fonts with Ukrainian glyphs, additionally created as part of the work, is shown. For this purpose, modern IT tools such as the Pillow library, Image Data Generator and the Scikit-Learn package were used to select training and test samples. Also, in addition to the initially proposed simplest CNN architectures of the Lenet type, the use of more complex architectures of the AlexNet and VGG-16 types is proposed. Taking into account the proposed reengineering measures, IDEF0 and IDEF3 diagrams were constructed for the «TO-BE» option. The paper analyzes in detail the results of recognizing handwritten Ukrainian letters and Arabic numerals using 6 different CNN architectures using a synthetic data set for training. The research presented in the work was carried out using a software application developed in the Python programming language using the Scikit Learn package, which provides the user with the ability to recognize handwritten text using a multilayer perceptron. carried out precisely using the developed program. It is justified that, in contrast to simple Lenet-type architectures, it is more appropriate to use more complex options, namely the VGG-16 type architecture. Experimental studies of the influence of the number of CNN parameters of different architectures on the recognition accuracy and training time of the neural network have been carried out. Also analyzed are the results of character recognition when recognizing images that do not belong to the training or test sample.*

**Key words:** recognition of handwritten text, recognition of characters, OCR, convolutional neural networks, CNN, digital recognition, deep learning, image processing.

**Постановка проблеми.** Розпізнавання символів (оптичне розпізнавання, OCR) – технологія, яка на сьогоднішній день широко використовується. В основі цієї технології лежить процес класифікації зображень символів, які виділені на вихідному цифровому зображенні, за відповідними зразками [1].

Інформаційні технології, які використовують оптичне розпізнавання, застосовуються для вирішення широкого кола різноманітних практичних завдань: ідентифікація реєстраційних номерів транспортних засобів за зображеннями номерних знаків, що допомагає контролювати рух [2], перетворення друкованих академічних записів у текст для зберігання в електронній базі даних, декодування стародавніх написів та текстів, автоматичне введення даних шляхом оптичного сканування карт, банківських чеків тощо. Використання системи OCR дозволяє суттєво знизити ймовірність виникнення помилок та зменшити витрати часу на рутинну роботу. Найбільш складним завданням досі залишається розпізнавання рукописних написів, що потребує додаткового дослідження та удосконалення наявних методологій та технологій.

Найчастіше робота сучасних OCR ґрунтується на нейронних мережах глибокого навчання [3, 4]. Для обробки зображень широко застосовуються згорткові нейронні мережі (convolutional neural network, CNN). Це один із найпопулярніших типів глибоких нейронних мереж, за його допомогою можна ефективно розпізнавати символи, присутні на зображенні [5].

**Метою даної роботи** є дослідження можливостей розпізнавання українських рукописних літер та арабських цифр з використанням згорткових нейронних мереж.

**Аналіз останніх досліджень та публікацій.** Спочатку більшість досліджень була зосереджена на розпізнаванні літер латинського алфавіту, але в останні роки збільшилася частка публікацій, в яких розглядаються методи роботи з кириличними літерами, а також арабськими, казахськими, китайськими та індійськими символами тощо [6-11]. Найчастіше для аналізу методів

та технологій розпізнавання латинських рукописних літер використовує стандартний набір EMNIST [12], а автори робіт пропонують застосування різних архітектур CNN.

Однією з перших успішних спроб використання глибокого навчання для розпізнавання символів стало створення архітектури LeNet-5 [13], що наразі залишається актуальним для комп'ютерів з низькою потужністю. Наприклад, ConvNet [14] має всього 60000 навчальних параметрів, що є значно меншим, ніж у мережі AlexNet (60 млн навчальних параметрів і 650000 нейронів) [15] та мережі GoogleNet (6,8 млн навчальних параметрів) [16].

Для досягнення більш високих показників точності розпізнавання доцільно використовувати більш складні варіанти архітектур нейронних мереж. Наприклад, для БД EMNIST Letters точність більше 95% досягалася при поєднанні марківських моделей випадкових полів та CNN, а для MNIST – при використанні капсульних шарів після згортки [17, 18].

Одним із способів покращення точності розпізнавання зображень є використання моделей складнішої архітектури. Наприклад, у [19] для розпізнавання ізольованих символів запропонована згорткова нейронна мережа, що містить 14 згорткових шарів для представлення ознак символів, два шари MaxPooling для зменшення розміру ознаки або для виділення сильних ознак, один шар softmax і один шар класифікації.

Для забезпечення необхідної швидкості навчання при збільшенні числа параметрів використовують попередньо треновані моделі, хоча такий підхід має свої недоліки, тому автори [20] пропонують навчання з випадковою ініціалізацією, що буде еквівалентним за загальною тривалістю навчання. У роботі [21], розглядаючи проблему розпізнавання арабських символів, демонструються більш високі результати для моделей, створених з нуля, у порівнянні з навченими моделями. При цьому автори роблять висновок про невисоку точність для менш складних архітектур CNN при одночасній перевазі у швидкості класифікації та навчання.

Результати застосування більш складних попередньо навчених архітектур CNN розглянуто у роботі [22] на прикладі класифікації 231 рукописних символів Bangla з використанням БД CMATERdb. Розмір зображень було зменшено до 28×28 пікселів. Ці зображення були використані як вхідні дані для архітектури CNN. Швидкість навчання було встановлено рівної 0,001. У якості функції помилок було використано категоріальну крос-ентропію. Після 50 епох InceptionResNetV2 досягла найкращої точності (96,99%). Високу точність розпізнавання продемонстрували також DenseNet121 (96,55%) та InceptionNetV3 (96,20%). Авторами також показано, що найкраща точність (97,69%) забезпечується комбінуванням навчених архітектур DenseNet121, InceptionResNetV2 та InceptionNetV3. Однак практичне використання вимагає великих обчислювальних потужностей та пам'яті і тому не є оптимальним. Натомість найбільш ефективною моделлю обрано InceptionResNetV2.

Автори [23] провели тестування двох варіантів CNN з різними архітектурями, варіюючи глибину, ширину та кількість параметрів мережі, для розпізнавання символів. Перша модель складалася з трьох згорткових шарів та одного повнозв'язного шару. Друга модель походила від сімейства LeNet і складалася з двох згорткових шарів, за якими йдуть два повнозв'язкові шари. Найкращу точність результат (понад 98%) було отримано при використанні моделі, у складі якої більше згорткових шарів. Аналогічний результат отримано й у роботі [24]. Авторами досліджували три варіанти архітектури мереж CNN: Lenet-5, модифікований варіант Lenet та Alexnet CNN. Останній варіант архітектури нейронної мережі дозволив досягти точності розпізнавання символів деванагарі на рівні 99%.

Авторами [25] проведено експериментальні дослідження ефективності роботи CNN, як ResNet та VGG-16 (з використанням відсівів та аугментації), що показали високі результати точності – 98,57% та 97,14% відповідно.

У роботі [26] також відзначено більш високу досягнуту точність розпізнавання при використанні глибшої архітектури нейронної мережі CNN. З іншого боку, підвищення точності розпізнавання досягається за допомогою використання аугментації вхідних даних. У роботі [27] отримано аналогічні результати на наборі EMNIST, при цьому показано, що GoogleNet демонструє найбільш високу точність, однак використовує у 2,5-3 рази більше ресурсу часу в порівнянні з ResNet18.

Однак слід враховувати, що архітектури нейронних мереж, які використовують попереднє навчання, було створено для класифікації кольорових зображень різних розмірів. Це обумовлює

потребу у зміні розміру зображення та його триразовому копіюванні для отримання триканального зображення для запуску моделі DCNN [27].

У разі використання моделей без завантаження ваг попередньо навченої моделі вхідні дані можуть містити одноканальні зображення, тому архітектура CNN модифікується. При порівнянні варіантів розпізнавання кольорових і монохромних зображень [27] зазначено, що варіанти з розміром вхідного зображення 40×40 пікселів (для набору даних EMNIST зі зміненим розміром) в монохромних версіях з аугментацією обертання і зсуву мають найвищі результати в обох моделях (ResNet18 та GoogleNet).

Для розпізнавання кирилических символів подібні дослідження досить нечисленні. Є досвід використання архітектури MobileNet, яка включала 30 шарів [28] для розпізнавання символів казахської та російської мов. Деякі результати розпізнавання кирилических символів представлені також у [29].

Відомо досить багато досліджень технологій розпізнавання рукописного введення, які ґрунтуються на використанні набору даних EMNIST [17].

Для розпізнавання кирилиці відомий досвід використання різних класифікаторів і нейромережових технологій, однак порівняльні дослідження технологій розпізнавання для них носять фрагментарний характер. Крім того, немає наборів даних, подібних до EMNIST, для українського алфавіту.

Щодо набору даних для розпізнавання українських букв, то відомі поодинокі роботи у цьому напрямі [30, 31]. На думку [30] при створенні набору даних для навчання моделі необхідно розпізнавати великі і малі літери, а також враховувати можливість різного написання однієї і тієї ж літери. Авторами [30] було виділено понад 70 класів, що утворюють повний набір символів української мови (наприклад, враховувалося різне написання малої літери «а»).

**Виклад основного матеріалу.** Для дослідження методології розпізнавання рукописних символів побудовано відповідну контекстну діаграму IDEF0, спочатку для варіанту «AS-IS». Декомпозицію даної діаграми наведено на рис. 1.

