

УДК 004.75

DOI 10.30837/bi.2022.1(98).02

О.Є. Українець¹, С.Г. Удовенко², Л.Е. Чала¹, В.Л. Шергін¹¹ХНУРЕ, м. Харків, Україна, oleksandr.ukrainets@nure.ua,
ORCID iD: 0000-0001-8936-6591²ХНЕУ ім. С. Кузнеця, м. Харків, Україна, serhiy.udovenko@hneu.net,
ORCID iD: 0000-0001-5945-8647¹ХНУРЕ, м. Харків, Україна, larysa.chala@nure.ua, ORCID iD: 0000-0002-9890-4790¹ХНУРЕ, м. Харків, Україна, vadim.shergin@nure.ua, ORCID iD: 0000-0002-4388-8180

НЕЙРОМЕРЕЖЕВА ТЕХНОЛОГІЯ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ КЛІТИННИХ АВТОМАТІВ

Пропонується технологія розпізнавання зображень, яка основана на нейромережевій обробці статичних та динамічних зображень з використанням клітинних автоматів (КА). Розглянуто принципи реалізації операцій обробки зображень з застосуванням КА (зокрема, операцій попередньої обробки, виділення контурів та розпізнавання). Проаналізовано особливості побудови згорткових нейронних мереж (ЗНМ) та їх використання для розпізнавання зображень. Розглянуто можливості інтеграції КА в нейромережеву модель класифікації та розпізнавання зображень. Наведено результати розробки та програмної реалізації нейромережевої технології розпізнавання зображень з використанням ЗНМ та КА. Здійснено моделювання запропонованої технології (на прикладі проблеми розпізнавання зорових об'єктів для користувачів з обмеженими можливостями зорового сприйняття) розробки та програмної реалізації нейромережевої технології розпізнавання зображень з використанням ЗНМ та КА.

ОБРОБКА ЗОБРАЖЕНЬ, КЛІТИННІ АВТОМАТИ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, APPLE VISION, РОЗПІЗНАВАННЯ ЗОРОВИХ ОБ'ЄКТІВ

Українець О.Є., Удовенко С.Г., Чала Л.Э., Шергін В.Л. Нейросетевая технология классификации изображений с использованием клеточных автоматов. Предлагается технология распознавания изображений, основанная на нейросетевой обработке статических и динамических изображений с использованием клеточных автоматов (КА). Рассмотрены принципы реализации операций обработки изображений с применением КА (в частности, операций предварительной обработки, выделения контуров и распознавания). Проанализированы особенности построения сверточных нейронных сетей (СНС) и их использование для распознавания изображений. Рассмотрены возможности интеграции КА в нейросетевую модель классификации и распознавания изображений. Представлены результаты разработки и программной реализации нейросетевой технологии распознавания изображений с использованием СНС и КА. Осуществлено моделирование предложенной технологии (на примере проблемы распознавания зрительных объектов для пользователей с ограниченными возможностями зрительного восприятия).

ОБРАБОТКА ИЗОБРАЖЕНИЙ, КЛЕТОЧНЫЕ АВТОМАТЫ, СВЕРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ, APPLE VISION, РАСПОЗНАВАНИЕ ЗРИТЕЛЬНЫХ ОБЪЕКТОВ

Ukrainets O.Ye., Udovenko S.G., Chala L.E., Shergin V.L. Neural network technology of image recognition using cellular automata. An image recognition technology based on neural network processing of static and dynamic images using cellular automata (CA) is proposed. The principles of the implementation of image processing operations with the use of CA (in particular, pre-processing operations, contour selection and recognition) are considered. The peculiarities of the construction of convolutional neural networks (CNN) and their use for image recognition are analyzed. The possibilities of integrating CA into a neural network model of image classification and recognition are considered. The results of the development and software implementation of the neural network technology of image recognition using SNM and CA are given. Modeling of the proposed technology was carried out (on the example of the problem of recognizing visual objects for users with limited visual perception). development and software implementation of neural network technology for image recognition using SNM and CA.