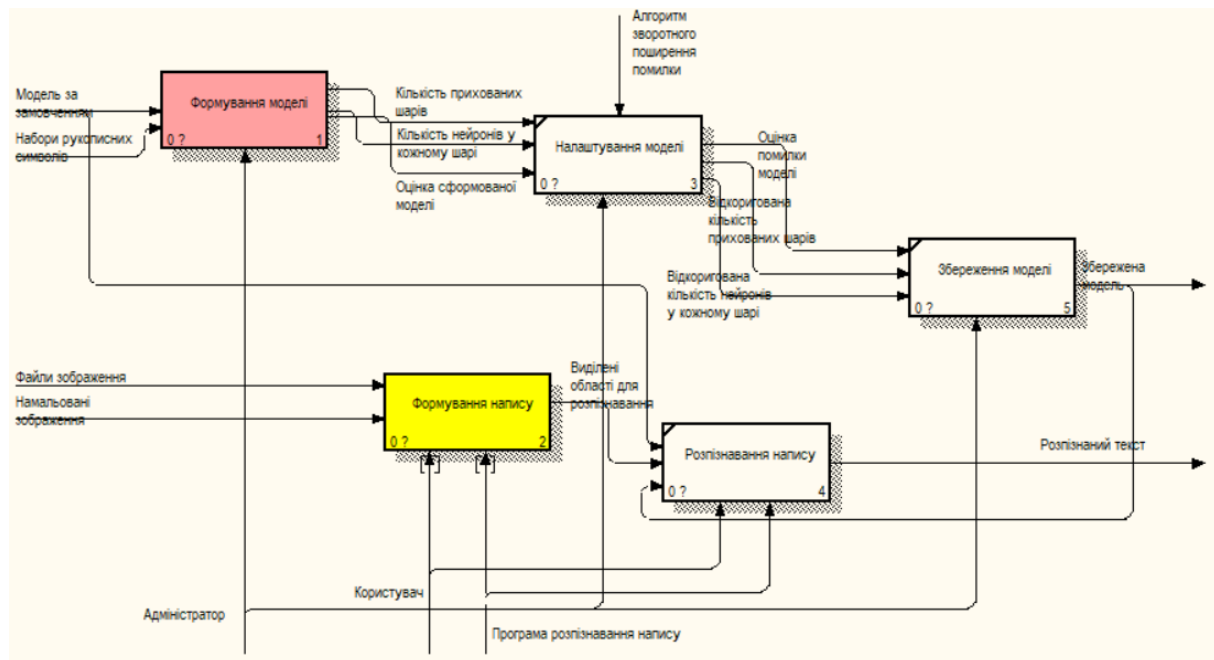


Рис. 1 – Діаграма першого рівня декомпозиції процесу «Методологія розпізнавання рукописних символів» (AS-IS)

У процесі розпізнавання рукописних символів можна виділити наступні підпроцеси:  
– формування моделі – результатом підпроцесу є визначена кількість прихованих шарів і кількість нейронів у кожному шарі, виконавцем підпроцесу є адміністратор програми;

- налаштування моделі – виконується налагодження моделі з використанням алгоритму зворотного поширення помилки, за результатами або приймається модель з параметрами за замовчуванням, або коригується кількість прихованих шарів і кількість нейронів у шарі, виконавцем підпроцесу є адміністратор програми;
- збереження моделі – у графічному інтерфейсі користувач задає ім'я файлу, під яким буде збережено модель (її можна буде потім обирати);
- формування напису;
- розпізнавання напису – користувач запускає програму розпізнавання для напису у вікні напису з використанням або моделі за замовченням, або попередньо збереженої та завантаженої моделі.

Найбільший інтерес викликає етап «Формування моделі» (рис. 2). Він включає два важливих підпроцеси: «Побудова набору даних для навчання моделі» та «Вибір архітектури згорткової нейронної мережі».

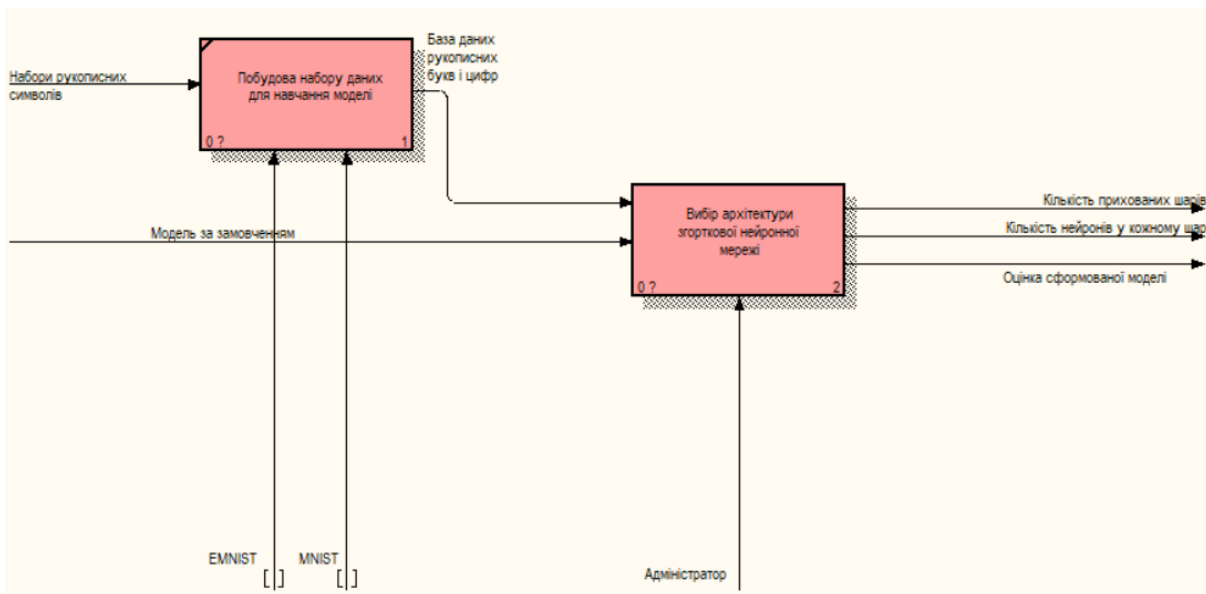


Рис. 2 – Діаграма IDEF0 «Формування моделі» (AS-IS)

Спочатку необхідно побудувати набір даних для навчання моделі. У початковому варіанті використовувалися два набори, позначені стрілками знизу для блоку 1.

Побудова та навчання моделі для розпізнавання арабських цифр здійснювалася з використанням набору MNIST (модифікований NIST) [32-35] – бази даних рукописних цифр, що налічує 70 000 зображень. У базі даних MNIST зображення різних авторів розміщуються в різних частинах для посилення унікальності. Зразки зображень із використаного набору даних MNIST наведено на рис. 3.

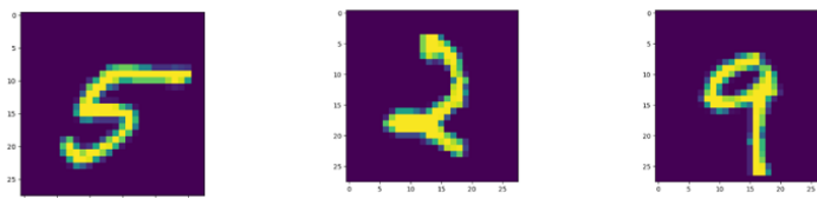


Рис. 3 – Зразки зображень із використовуваного набору даних MNIST

Побудова та навчання моделі для розпізнавання рукописних літер українського алфавіту здійснювалася з використанням набору EMNIST [12, 17] – бази даних рукописних літер та цифр, складається з шести наборів. У даній роботі було обрано саме EMNIST Letters – набір даних, що

містить 124 800 зразків для тренування та 20 800 зразків для випробувань, у тому числі великі та маленькі латинські літери. Деякі експерименти були проведені з класом цифр EMNIST, який містить збалансовану підмножину набору цифр, що містить 28 000 зразків кожної цифри. Зразки зображень із використаного набору даних EMNIST наведено на рис. 4.



Рис. 4 – Зразки зображень із використаного набору даних EMNIST

Далі необхідно вибрати архітектуру CNN. Для початкового варіанту було запропоновано лише 3 архітектури типу LeNet. Розглянемо їх основні характеристики, виконавши декомпозицію блоку 2. На рис. 5 наведено декомпозицію діаграми «Вибір архітектури згорткової нейронної мережі».