IMAGE PROCESSING, CELLULAR AUTOMATA, CONVOLUTIONAL NEURAL NETWORKS, APPLE VISION, VISUAL OBJECT RECOGNITION

Вступ

В даний час отримали значний розвиток теоретичні та практичні дослідження проблеми розпізнавання образів у реальному часі [1, 2]. Розпізнавання образів вирішує два основних завдання: ідентифікацію та верифікацію. Завдання ідентифікації зазвичай полягає в послідовному порівнянні зображень виявленого об'єкту з зображеннями аналогічних об'єктів, що зберігаються в базі даних. Мета верифікації полягає

в порівнянні отриманого результуючого зображення з відповідним еталоном (в разі його наявності) та кількісній оцінці їх близькості, що дозволяє віднести виявлений об'єкт до деякого класу.

При цьому виникає необхідність вирішення цілої низки складних завдань, зокрема, завдання динамічного пошуку та детекції об'єкта в часі та просторі з використанням наявних відеокамер. В разі використання камер сучасних мобільних телефонів вони

мають бути оснащені додатковим спеціалізованим програмним забезпеченням (ПЗ) для оперативного розпізнавання аналізованих зображень. Таке ПЗ може сприяти, наприклад, вирішенню проблеми розпізнавання зорових об'єктів для користувачів з обмеженими можливостями зорового сприйняття. При цьому мають застосовуватись інтелектуальні технології, що дозволяють користувачам компенсувати проблеми з зором шляхом часткового перенесення їх зорових функцій на обчислювальну техніку.

Основною перевагою такого підходу є доступність мобільних пристроїв у різних цінових сегментах. Смартфони з камерами, що дозволяють отримувати дані у вигляді зображень або відео є цілком доступними для більшості потенційних користувачів. При цьому слід відзначити варіативність використання різних за ціною мобільних пристроїв, що передбачають можливість завантаження та використання програмних засобів розпізнавання аналізованих зображень. Такі програмні засоби мають низку переваг та недоліків. До переваг можна віднести відносно простоту й швидкість їх розробки, оскільки для мобільних пристроїв на основі сучасних операційних систем (зокрема, операційних систем iOS та Android) існують окремі мови програмування з великою кількістю бібліотек зразків зображень для навчання нейромережових моделей, що використовуються для розпізнавання та класифікації виявлених об'єктів.

Недоліками такого рішення є обмеженість якості зображень, що передаються через камеру мобільних пристроїв. Таке обмеження обумовлюється, зокрема, необхідністю постійного переміщення мобільного пристрою в процесі пошуку об'єкта, через що на якість отримуваної відеоінформації суттєво впливають динамічні викривлення зображень. У такому разі виникає необхідність застосування інтелектуальних засобів, які можуть нівелювати або зменшувати вплив викидів та перешкод в процесі розпізнавання об'єктів. До таких засобів слід віднести використання клітинних автоматів (КА) в схемах нейромережового навчання систем розпізнавання.

У даній роботі пропонується технологія розпізнавання статичних та динамічних зображень, яка основана на використанні узагальненої моделі розпізнавання образів за допомогою поєднання можливостей КА та згорткових нейронних мереж (ЗНМ). Метою застосування цієї технології є підвищення точності та швидкості розпізнавання складних зображень.

Відповідно до поставленої мети, необхідно вирішити наступні завдання:

- аналіз принципів обробки зображень з використанням КА;
- аналіз особливостей використання ЗНМ для розпізнавання зображень;
- аналіз можливості інтеграції КА в нейромережову модель розпізнавання зображень;

– розробка та програмна реалізація нейромережової технології розпізнавання зображень з використанням ЗНМ та КА;

– моделювання запропонованої технології (на прикладі проблеми розпізнавання зорових об'єктів для користувачів з обмеженими можливостями зорового сприйняття).

1. Принципи обробки зображень з використанням клітинних автоматів

Останнім часом набувають розвиток дослідження, пов'язані з використанням клітинних автоматів для обробки зображень, оскільки можливість їх поєднання із нейромережовими технологіями надає можливість підвищити рівень надійності розпізнавання зорових об'єктів [3-5].