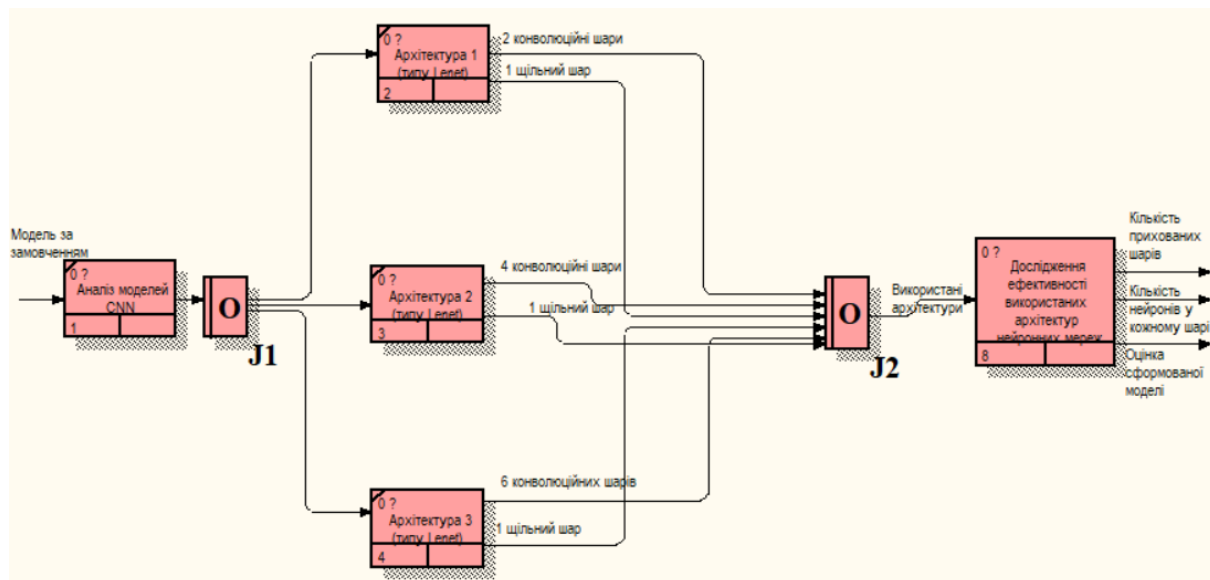


Рис. 5 – Діаграма IDEF3 «Вибір архітектури згорткової нейронної мережі» (AS-IS)

Архітектура 1 включала вхідний шар, один конволюційний блок з 2 шарів, шар підвибірки MaxPooling, шар регуляризації Dropout, 1 щільний шар, шар перетворення розмірності Flatten, ще один шар регуляризації та вихідний шар. Відмінності архітектур 2 та 3 полягали у кількості конволюційних та щільних шарів, адже усі три запропоновані архітектури були аналогами LeNet. Архітектура 2 – 4 конволюційні шари, 1 щільний шар. Архітектура 3 – 6 конволюційних шарів, 1 щільний шар. Усі вище перелічені найпростіші варіанти архітектури згорткових нейронних мереж типу LeNet для розпізнавання зображень символів представлені на рис. 6.

Саме підпроцес «Вибір архітектури згорткової нейронної мережі» буде потребувати реінжинірингу шляхом додавання інших архітектур різних типів для досягнення більшої точності розпізнавання рукописного тексту. Це буде розглянуто при побудові аналогічних діаграм для варіанту TO-BE.

Архітектура 1	Архітектура 2	Архітектура 3
<p><b>Два конволюційні шари, один щільний шар</b></p> <p>Input (28x28x1)</p> <p>conv2d, 128 filters conv2d, 128 filters MaxPooling2D Dropout</p> <p>Flatten Dense, 256 filters Dropout Dense ( output - 76 classes)</p>	<p><b>Чотири конволюційні шари, один щільний шар</b></p> <p>Input (28x28x1)</p> <p>conv2d, 64 filters conv2d, 64 filters MaxPooling2D Dropout</p> <p>conv2d, 128 filters conv2d, 128 filters MaxPooling2D Dropout</p> <p>Flatten Dense, 256 filters Dropout Dense (output – 76 classes)</p>	<p><b>Шість конволюційних шарів , один щільний шар</b></p> <p>Input (28x28x1)</p> <p>conv2d, 128 filters conv2d, 128 filters MaxPooling2D Dropout</p> <p>conv2d, 256 filters conv2d, 256 filters MaxPooling2D Dropout</p> <p>conv2d, 512 filters conv2d, 512 filters MaxPooling2D Dropout</p> <p>Flatten Dense, 1024 filters Dropout Dense (output – 76 classes)</p>

Рис. 6 – Найпростіші варіанти архітектури згорткових нейронних мереж типу Lenet

Після того, як для експерименту було обрано одну або декілька архітектур нейронної мережі із запропонованих, треба дослідити ефективність їх використання. Виконаємо декомпозицію цього підпроцесу, що проілюстровано на рис. 7.

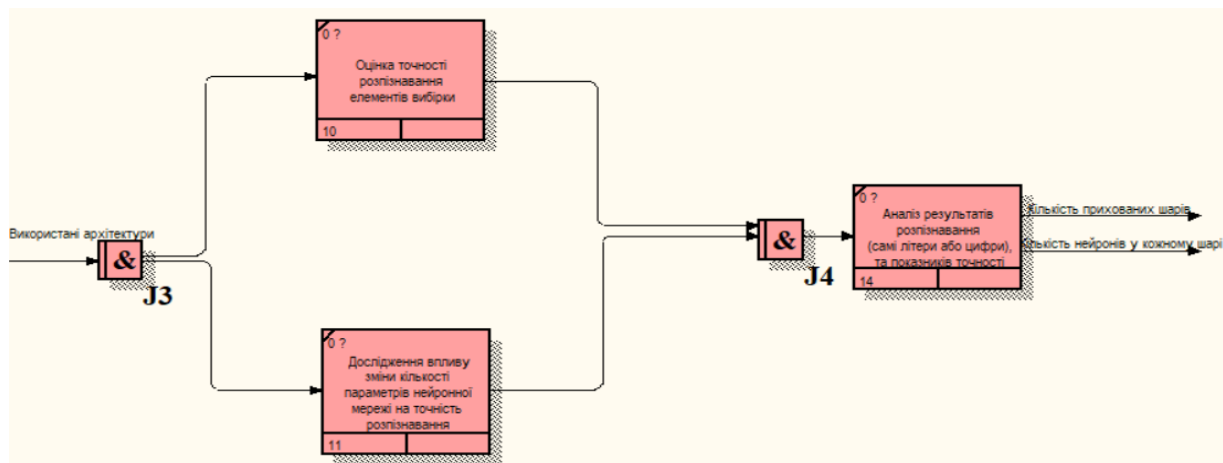


Рис. 7 – Діаграма IDEF3 «Дослідження ефективності використання архітектур нейронних мереж» (AS-IS)

У початковому варіанті планувалося провести наступні дослідження:

- оцінити точність розпізнавання елементів вибірки;
- провести дослідження впливу зміни кількості параметрів нейронної мережі на точність розпізнавання.

У рамках реінжинірингу цього підпроцесу було розширено перелік експериментальних досліджень, які дозволять більш точно оцінити ефективність використання запропонованих архітектур нейронних мереж.

Таким чином, реінжинірингу потребують два процеси – «Формування моделі» та «Формування напису», що буде розглянуто далі.

Враховуючи недоліки процесів та підпроцесів, викладені вище, було побудовано аналогічні діаграми IDEF0 та IDEF3 для варіанту «ТО-ВЕ». Діаграму першого рівня декомпозиції процесу «Методологія розпізнавання рукописних символів» для варіанту «ТО-ВЕ» наведено на рис. 8.

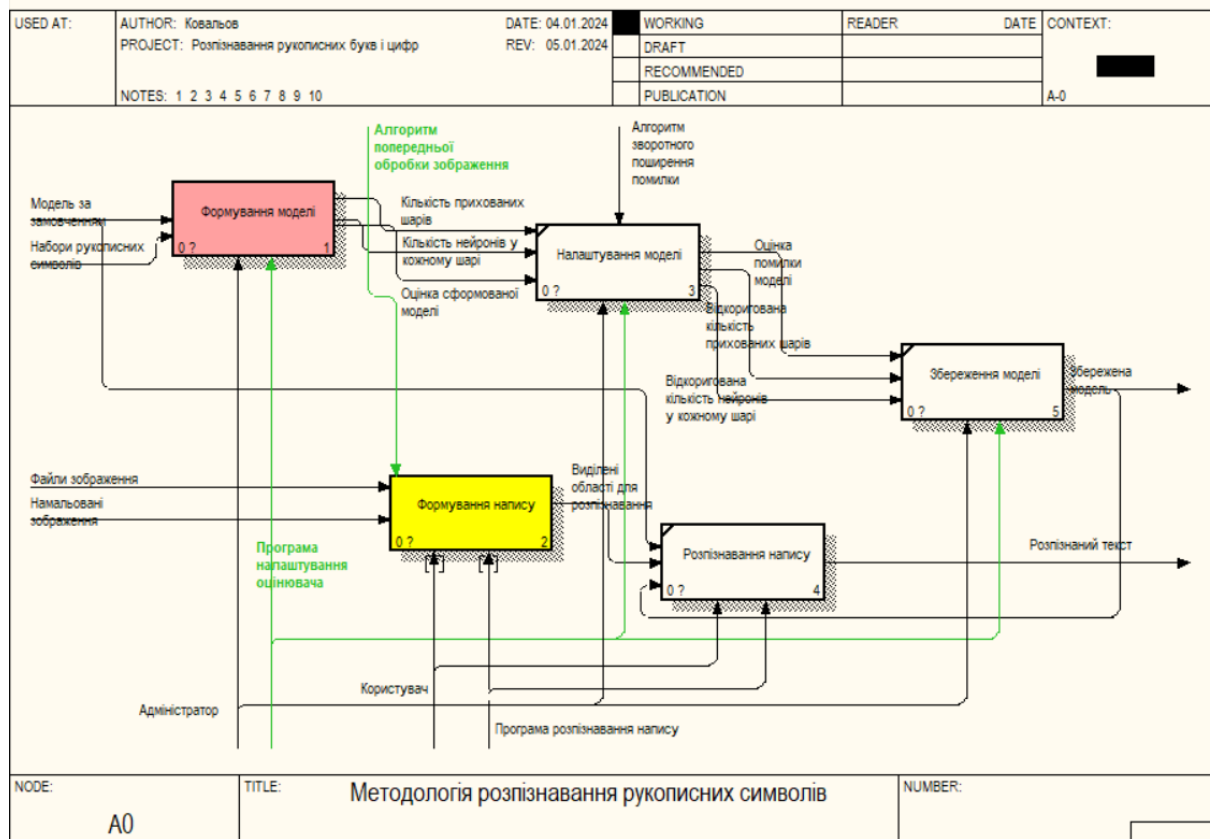


Рис. 8 – Діаграма першого рівня декомпозиції процесу «Методологія розпізнавання рукописних символів» (ТО-ВЕ)

Декомпозиція першого рівня показала, що нову стрілку керуючого впливу «Алгоритм попередньої обробки зображення» буде використано для підпроцесу «Формування напису», а нову стрілку механізму «Програма налаштування оцінювача» – для блоку «Формування моделі».

Спершу розглянемо декомпозицію процесу «Формування моделі» з урахуванням заходів з реінжинірингу, тобто для варіанту «ТО-ВЕ» (рис. 9).

У порівнянні з першим варіантом було додано наступні стрілки механізмів: «Бібліотека Pillow», «Image Data Generator пакету TensorFlow» та «Scikit-Learn» – усе для блоку «Побудова набору даних для навчання моделі», а також «Програма налаштування оцінювача» для блоку «Вибір архітектури згорткової нейронної мережі».

Розглянемо, які зміни буде передбачено для блоку «Побудова набору даних для навчання моделі».

У даній роботі для варіанту TO-VE окрім використання загально застосованих наборів EMNIST та MNIST було запропоновано створити додатковий набір даних, використавши для цього рукописні та курсивні шрифти. У результаті було створено 48 варіацій шрифтів (у тому числі з українськими гліфами). Створений набір містив 66 класів літер (по 33 – великі та маленькі



відповідно) та 10 класів цифр. До отриманих зображень також застосовували різні види деформації (масштабування, обертання та нахил, стискання та розтягування тощо), при цьому для кожного символу було згенеровано до 48 зображень. У результаті отримано набір даних з 99 072 зображень, які було відцентровано та приведено до однакового розміру (28 або 128 пікселів по ширині та висоті). Для реалізації такого підходу було використано бібліотеки Pillow та Image Data Generator з пакету TensorFlow. Пакет Scikit-Learn, а саме метод train\_test\_split, було використано для виділення навчальних та тестових вибірок.

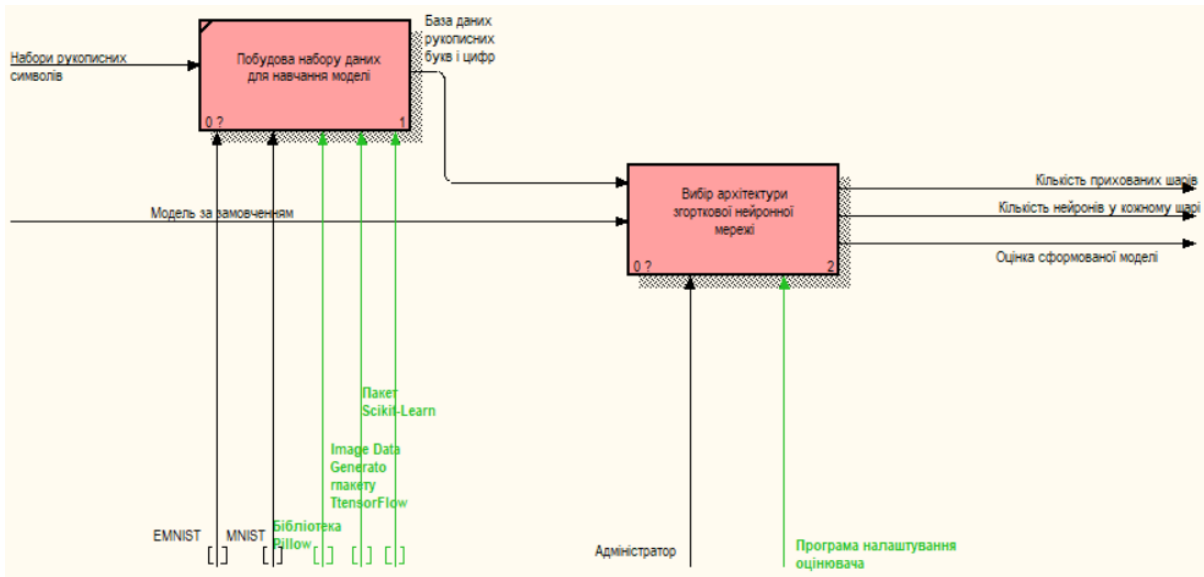


Рис. 9 – Діаграма IDEF0 «Формування моделі» (ТО-ВЕ)

Тепер розглянемо більше детально зміни, що торкнулися блоку «Вибір архітектури згорткової нейронної мережі» (рис. 10).

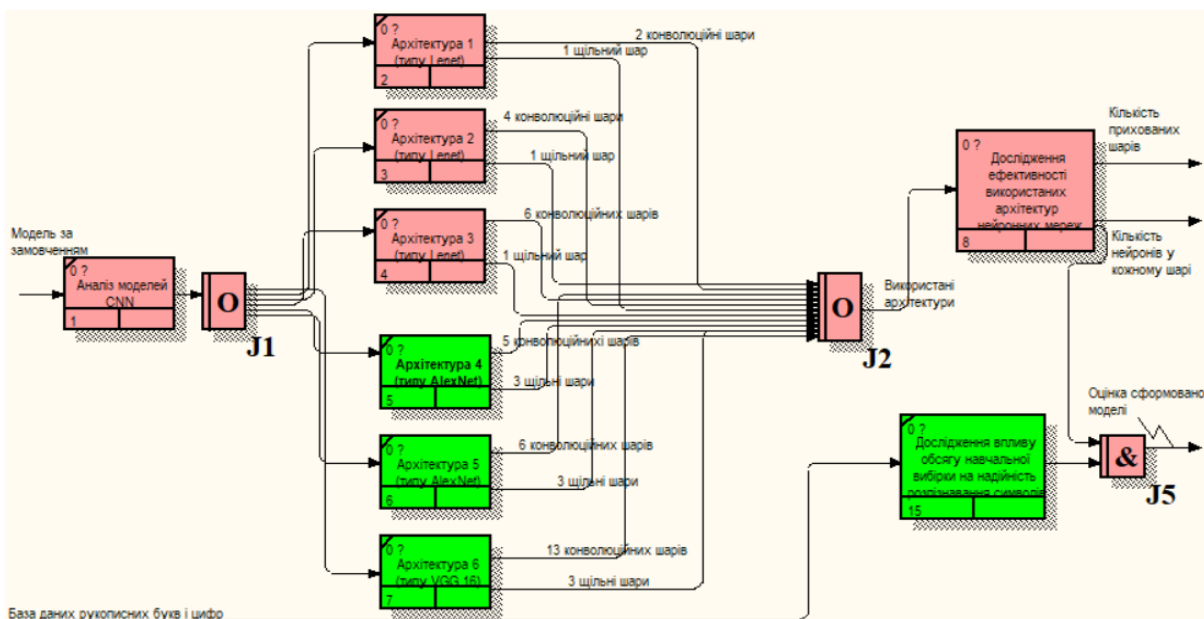


Рис. 10 – Діаграма IDEF3 «Вибір архітектури згорткової нейронної мережі» (ТО-ВЕ)

У рамках заходів з реінжинірингу крім найпростіших архітектур 1-3 типу Lenet було додано ще 3 більш складні архітектури. Розглянемо їх детальніше.

Архітектура 4 має тип AlexNet і включає 5 конволюційних шарів та 3 шари щільності. Архітектура 5 також має тип AlexNet, але включає 6 конволюційних шарів та 3 шари щільності. Архітектура 6 має 13 конволюційних шарів та 3 шари щільності, а також шари MaxPooling та Dropout. Цей варіант архітектури найбільш складний і повторює архітектуру VGG-16 стосовно одноканальних зображень. Більш детально варіанти архітектур згорткових нейронних мереж типу AlexNet та VGG 16 наведено на рис. 11.

Архітектура 4	Архітектура 5	Архітектура 6
<b>П'ять конволюційних шарів, три щільні шари</b>	<b>Шість конволюційних шарів, три щільні шари</b>	<b>Тринадцять конволюційних шарів, три щільні шари</b>
Input (28x28x1)  conv2d, 128 filters MaxPooling2D Dropout  conv2d, 256 filters MaxPooling2D Dropout  conv2d, 512 filters conv2d, 512 filters conv2d, 256 filters MaxPooling2D Dropout  Flatten Dense, 4096 filters Dense, 2048 filters Dense, 1024 filters Dropout Dense (output – 76 classes)	Input (28x28x1)  conv2d, 128 filters MaxPooling2D Dropout  conv2d, 256 filters MaxPooling2D Dropout  conv2d, 256 filters MaxPooling2D Dropout  conv2d, 512 filters conv2d, 512 filters conv2d, 512 filters MaxPooling2D Dropout  Flatten Dense, 4096 filters Dense, 2048 filters Dense, 1024 filters Dropout Dense (output – 76 classes)	Input (28x28x1)  conv2d, 512 filters conv2d, 512 filters conv2d, 512 filters conv2d, 64 filters conv2d, 64 filters MaxPooling2D Dropout  Flatten Dense, 4096 filters Dense, 4096 filters Dense, 4096 filters Dropout Dense (output - 76 classes)  conv2d, 256 filters conv2d, 256 filters conv2d, 256 filters MaxPooling2D Dropout  conv2d, 512 filters conv2d, 512 filters conv2d, 512 filters MaxPooling2D Dropout

Рис. 11 – Варіанти архітектури згорткових нейронних мереж типу AlexNet та VGG 16

Також додано новий блок «Дослідження впливу обсягу навчальної вибірки на надійність розпізнавання символів» у порівнянні з першим варіантом. Цей підпроцес включає наступні кроки:

- надати оцінку впливу обсягу навчальної вибірки на надійність розпізнавання символів;
- встановити граничну величину, що забезпечує прийнятну точність розпізнавання при генерації певної кількості зображень на одну літеру або цифру;
- надати оцінку впливу збільшення обсягу вибірки та витрати часу при навчанні моделі.