Клітинні автомати є дискретними динамічними системами, поведінка яких повністю визначається в термінах локальних залежностей. В значній мірі КА можуть бути застосовані і для опису великого класу безперервних динамічних систем, моделі яких, наприклад, визначаються рівняннями в часткових похідних. У цьому сенсі клітинні автомати можуть функціонувати у просторі, що представлений рівномірною сіткою, кожна комірка якої містить декілька бітів даних. При цьому функціонування КА реалізується у дискретному часі за заданим набором правил (наприклад, таблицею, за якою будь-яка клітина на кожному кроці обчислює свій новий стан за станами її близьких сусідів). Клітинні автомати дозволяють побудувати корисні моделі для прикладних досліджень в природничих науках і комбінаторній математиці, які можуть бути застосовані для вивчення еволюції фізичних систем різного функціонального призначення. Слід зазначити, що клітинні автомати утворюють загальну парадигму паралельних обчислень. Клітинні автомати, важливе застосування в якості концептуальних і практичних моделей просторово розподілених динамічних систем, для яких фізичні системи є певними прототипами. Основний напрям дослідження клітинних автоматів – алгоритмічна вирішувальність поставлених задач (включно питання визначення початкових станів, при яких клітинний автомат матиме змогу їх вирішення).

Клітинні автомати можуть, зокрема, використовуватися для різних завдань обробки зображень, серед яких: геометричні перетворення, фільтрування шуму, виявлення особливостей, виявлення контурів. При цьому недостатньо вивченою можна вважати проблему сегментації та розпізнавання зображень різних типів за допомогою клітинних автоматів. Використання алгоритмів клітинних автоматів для сегментації зображень та їх подальшого розпізнавання має такі вагомні переваги:

- простота впровадження;
- паралельна реалізація;

- довільна кількість сформованих класів при проведенні сегментації (можлива сегментація зображень із двома мітками та з кількома мітками);
- варіативність правил переходу станів (зазвичай для цих правил використовуються зміни значень інтенсивності пікселю, але в механізм оновлення можуть бути включені і інші функції зображення, такі як текстура або краї);
- можливість роботи з зображеннями будь-якого розміру (на обчислювальну складність процесу безпосередньо не впливає розмір зображення або кількість функцій зображення) [6].

В обробці зображень зазвичай використовуються двовимірні КА. Пікселі зображення являють собою комірки КА і вони оновлюють свій стан на основі станів сусідніх комірок (пікселів). Кілька станів комірок КА дозволяють обробляти зображення сірого масштабу або кольорові зображення. Визначення правил, що застосовуються до комірок, щоб відповісти на певний запит при обробці зображень, є нетривіальним завданням.

Розглянемо зв'язок двовимірного КА з зображенням. Зображення I у загальному вигляді може бути описане як двовимірна функція:

$$I = f(x, y), \tag{1}$$

де x і y – просторові координати.

Амплітуда f в будь-якій парі координат (x, y) визначає інтенсивність I для сірих значень цифрового зображення. Цифрове зображення I представляється єдиним двовимірним цілим масивом для зображення сірого масштабу та серії з трьох двовимірних масивів для кожної кольорової смуги. Для зображення, що задається двовимірним масивом $m \times n$ пікселів, використовується двовимірна сіткова модель КА. При цьому зображення розглядається як двовимірний КА, де кожна комірка відповідає пікселю у зображенні, а інтенсивність пікселя представлена станом цієї комірки. Значення кольорів пікселів оновлюються синхронно в дискретному часі.

Топологію сітки двовимірного КА з околком фон Неймана наведено на рис. 1.

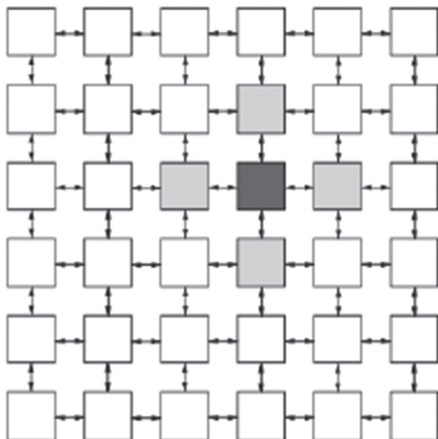


Рис. 1. Топологія сітки двовимірного КА з околком фон Неймана

Така топологія, де пов'язані лише найближчі сусіди аналізованої комірки (пікселя зображення), визначає окол фон Неймана з радіусом $r = 1$. Загалом, різні околи визначаються різними наборами зв'язків (рис. 2).