Архітектура 6, як потім покажуть дослідження, виявиться найкращою за точністю та надійністю розпізнавання тестової вибірки та реальних написів. Для більш повного аналізу результатів розпізнавання в рамках реінжинірингу було внесено зміни у підпроцес «Дослідження ефективності використання архітектур нейронних мереж» (рис. 12).

Після реінжинірингу вирішено було додати ще два пункти дослідження:

- провести дослідження впливу кількості настроюваних параметрів (при порівнянні архітектур 1 і 6 – приблизно порядок) на час навчання нейронної мережі;
- провести дослідження можливості розпізнавання символів при розпізнаванні зображень, що не належать до навчальної або тестової вибірки.

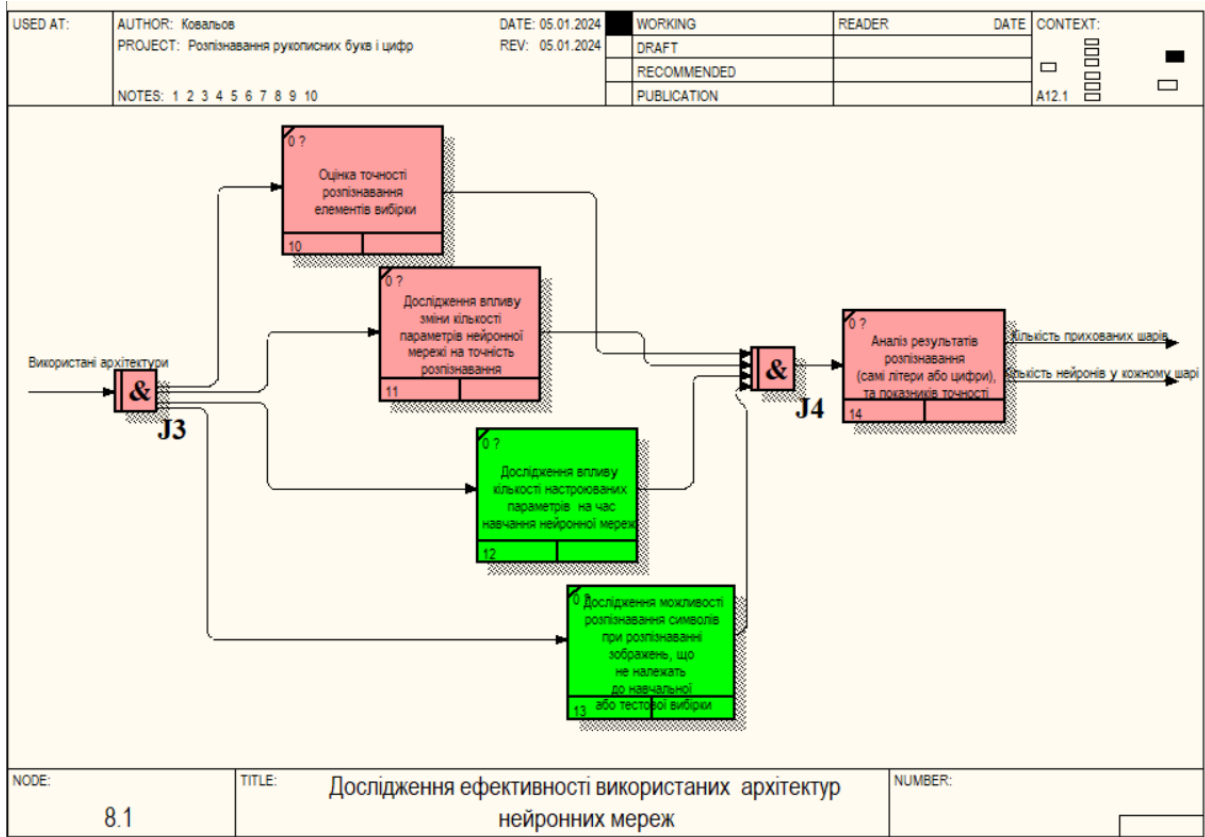


Рис. 12 – Діаграма IDEF3 «Дослідження ефективності використання архітектур нейронних мереж» (ТО-ВЕ)

Після детального розгляду процесу «Формування моделі» для варіанту «ТО-ВЕ» перейдемо до розгляду процесу «Формування напису» після реінжинірингу, в рамках якого пропонується застосування алгоритму попередньої обробки зображення. Такий алгоритм не був передбачений у початковому варіанті, тому було додано ще один рівень декомпозиції підпроцесу «Формування напису» (рис. 13).

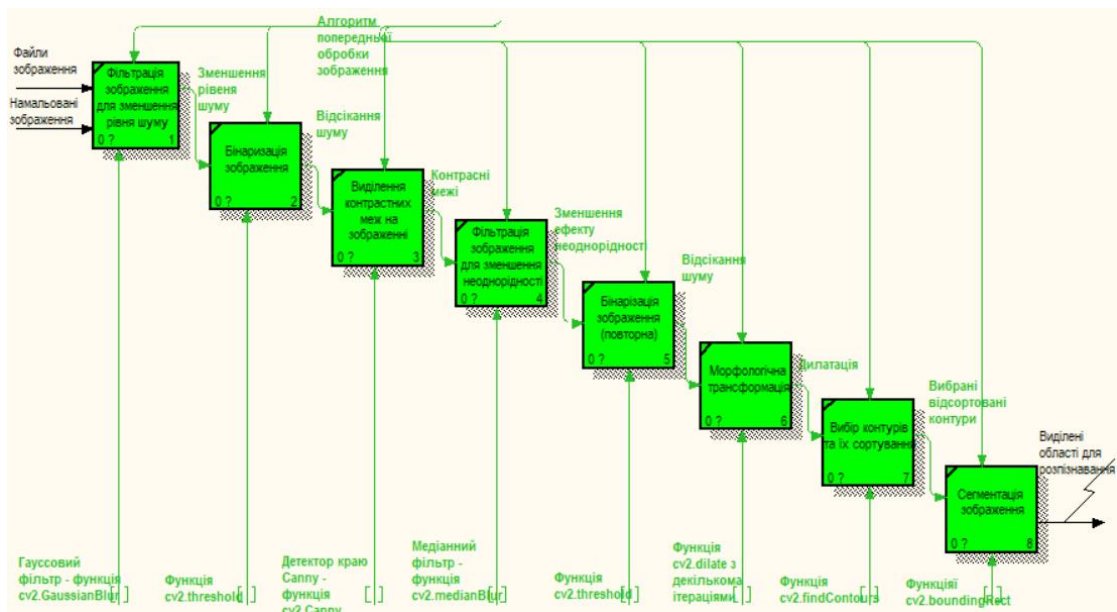


Рис. 13 – Діаграма IDEF0 «Формування напису» (ТО-ВЕ)

У рамках реінжинірингу пропонується застосувати алгоритму попередньої обробки зображення, який покроково наведений на рис. 13.

Таким чином, проведене моделювання об'єкта дослідження з використання методології IDEF для варіантів «AS-IS» та «TO-BE» після реінжинірингу, який стосувався процесів «Формування моделі» та «Формування напису».

Для оцінки ефективності розпізнавання була розроблена тестова програма на Python. За допомогою розрахункових даних, отриманих з програми, було побудовано графіки залежності точності розпізнавання та величини втрат залежно від кількості епох навчання. Результати навчання моделі наведено на рис. 14 для двох варіантів CNN: архітектури 1 – найпростішої моделі типу Lenet, а також архітектури 6 – найбільш складної, аналогу VGG-16.

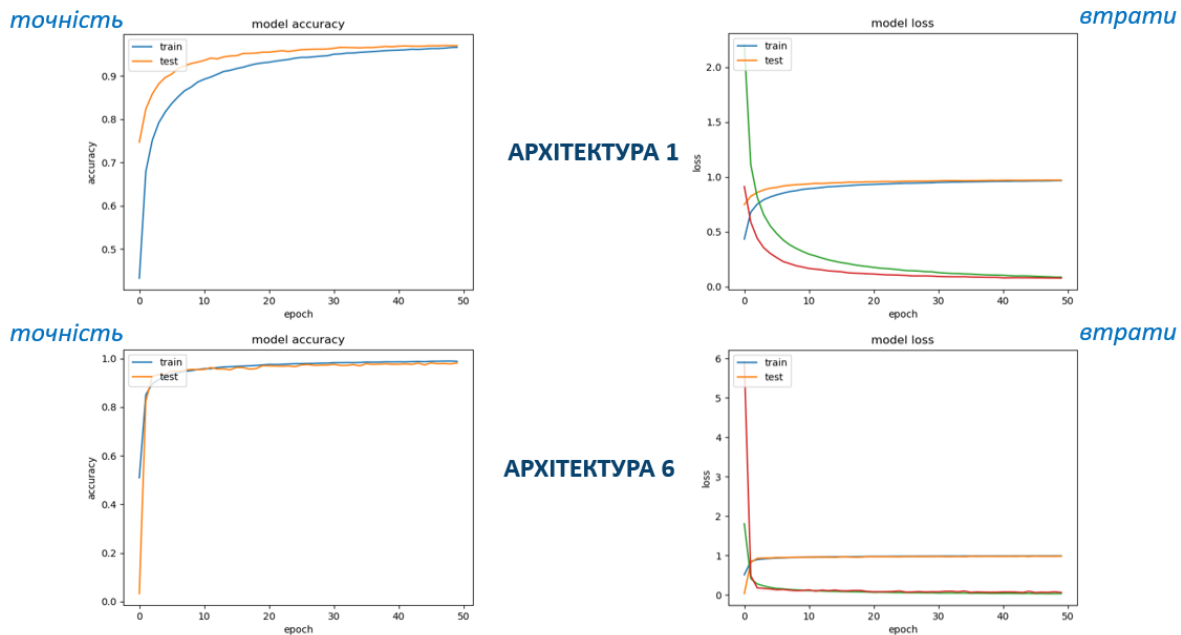


Рис. 14 – Результати точності оцінювання та витрат при навчання моделей типу Lenet (архітектура 1) та типу VGG-16 (архітектура 6)

Експериментальні дослідження показали, що усі 6 архітектур CNN під час навчання на вибірці з максимальним обсягом показують точність розпізнавання на рівні 95-99%. Підвищення точності розпізнавання досягається шляхом збільшення кількості параметрів CNN при використанні нейронних мереж більшої глибини. Час обчислень під час навчання нейронної мережі зростає зі збільшенням кількості настроюваних параметрів (при порівнянні архітектур 1 і 6 – приблизно порядок).