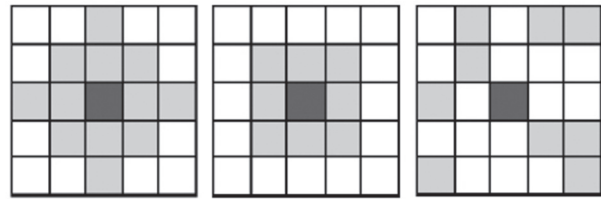


Рис. 2. Деякі типи околів двовимірного КА

На рис. 2 наведено (зліва направо) деякі типи найбільш вживаних околів двовимірного КА: фон Неймана (радіус 2), Мура (радіус 1) і випадковий.

Таким чином, двовимірний клітинний автомат складається з решітки квадратів $L = m \times n$, де кожен квадрат (клітина) представлений одним КА.

Змінні комірок у межах околів використовуються для оцінки нових значень оновленої комірки за допомогою функції (правила) переходу

Будь-яке правило переходу $\varphi(t)$ може бути записане у вигляді функції $\varphi(t)$, використовуючи стани $\sigma(t)$ всіх комірок в околах:

$$\sigma_{i,j}(t+1) = \varphi(\sigma_{i,j}(k, 1) \mid \sigma_{i,j}(k, 1) \in N), \tag{2}$$

де N – множина елементів околу (включно центральну комірку)

Правилом переходу $\varphi(t)$ може бути будь-яке поєднання арифметичних і логічних операцій, включаючи функції. Зазвичай правила переходу мають або форму таблиці переходу, що визначає залежність значення виходу від вхідних значень, або форму комп'ютерної програми.

Кількість правил стрімко зростає зі зростанням кількості сусідів та станів (табл. 1).

Таблиця 1
Залежність кількості правил двовимірного КА від кількості станів та сусідів

Кількість станів	Кількість сусідів	Кількість правил
2	2	16
2	3	256
2	5	4294967296

Навчання клітинних автоматів для завдання обробки зображень означає вибір оптимального набору правил серед усіх можливих правил, які найкраще відповідають конкретному завданню.

Отримати ефективні правила для досягнення бажаного результату дуже складно. Навіть для бінарних зображень кількість комбінацій всіх можливих правил є занадто велика. В цілому без перерахування всіх комбінацій оптимальний вибір правил не може бути

гарантований. Необхідним є пошук поєднань між правилами з використанням оптимізаційних методів. В роботі [7], наприклад, запропоновано використовувати генетичний алгоритм (ГА) для вирішення проблеми виявлення контурів зображень (edge detection) за допомогою КА.

Ініціалізація ГА здійснюється шляхом побудови випадковим чином пакетів правил, отриманих з околів моделі КА на всьому зображенні. Для визначення елементів контуру ГА здійснює пошук серед поточних пакетів вхідного зображення правило, що відповідає околу аналізованого пікселя, та модифікує центральний піксель відповідно до визначеного переходу.

Правило переходу КА представляється за допомогою конкатенації клітин стану околу комірки до оновлення та додаванням її майбутнього стану після оновлення. Це правило (шаблон) трансформується як лінійний формат хромосоми.

Функція фітнесу, яка використовується для виявлення контурів, має такий вигляд:

$$F = 1 - (Err/L \times H), \quad (3)$$

де L та H – ширина та висота зображення; Err – кількість пікселів, де $ImageED \neq ImageIdealED$ (ED – Edge Detection).

Функція Err обчислює кількість неспівпадаючих точок на отриманому та ідеальному зображеннях. Далі генерується нова сукупність з застосуванням характерних для ГА операторів селекції, кросовера та мутації. Цей процес триває до тих пір, поки не буде відбуватися подальше поліпшення функції фітнесу або до встановленої максимальної кількості ітерацій.

Таким чином, виявлення контурів зображень (або фрагментів зображень) можна інтерпретувати як еволюційний феномен множини КА, керований кращими пакетами правил КА, застосованих до зображення. Коли така система буде навчена на наборі зображень конкретного типу, вона зможе обробляти і інші зображення цього типу.

У випадку із зображеннями у відтінках сірого тренувальний процес для такої системи буде дещо ускладнений. В разі використання околів Мура, де комірки можуть мати 256 можливих інтенсивностей, для центрального пікселя зі значенням можливі 256^8 моделей сусідства. Якщо задане зображення сірого масштабу перетворюється на бінарне шляхом порогової декомпозиції, тоді результат попереднього розділу можна застосувати до отриманого зображення. Найбільш прийнятним методом такої декомпозиції слід вважати метод Оцу [8]. Завдяки властивості мінімізувати дисперсію в межах класу та максимізувати між дисперсією класу, метод Оцу здатний відображати максимальну кількість об'єктів, розташованих на зображенні, на порогове зображення.