Значна різниця між ефективністю застосування різних архітектур демонструється при розпізнаванні зображень, що не належать до навчальної та тестової вибірки. Характерний приклад розпізнавання напису, що містить літери, наведено в табл. 1. Як видно з отриманих результатів, 100% точності розпізнавання забезпечує архітектура 6.



Спроба розпізнавання напису, що містить лише цифри, дала ще більш виражений результат точності розпізнавання зображення з ізольованими цифрами (табл. 2).

Помилки розпізнавання зустрічалися і під час використання глибоких архітектур. Приклад такої ситуації наведено у табл. 3. Архітектура 6 розпізнала другий символ напису як «Ж», а не як «Щ». Можливі також неточності у розпізнаванні малих та великих літер.

Генерація 116736 зразків для розпізнавання стала граничною величиною, при якій досягається необхідна точність. Зменшення обсягу вибірки веде до суттєвого погіршення точності розпізнавання (наприклад, зменшення вибірки в 4 рази призводить до падіння точності з 100 % до 40-60%), при цьому збільшення обсягу вибірки вимагає значно більшого часу для навчання моделі.



Таблиця 1

Результати розпізнавання напису з літерами

Напис на зображенні	Архітектура CNN	Розпізнано	Оцінка точності
 Напис ( АБІІв ) _	Архітектура 1	ДБІ6	50%
	Архітектура 2	АБіів	100%
 Виділені символи _	Архітектура 3	ДБіів	80%
	Архітектура 4	ДБіів	80%
	Архітектура 5	АБіів	100%
	Архітектура 6	АБіів	100%



Таблиця 2

Результати розпізнавання напису з цифрами

Напис на зображенні	Архітектура CNN	Розпізнано	Оцінка точності
 Напис (12345)	Архітектура 1	Іг5ц5	20%
	Архітектура 2	Іг3ц5	40%
	Архітектура 3	Іг5ц5	40%
 Виділені символи	Архітектура 4	123ц5	80%
	Архітектура 5	123ц5	80%
	Архітектура 6	12345	100%

Таблиця 3

Результати розпізнавання складного напису

Напис на зображенні	Архітектура CNN	Розпізнано	Оцінка точності
 Напис ( ШШнт )	Архітектура 1	ЩШнт	50%
	Архітектура 2	ШШнт	75%
	Архітектура 3	ШЩнт	100%
 Виділені символи	Архітектура 4	ШЖнт	75%
	Архітектура 5	МЖнт	50%
	Архітектура 6	ШЖнт	75%

### Висновки

Таким чином, у статті проаналізовано використання нейронних мереж різних архітектур для розпізнавання ізольованих рукописних символів, а саме українських літер та арабських цифр.

Дослідження ефективності застосування CNN з 6 різними архітектурами, що належать до типів Lenet, AlexNet та VGG-16, показало, що збільшення кількості згорткових шарів дозволяє підвищити надійності розпізнавання у переважній кількості випадків. При цьому найвищі показники точності та надійності розпізнавання досягнуто при використанні CNN найбільш складної архітектури типу VGG-16, яка налічувала 13 згорткових і 3 щільні шари.

Для навчання моделей додатково використано набір даних з 48 варіантів шрифтів (у тому числі з українськими гліфами), побудованих за допомогою набору рукописних і курсивних шрифтів із застосуванням бібліотеки Pillow, Image Data Generator та пакету Scikit-Learn.

Експериментально показано, що використаний у роботі набір даних з 116736 зразків для розпізнавання 76 класів цифр та літер, при цьому зазначений обсяг фактично є граничним, який дозволяє забезпечити необхідну точність розпізнавання. Експериментально доведено, що зменшення обсягу вибірки у 4 рази веде до зменшення точності розпізнавання майже в середньому у 2 рази, однак негативним наслідком збільшення обсягу вибірки є зростання витрат часу на навчання моделі.

#### Перелік використаних джерел:

1. Studies in fuzziness and soft computing. Springer Cham, 2017. Vol. 352: Optical character recognition systems for different languages with soft computing / Chaudhuri A., Mandaviya K., Badelia P., Ghosh S. 248 p. DOI: <https://doi.org/10.1007/978-3-319-50252-6>.
2. Li H., Wang P., Shen C. Toward end-to-end car license plate detection and recognition with deep neural networks. *IEEE Transactions on Intelligent Transportation Systems*. 2018. Vol. 20(3). Pp. 1126-1136. DOI: <https://doi.org/10.1109/TITS.2018.2847291>.
3. Rajavelu A., Musavi M.T., Shirvaikar M.V. A neural network approach to character recognition. *Neural Networks*. 1989. Vol. 2(5). Pp. 387-393. DOI: [https://doi.org/10.1016/0893-6080\(89\)90023-3](https://doi.org/10.1016/0893-6080(89)90023-3).
4. Image character recognition using deep convolutional neural network learned from different languages / Bai J., Chen Z., Feng B., Xu B. *2014 IEEE International Conference on Image Processing ICIP 2014*, Paris, France, 27-30 October 2014. Pp. 2560-2564. DOI: <https://doi.org/10.1109/ICIP.2014.7025518>.
5. Maitra D.S., Bhattacharya U., Parui S.K. CNN based common approach to handwritten character recognition of multiple scripts. *3th International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, 23-26 August 2015. Pp. 1021-1025. DOI: <https://doi.org/10.1109/ICDAR.2015.7333916>.
6. Bilgin Taşdemir E.F. Online Turkish handwriting recognition using synthetic data. *Avrupa Bilim Ve Teknoloji Dergisi*. 2021. Vol. 32. Pp. 649-656. DOI: <https://doi.org/10.31590/ejosat.1039846>.
7. Handwritten Kazakh and Russian (HKR) database for text recognition / D. Nurseitov, K. Bostanbekov, D. Kurmankhojayev, A. Alimova, A. Abdallah, R. Tolegenov. *Multimedia Tools and Applications*. 2021. Vol. 80. Pp. 33075-33097. DOI: <https://doi.org/10.1007/s11042-021-11399-6>.
8. Abdallah A., Hamada M., Nurseitov D. Attention-based fully gated CNN-BGRU for Russian handwritten text. *Journal of Imaging*. 2020. Vol. 6(12). Pp. 1-23. DOI: <https://doi.org/10.3390/jimaging6120141>.
9. Ullah Z., Jamjoom M. An intelligent approach for Arabic handwritten letter recognition using convolutional neural network. *PeerJ Computer Science*. 2022. Vol. 8. Pp. 1-22. DOI: <https://doi.org/10.7717/peerj-cs.995>.
10. Handwritten letter recognition using artificial intelligence / Jeevitha D., Muthu S., Nila I., Santhoshi V. *International journal for research in applied science and engineering technology*. 2022. Vol. 10. Pp. 2752-2758. DOI: <https://doi.org/10.22214/ijraset.2022.42949>.
11. An exploratory study on the handwritten allographic features of multi-ethnic population with different educational backgrounds / L. Gannetion, K.Y. Wong, P.Y. Lim, K.H. Chang, A.F.L. Abdallah. *PloS one*. 2022. Vol. 17(10). Pp. 1-16. DOI: <https://doi.org/10.1371/journal.pone.0268756>.
12. EMNIST: Extending MNIST to handwritten letters / Cohen G., Afshar S., Tapson J., Van Schaik A. *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, 14-19 May 2017. Pp. 2921-2926. DOI: <https://doi.org/10.48550/arxiv.1702.05373>.
13. Backpropagation applied to handwritten zip code recognition / Y. LeCun et al. *Neural Computation*. 1989. Vol. 1. Iss. 4. Pp. 541-551. DOI: <https://doi.org/10.1162/neco.1989.1.4.541>.
14. Núñez Fernández D., Hosseini S. Real-time handwritten letters recognition on an embedded computer using ConvNets. *IEEE Sciences and Humanities International Research Conference (SHIRCON)*, Lima, Peru, 20-22 November 2018. Pp. 1-4. DOI: <https://doi.org/10.1109/SHIRCON.2018.8592981>.

15. Krizhevsky A., Sutskever I., Hinton G. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 2012. Vol. 60. Iss. 6. Pp. 84-90. DOI: <https://doi.org/10.1145/3065386>.
16. Going deeper with convolutions / C. Szegedy et al. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 7-12 June 2015. Pp. 1-9. DOI: <https://doi.org/10.1109/CVPR.2015.7298594>.
17. Baldominos A., Sáez Y., Isasi P. A survey of handwritten character recognition with MNIST and EMNIST. *Applied Sciences*. 2019. Vol. 9(15). Pp. 1-16. DOI: <https://doi.org/10.3390/app9153169>.
18. Handwritten indic character recognition using capsule networks / B. Mandal, S. Dubey, S. Ghosh, R. Sarkhel, N. Das. *2018 IEEE Applied Signal Processing Conference (ASPCON)*, Kolkata, India, 07-09 December 2018. Pp. 304-308. DOI: <https://doi.org/10.1109/ASPCON.2018.8748550>.
19. Recognition of isolated characters across different input interfaces using 2D DCNN / K. Yadav, A. Kirupakaran, S. Barlaskar, N. Ahmad, R. Laskar, M. Bhuyan. *2021 IEEE Region 10 Conference (TENCON)*, Auckland, New Zealand, 07-10 December 2021. Pp. 504-509. DOI: <https://doi.org/10.1109/TENCON54134.2021.9707451>.
20. He K., Girshick R., Dollár P. Rethinking imagenet pre-training. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Korea (South), 27 October - 02 November 2019. Pp. 4918-4927. DOI: <https://doi.org/10.1109/ICCV.2019.00502>.
21. Albattah W., Albahli S. Intelligent Arabic handwriting recognition using different standalone and hybrid CNN architectures. *Applied Sciences*. 2022. Vol. 12. Pp. 1-23. DOI: <https://doi.org/10.3390/app121910155>.
22. Performance analysis of state-of-the-art convolutional neural network architectures in bangla handwritten character recognition / T. Ghosh, Min-Ha-Zul Abedin, H. Al Banna, N. Mumenin, M. Abu Yousuf. *Pattern recognition and image analysis*. 2021. Vol. 31. Pp. 60-71. DOI: <https://doi.org/10.1134/S1054661821010089>.
23. Bhardwaj A., Ravendra Singh R. Handwritten devanagari character recognition using deep learning – convolutional neural network (CNN) model. *PalArch's Journal of Archaeology of Egypt/Egyptology*. 2020. Vol. 17(6). Pp. 7965-7984.
24. Handwritten devanagari character recognition using modified Lenet and Alexnet convolution neural networks / Duddela P., Vasanth Kumar M.R., Ramana K., Vidhyacharan B. *Wireless Personal Communications*. 2022. Vol. 122. Pp. 349-378. DOI: <https://doi.org/10.1007/s11277-021-08903-4>.
25. Recognizing arabic handwritten literal amount using convolutional neural networks / A. Korichi, S. Sihem, T. Najiba, Z. Ramzi, A. Oussama. *Artificial Intelligence and Its Applications : Proceeding of the 2nd International Conference, EL-Oued, Algeria, 28-30 September 2021*. Pp. 153-165. DOI: [https://doi.org/10.1007/978-3-030-96311-8\\_15](https://doi.org/10.1007/978-3-030-96311-8_15).
26. A new Arabic handwritten character recognition deep learning system (AHCR-DLS) / Balaha H., Sabry M., Ali H., Badawy M. *Neural Computing and Applications*. 2020. Vol. 33. Pp. 6325-6367. DOI: <https://doi.org/10.1007/s00521-020-05397-2>.
27. Samra A., Al Amin G., Hadi O. An optimized deep residual network with a depth concatenated block for handwritten characters classification. *Computers, Materials & Continua*. 2021. Vol. 680. Pp. 1-28. DOI: <https://doi.org/10.32604/cmc.2021.015318>.
28. Classification of handwritten names of cities and handwritten text recognition using various deep learning models / D. Nurseitov, K. Bostanbekov, M. Kanatov, A. Alimova, A. Abdallah, G. Abdimanap. *Advances in Science, Technology and Engineering Systems Journal (ASTESJ)*. 2021. Vol. 5. Iss. 5. Pp. 934-943. DOI: <https://doi.org/10.25046%2Faj0505114>.
29. Vovchuk O., Kyrychenko M. Recognition of handwritten cyrillic letters using PCA. 2019. 6 p.
30. Economic efficiency of innovative projects of CNN modified architecture application / V. Khavalko, V. Mykhailyshyn, R. Zhelizniak, I. Kovtyk, A. Mazur. *CEUR Workshop Proceedings. Vol. 2654: Proceedings of the International workshop on cyber hygiene (CybHyg-2019) co-located with 1st International conference on cyber hygiene and conflict management in global information networks (CyberConf 2019)*, Kyiv, Ukraine, 30 November 2019. Pp. 182-193.
31. Розпізнавання рукописних українських літер та цифр з використанням синтетичного набору даних та згорткових нейронних мереж / Є. Чичкарьов, О. Зінченко, О. Балалаєва, А. Сергієнко, О. Ковальов. *Grail of Science*. 2023. Vol. 23. Pp. 241-253. DOI: <https://doi.org/10.36074/grail-of-science.23.12.2022.36>.

32. Study and observation of the variations of accuracies for handwritten digits recognition with various hidden layers and epochs using convolutional neural network / Arif R.B., Siddique M.A.B., Khan M.M.R., Oishe M.R. *4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEICT)*, Dhaka, Bangladesh, 13-15 September 2018. Pp. 112-117. DOI: <https://doi.org/10.1109/CEEICT.2018.8628078>.
33. Siddique F., Sakib S., Siddique M.A.B. Handwritten digit recognition using convolutional neural network in python with tensorflow and observe the variation of accuracies for various hidden layers. *Preprints*. 2019. Pp. 1-6. DOI: <https://doi.org/10.20944/preprints201903.0039.v1>.
34. LeCun Y., Cortes C., Burges C.J.C. The MNIST database of handwritten digits. URL: <http://yann.lecun.com/exdb/mnist> (дата звернення: 30.05.2023).
35. Grother P.J. NIST special database 19 – handprinted forms and characters database. National Institute of Standards and Technology (NIST), Tech. Rep., 1995. DOI: <http://doi.org/10.18434/T4H01C>.

#### References:

1. A. Chaudhuri, K. Mandaviya, P. Badelia, and S. Ghosh, «Optical character recognition systems for different languages with soft computing», in *Studies in fuzziness and soft computing* (vol. 352), Springer Cham Publ., 2017. doi: **10.1007/978-3-319-50252-6**.
2. H. Li, P. Wang, and C. Shen, «Toward end-to-end car license plate detection and recognition with deep neural networks», *IEEE Transactions on Intelligent Transportation Systems*, vol. 20(3), pp. 1126-1136, 2018. doi: **10.1109/TITS.2018.2847291**.
3. A. Rajavelu, M.T. Musavi, and M.V. Shirvaikar, «A neural network approach to character recognition», *Neural Networks*, vol. 2(5), pp. 387-393, 1989. doi: **10.1016/0893-6080(89)90023-3**.
4. J. Bai, Z. Chen, B. Feng, and B. Xu, «Image character recognition using deep convolutional neural network learned from different languages», in *2014 IEEE International Conference on Image Processing ICIP 2014*, Paris, France, 2014, pp. 2560-2564. doi: **10.1109/ICIP.2014.7025518**.
5. D.S. Maitra, U. Bhattacharya, and S.K. Parui, «CNN based common approach to handwritten character recognition of multiple scripts», in *3th International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, 2015, pp. 1021-1025. doi: **10.1109/icdar.2015.7333916**.
6. E.F. Bilgin Taşdemir, «Online Turkish Handwriting Recognition Using Synthetic Data», *Avrupa Bilim Ve Teknoloji Dergisi*, vol. 32, pp. 649-656, 2021. doi: **10.31590/ejosat.1039846**.
7. D. Nurseitov, K. Bostanbekov, D. Kurmankhojayev, A. Alimova, A. Abdallah, and R. Tolegenov, «Handwritten Kazakh and Russian (HKR) database for text recognition», *Multimedia tools and applications*, vol. 80, pp. 33075-33097, 2021. doi: **10.1007/s11042-021-11399-6**.
8. A. Abdallah, M. Hamada, and D. Nurseitov, «Attention-Based Fully Gated CNN-BGRU for Russian Handwritten Text», *Journal of Imaging*, vol. 6(12), pp. 1-23, 2020. doi: **10.3390/jimaging6120141**.
9. Z. Ullah, and M. Jamjoom, «An intelligent approach for Arabic handwritten letter recognition using convolutional neural network», *PeerJ Computer Science*, vol. 8, pp. 1-22, 2022. doi: **10.7717/peerj-cs.995**.
10. D. Jeevitha, S. Muthu, I. Nila, and V. Santhoshi, «Handwritten letter recognition using artificial intelligence», *International journal for research in applied science and engineering technology*, vol. 10, pp. 2752-2758, 2022. doi: **10.22214/ijraset.2022.42949**.
11. L. Gannetion, K.Y. Wong, P.Y. Lim, K.H. Chang, and A.F.L. Abdullah, «An exploratory study on the handwritten allographic features of multi-ethnic population with different educational backgrounds», *PloS one*, vol. 17(10), pp. 1-16, 2022. doi: **10.1371/journal.pone.0268756**.
12. G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, «EMNIST: Extending MNIST to handwritten letters», in *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, 2017, pp. 2921-2926. doi: **10.48550/arxiv.1702.05373**.
13. Y. LeCun et al., «Backpropagation applied to handwritten zip code recognition», *Neural Computation*, vol. 1, iss. 4, pp. 541-551, 1989. doi: **10.1162/neco.1989.1.4.541**.
14. D. Núñez Fernández, and S. Hosseini, «Real-time handwritten letters recognition on an embedded computer using ConvNets», in *IEEE Sciences and Humanities International Research Conference (SHIRCON)*, Lima, Peru, 2018, pp. 1-4. doi: **10.1109/SHIRCON.2018.8592981**.
15. A. Krizhevsky, I. Sutskever, and G. Hinton, «ImageNet classification with deep convolutional neural networks», *Communications of the ACM*, vol. 60, iss. 6, pp. 84-90, 2012. doi: **10.1145/3065386**.