В роботі [9] наведено результати навчання КА для виконання різних завдань обробки зображень. При цьому можливий набір правил зменшується з 256 до 51 на основі симетрії та рефлексії для центральних чорно-білих пікселів. Після цього проводиться навчання на основі скороченого набору правил.

Кожен зразок правила узгоджується із схемою сусідства пікселів зображення. Якщо він збігається, то колір центрального пікселя інвертується. Перед обробкою кожне правило перетворюється на 8-бітний рядковий шаблон і потім узгоджується з перетвореною схемою околу пікселя зображення. Для тренування системи мети використано метод послідовного плаваючого пошуку вперед (SFFS) з застосуванням цільової функції середньоквадратичної похибки між вхідним та цільовим зображеннями.

2. Особливості використання згорткових нейронних мереж для розпізнавання зображень

До найбільш розвинених та ефективних напрямків нейромережевої обробки статичних та динамічних зображень слід віднести застосування згорткових нейронних мереж (ЗНМ; англ. CNN – convolution neural networks). В загальному випадку мережі ЗНМ мають вхідний рівень (input), вихідний рівень (output) і кілька прихованих шарів.

Базову архітектуру ЗНМ, призначеної для класифікації та розпізнавання об'єктів (зокрема, зображень), наведено на рис. 3. В цій архітектурі присутні дві основні частини: виділення ознак об'єктів (feature extraction) та класифікація (classification). Виділення ознак є інструментом згортки, який розділяє та ідентифікує різні характеристики зображення для їх аналізу та подальшого використання на етапі класифікації.

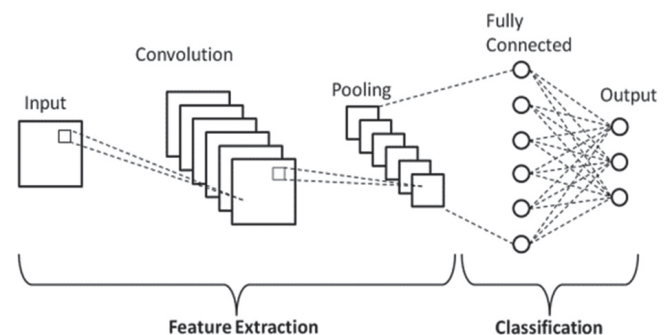


Рис. 3. Базова архітектура ЗНМ для класифікації зображень

Мережа виділення ознак ЗНМ складається з багатьох пар згорткових (convolution) та об'єднувальних (pooling) шарів. Ця мережа має на меті зменшити кількість ознак, присутніх у наборі даних. При цьому можуть створюватися нові функції, які узагальнюють існуючі функції, що містяться в оригінальному наборі функцій.

Згортковий шар є першим шаром, який використовується для виділення різних функцій із вхідних

зображень. У цьому шарі виконується математична операція згортки між вхідним зображенням і фільтром (ядром), що є квадратною матрицею $n \times n$ (n – непарне число). Пересування ядра по вхідному зображенню дозволяє сформувати скалярний добуток між ядром і частинами вхідного зображення. Під час обчислення нового значення обраного пікселя ядро згортки «прикладається» своїм центром до даного пікселя. Навколишні пікселі так само накриваються ядром. Далі вираховується сума, де складовими є помножені значення пікселів на значення комірки ядра, що накрила даний піксель. Сума ділиться на коефіцієнт нормування (div) або на суму всіх елементів ядра згортки. Отримане значення є новим значенням обраного пікселя. Якщо застосувати згортку до кожного пікселя зображення, то буде отримано результат, що залежить від обраного ядра згортки.

Приклад роботи ядра згорткового шару наведено на рис. 4.