16. C. Szegedy et al., «Going deeper with convolutions», in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1-9. doi: **10.1109/CVPR.2015.7298594**.
17. A. Baldominos, Y. Sáez, and P. Isasi, «A survey of handwritten character recognition with MNIST and EMNIST», *Applied Sciences*, vol. 9(15), pp. 1-16, 2019. doi: **10.3390/app9153169**.
18. B. Mandal, S. Dubey, S. Ghosh, R. Sarkhel, and N. Das, «Handwritten indic character recognition using capsule networks», in 2018 IEEE Applied Signal Processing Conference (ASPCON), Kolkata, India, 2018, pp. 304-308. doi: **10.1109/ASPCON.2018.8748550**.
19. K. Yadav, A. Kirupakaran, S. Barlaskar, N. Ahmad, R. Laskar, and M. Bhuyan, «Recognition of isolated characters across different input interfaces using 2D DCNN», in 2021 IEEE Region 10 Conference (TENCON), Auckland, New Zealand, 2021, pp. 504-509. doi: **10.1109/TENCON54134.2021.9707451**.
20. K. He, R. Girshick, and P. Dollár, «Rethinking imagenet pre-training», in Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea (South), 2019, pp. 4918-4927. doi: **10.1109/ICCV.2019.00502**.
21. W. Albattah, and S. Albahli, «Intelligent Arabic Handwriting Recognition Using Different Standalone and Hybrid CNN Architectures», *Applied Sciences*, vol. 12, pp. 1-23, 2022. doi: **10.3390/app121910155**.
22. T. Ghosh, Min-Ha-Zul Abedin, H. Al Banna, N. Mumenin, and M. Abu Yousuf, «Performance analysis of state of the art convolutional neural network architectures in bangla handwritten character recognition», *Pattern recognition and image analysis*, vol. 31, pp. 60-71, 2021. doi: **10.1134/S1054661821010089**.
23. A. Bhardwaj, and R. Ravendra Singh, «Handwritten devanagari character recognition using deep learning – convolutional neural network (CNN) model», *PalArch's Journal of Archaeology of Egypt/Egyptology*, vol. 17(6), pp. 7965-7984, 2020.
24. P. Duddela, M.R. Vasanth Kumar, K. Ramana, and B. Vidhyacharan, «Handwritten Devanagari Character Recognition Using Modified Lenet and Alexnet Convolution Neural Networks», *Wireless Personal Communications*, vol. 122, pp. 349-378, 2022. doi: **10.1007/s11277-021-08903-4**.
25. A. Korichi, S. Sihem, T. Najiba, Z. Ramzi, and A. Oussama, «Recognizing arabic handwritten literal amount using convolutional neural networks», in Proceeding of the 2nd International Conference Artificial Intelligence and Its Applications, EL-Oued, Algeria, 2021, pp. 153-165. doi: **10.1007/978-3-030-96311-8\_15**.
26. H. Balaha, M. Sabry, H. Ali, and M. Badawy, «A new Arabic handwritten character recognition deep learning system (AHCR-DLS)», *Neural Computing and Applications*, vol. 33, pp. 6325-6367, 2020. doi: **10.1007/s00521-020-05397-2**.
27. A. Samra, G. Al Amin, and O. Hadi, «An optimized deep residual network with a depth concatenated block for handwritten characters classification», *Computers, Materials & Continua*, vol. 680, pp. 1-28, 2021. doi: **10.32604/cmc.2021.015318**.
28. D. Nurseitov, K. Bostanbekov, M. Kanatov, A. Alimova, A. Abdallah, and G. Abdimanap, «Classification of handwritten names of cities and handwritten text recognition using various deep learning models», *Advances in Science, Technology and Engineering Systems Journal (ASTESJ)*, vol. 5, iss. 5, pp. 934-943, 2021. doi: **10.25046/2Faj0505114**.
29. O. Vovchuk, and M. Kyrychenko, *Recognition of handwritten cyrillic letters using PCA*. 2019.
30. V. Khavalko, V. Mykhailyshyn, R. Zhelizniak, I. Kovtyk, and A. Mazur, «Economic efficiency of innovative projects of CNN modified architecture application», in CEUR Workshop Proceedings. Vol. 2654: Proceedings of the International workshop on cyber hygiene (CybHyg-2019) co-located with 1st International conference on cyber hygiene and conflict management in global information networks (CyberConf 2019), Kyiv, Ukraine, 2019, pp. 182-193.
31. Ye. Chychkarov, O. Zinchenko, O. Balalaieva, A. Serhienko, and O. Kovalov, «Rozpiznavannia rukopysnykh ukrainskykh liter ta tsyfr z vykorystanniam syntetychnoho naboru danykh ta zghortkovykh neironnykh merezh» [«Recognition of handwritten Ukrainian letters and numbers using a synthetic dataset and convolutional neural networks»], *Grail of Science*, vol. 23, pp. 241-253, 2023. doi: **10.36074/grail-of-science.23.12.2022.36**. (Ukr.)
32. R.B. Arif, M.A.B. Siddique, M.M.R. Khan, and M.R. Oishe, «Study and observation of the variations of accuracies for handwritten digits recognition with various hidden layers and epochs using

- convolutional neural network», in 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEICT), Dhaka, Bangladesh, 2018, pp. 112-117. doi: **10.1109/CEEICT.2018.8628078**.
33. F. Siddique, S. Sakib, and M.A.B. Siddique, «Handwritten digit recognition using convolutional neural network in python with tensorflow and observe the variation of accuracies for various hidden layers», *Preprints*, pp. 1-6, 2019. doi: **10.20944/preprints201903.0039.v1**.
34. Y. LeCun, C. Cortes, and C.J.C. Burges, The MNIST database of handwritten digits [Online]. Available: <http://yann.lecun.com/exdb/mnist>. Accessed on: May 30, 2023.
35. P.J. Grother, *NIST special database 19 – handprinted forms and characters database*, National Institute of Standards and Technology (NIST), Tech. Rep. Publ., 1995. doi: **10.18434/T4H01C**.

Рецензент: О.І. Проніна,  
канд. техн. наук, доц., ДВНЗ «ПДТУ»

Стаття надійшла 01.10.2023

Стаття прийнята 05.11.2023

УДК 004.896

doi: 10.31498/2225-6733.47.2023.299990

© Сергієнко А.В.<sup>1</sup>, Єфімов П.С.<sup>2</sup>, Обиденний Є.О.<sup>3</sup>, Бешта Л.В.<sup>4</sup>

#### ВИКОРИСТАННЯ ХМАРНИХ ТЕХНОЛОГІЙ ДЛЯ АВТОМАТИЧНОЇ РОЗСИЛКИ ПОВІДОМЛЕНЬ В TELEGRAM

*В статті розглянуто питання автоматизації розповсюдження повідомлень в месенджері Telegram за допомогою хмарних технологій. Описано загальну проблему розсилки повідомлень декількома каналами. Зроблений огляд останніх досліджень та публікацій сучасних технологій обміну повідомленнями в Telegram, проаналізовані бібліотеки, фреймворки та сучасні патерни. Запропоновано методикою використання потужностей AWS EC2 для створення масштабованого та надійного рішення для автоматизованої розсилки повідомлень. Детально описаний процес створення програми за запропонованою методикою. Описано її архітектуру та взаємодію модулів. Архітектура структурована таким чином, щоб інкапсулювати різні функції, кожна з яких служить певній меті в ширшому контексті автоматизованого розповсюдження повідомлень на платформі Telegram. Наводяться характеристики обраного сервісу AWS, обраної операційної системи, мови програмування та використаних бібліотек. Продемонстровано використання автоматичного масштабування в AWS, що автоматизує налаштування ресурсів на основі попередньо визначених критеріїв та забезпечує ефективне рішення для обробки коливань попиту та підтримки оптимальної продуктивності системи. Використані мова програмування Python та бібліотека Telethon. Показано взаємодію AWS EC2 з API Telegram та використання модуля Asynchronous Message Forwarding для організації обробки декількох каналів одночасно. Описані можливості програми – налаштування облікових записів Telegram, пошук повідомлень, автоматизована переадресація на кілька каналів та інші дії. Показані результати проведених тестувань розробленої програми, що показують, що автоматизація процесу пересилки повідомлень збільшує швидкість їх доставки у велику кількість чатів на 120 секунд, а це ефективніше в 40 разів, а її*

<sup>1</sup> канд. техн. наук, ДВНЗ «Приазовський державний технічний університет», м. Дніпро, ORCID: 0000-0003-1328-2572, [sergienko\\_a\\_v@pstu.edu](mailto:sergienko_a_v@pstu.edu)

<sup>2</sup> магістрант, ДВНЗ «Приазовський державний технічний університет», м. Дніпро, [efor1999@gmail.com](mailto:efor1999@gmail.com)

<sup>3</sup> асистент, НТУ «Дніпровська політехніка», м. Дніпро, ORCID: 0000-0001-9065-6369, [Obydennyi.Ye.O@nmu.one](mailto:Obydennyi.Ye.O@nmu.one)

<sup>4</sup> асистент, НТУ «Дніпровська політехніка», ORCID: 0000-0003-1461-4399, [Beshta.l.v@nmu.one](mailto:Beshta.l.v@nmu.one)