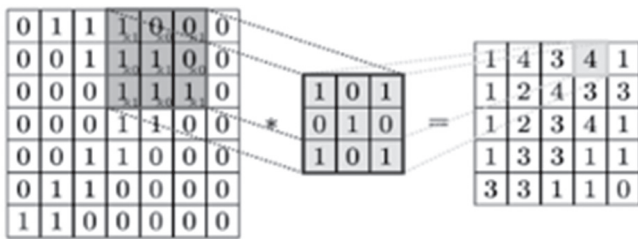


Рис. 4. Приклад роботи ядра згорткового шару

Результат операції згортки називається картою функцій, яка надає інформацію про зображення (наприклад, про кути та краї). Пізніше ця карта функцій передається на інші шари для вивчення особливостей вхідного зображення.

Ядро переміщується по всій області карти функцій і знаходить певні ознаки об'єктів. Наприклад, якщо мережу навчали на множині зображень осіб, то одне з ядер в процесі навчання може видавати найбільший сигнал в області очей, рота, інше ядро може виявляти інші ознаки. Якщо розмір ядра недостатній, то воно не зможе виділити потрібні ознаки, якщо ж занадто великий, то збільшується кількість зв'язків між нейронами. Зазвичай розмір ядра задають в діапазоні від 3×3 до 7×7 . Ядро реалізує систему поділюваних ваг, що є однією з головних особливостей ЗНМ. У звичайній багат шаровій нейронній мережі існує занадто багато зв'язків (синапсів) між нейронами, що суттєво уповільнює процес детектування. У ЗНМ узагальнення ваг дозволяє зменшувати число зв'язків і знаходити одну і ту саму ознаку по всій області зображення.

Рівень згортки в ЗНМ передає результат на наступний рівень (шари об'єднання) після застосування операції згортки у вхідних даних. Згорткові шари в ЗНМ забезпечують безперервність просторового відношення між пікселями зображення.

Основна мета субдискретизуючих шарів об'єднання (Pooling Layers) – зменшити розмір згорткової карти функцій, щоб скоротити витрати на обчислення. Це виконується шляхом зменшення зв'язків між шарами та незалежно працює на кожній карті об'єктів. Ці шари зазвичай розміщуються одразу після згорткових шарів, тобто виходить згорткових шарів є входами для субдискретизуючих шарів мережі. Операції об'єднання скорочують розміри карт характеристик за рахунок використання деяких функцій для узагальнення субрегіонів, наприклад, взяття загального або максимального значення. Шари субдискретизації націлені на поступове зменшення розмірності даних і, отже, додаткове скорочення кількості параметрів, а також складність процедури моделі і, таким чином, контроль над питанням перенавчання. Існує ряд методів об'єднання (максимальне об'єднання, середнє об'єднання, стохастичне об'єднання, спектральне об'єднання, об'єднання просторової піраміди, об'єднання по нормі $L2$ тощо).

Залежно від методу, який використовується, реалізуються кілька типів операцій об'єднання. Зокрема, операція Max Pooling полягає у визначенні найбільшого елемента з карти функцій. Ця операція розділяє вхідне зображення на набір прямокутників без перекриттів, і для кожної такої підобласті виводить її максимум. При цьому точне положення ознаки не так важливе, як її положення відносно інших ознак. Об'єднувальний шар сприяє поступовому скороченню просторового розміру представлення для зменшення кількості параметрів та об'єму обчислень у мережі, і відтак також для контролю перенавчання. Операція Average Pooling обчислює середнє значення елементів для зображення попередньо визначеного розміру. Загальна сума елементів у попередньо визначеному фрагменті визначається як об'єднання сум елементів. Рівень об'єднання зазвичай служить мостом між згортковим рівнем і рівнем класифікації. Ця модель ЗНМ узагальнює ознаки, витягнуті шаром згортки, і допомагає мережам розпізнавати функції незалежно. За допомогою цього також зменшуються обчислення в мережі. Найбільш поширеним є об'єднувальний шар шар з ядрами розміру 2×2 , що застосовуються з кроком 2. В такому шарі знижується дискретизація кожного зрізу глибини входу в 2 рази як за шириною, так і за висотою, відкидаючи 75 % збуджень. В архітектурі ЗНМ інколи вставляють об'єднувальний шар між послідовними згортковими шарами.

Об'єднувальний шар, як і згортковий також має карти. Одним з важливих завдань цього шару є зменшення розмірності карт попереднього шару. Якщо на попередній операції згортки вже були виявлені деякі ознаки, то для операції об'єднання докладне зображення вже не потрібно, і воно суттєво ущільнюється.

Слід відзначити, що фільтрація вже непотрібних деталей зменшує ризик виникнення перенавчання мережі. У процесі сканування ядром об'єднувального шару карти попереднього шару, скануюче ядро не перетинається на відміну від згорткового шару. Зазвичай, кожна карта має ядро розміром 2×2 , що дозволяє зменшити попередні карти згорткового шару в 2 рази. Вся карта ознак поділяється на осередки 2×2 елементи, з яких вибираються максимальні за значенням.

Повнозв'язний рівень Fully Connected (FC) використовує вихідні дані процесу згортання та передбачає класифікацію зображення на основі ознак, виділених на попередніх етапах. Цей рівень складається з ваг і зміщень разом із нейронами та використовується для з'єднання нейронів між двома різними шарами. Ці шари зазвичай розміщуються перед вихідним шаром і утворюють кілька останніх шарів архітектури ЗНМ.

У цьому випадку вхідне зображення з попередніх шарів вирівнюється та подається на повнозв'язний шар FC. Потім сплющений вектор проходить ще кілька шарів FC, де зазвичай виконуються операції математичних функцій. На цьому етапі починається процес класифікації. Доцільність з'єднання двох шарів полягає в тому, що два повністю з'єднані шари працюватимуть краще, ніж один з'єднаний шар.

Повнозв'язний шар зазвичай моделює складну нелінійну функцію, оптимізація якої сприяє підвищенню якості класифікації та розпізнавання об'єктів на вхідних зображеннях.

Нейрони кожної карти попереднього об'єднувального шару пов'язані з одним нейроном повнозв'язного шару. Таким чином, число нейронів повнозв'язного шару дорівнює числу карт об'єднувального шару, але при цьому тільки частина нейронів будь-якої з карт об'єднувального шару може бути пов'язана з першим нейроном прихованого шару. Частина, що залишилася, може бути пов'язана з другим нейроном прихованого шару.

Визначення кількості нейронів у шарі є важливим етапом побудови ЗНМ. На сьогодні не існує універсальних рекомендацій до вибору кількості прихованих шарів та кількості нейронів в них. Їх недостатня кількість не дозволяє мережі ефективно навчатися, а занадто велика кількість значно збільшує час навчання ЗНМ або призводить до її перенавчання (англ. *overfitting*). В разі виникнення перенавчання створена модель починає пояснювати лише приклади з навчальної вибірки, адаптуючись до них. Разом з тим вона не вчиться класифікувати приклади, які не залучалися до участі в навчанні, що зменшує здатність мережі до узагальнення можливих ситуацій класифікації зображень.

Найбільш поширеним емпіричним методом запобігання перенавчанню в ЗНМ є метод виключення (англ. *dropout*). Ідея метода полягає в тому, що

в процесі навчання випадково із загальної мережі багаторазово видаляється деяка підмережа, оновлення ваг виконується тільки в межах цієї підмережі. Нейрони потрапляють в підмережу з ймовірністю p , яка має назву коефіцієнт дропаута. На наступному етапі на даних тренується лише редукована мережа, після чого видалені вузли повторно вставляються до ЗНМ з первинними вагами (рис. 5).

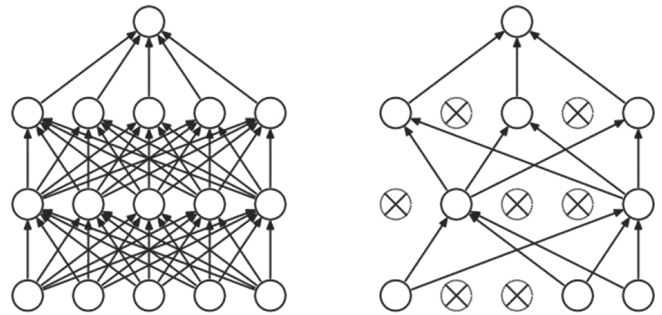


Рис. 5. Ілюстрація до виключення вузлів ЗНМ за методом Dropout

Таким чином, процедура виключення дозволяє уникати необхідності тренування всіх вузлів на всіх тренувальних даних, що знижує прояв ефекту перенавчання. Вона також значно покращує швидкість тренування та дозволяє здійснювати практичне поєднання моделей для ЗНМ.

Однією з найважливіших задач побудови ЗНМ є вибір функцій активації. Вони використовуються для вивчення та апроксимації будь-якого виду безперервних і складних зв'язків між змінними мережі.

Зазвичай, в об'єднувальному шарі застосовується функція активації *ReLU* (англ. *Rectified Linear Units* – зрізані лінійні вузли). Цей шар застосовує ненасичувальну передавальну функцію $f(x) = \max(0, x)$. Він посилює нелінійні властивості функції ухвалення рішення і мережі в цілому, не зачіпаючи рецептивних полів згорткового шару. Для посилення нелінійності застосовуються й інші функції, наприклад, насичувальні гіперболічний тангенс $f(x) = \tanh(x)$, $f(x) = |\tanh(x)|$, сигмоїдна функція $f(x) = (1 + e^{-x})^{-1}$ та *Softmax* (нормалізована сигмоїда). Кожна з цих функцій має певні рекомендації до використання: зрізаному лінійному вузлові *ReLU* часто віддають перевагу перед іншими функціями, оскільки він тренує нейронну мережу доволі швидко без втрати точності узагальнення; сигмоїдна функція використовується, зазвичай, для бінарної класифікації, а *Softmax* – для багатокласової класифікації.

Крім стандартних архітектур ЗНМ останнім часом набули поширення нові способи побудови згорткових шарів, щоб підвищити ефективність навчання. Ці архітектури надають загальні архітектурні рекомендації для практиків машинного навчання, які необхідно адаптувати для вирішення різноманітних проблем комп'ютерного зору. Ці архітектури можна використовувати як екстрактори багатих функцій для

класифікації та розпізнавання зображень. До перспективних модифікованих архітектур ЗНМ слід, зокрема, віднести комбіновані архітектури з використанням можливостей КА.

3. Інтеграція клітинних автоматів в нейромережеву модель розпізнавання зображень

Розглянемо ознаки певної еквівалентності клітинних автоматів та згорткових нейромережевих моделей.

Визначимо КА як динамічну систему з M можливими станами, яка оновлює своє значення на основі свого поточного значення та значення D інших клітин (безпосередніх сусідів у квадратній сітці). Вочевидь існує M^D можливих вхідних рядків σ для функції КА, які ми індивідуально позначаємо як σ . Клітинний автомат реалізує оператор $G(\sigma)$, який повністю задає список правил переходу $\sigma \rightarrow m, m \in 0, 1, \dots, M - 1$. При цьому існує M^D можливих унікальних $G(\sigma)$, кожен з яких реалізує свій набір правил. Наприклад, для гри в життя $M = 2, D = 9$, а $G(\sigma)$ — булева функція, яка відображає кожен з 512 можливих 9-бітних вхідних рядків в один біт. Визначальною особливістю КА є локальність динамічного оновлення правила, що гарантує обмеженість області правил, де розмір D обумовлює її верхню межу.

Визначимо конволюційну нейронну мережу як функцію, яка приймає на вхід багатоканальне зображення, до якого застосовується серія локальних згорток через «ядро», що навчається. Те саме ядро застосовується до всіх пікселів зображення, і кожен згортковий шар поєднує інформацію в межах фіксованого локального радіусу кожного пікселя вхідного зображення [8]. Багато стандартних конволюційних архітектур включають «об'єднуючі» шари, які знижують дискретизацію попереднього шару і тим самим консолідують локальну інформацію на все більших просторових масштабах; однак, всі ЗНМ, що розглядаються в цій статті, не включають кроки зниження дискретизації і, таким чином, зберігають повну розмірність вхідного зображення.

Розглянемо КА як рекурентні мережі mlpconv . Основна аналогія між клітинними автоматами і традиційними згортковими нейронними мережами виникає через локальність динаміки і одночасне темпоральне оновлення всіх просторових точок. Оскільки нейронні мережі в принципі можуть діяти як універсальні апроксиматори функцій, досить складна архітектура нейронної мережі може бути використана для повної апроксимації кожного правила $\sigma \rightarrow m$, що становить функцію КА $G(\sigma)$. Потім цей оператор для одного околу може бути реалізований як згортковий оператор у складі ЗНМ, що дозволяє синхронно застосовувати його до всіх околів пікселів на вхідному зображенні.

Таким чином, представлення КА за допомогою ЗНМ вимагає двох кроків: вилучення ознак для

ідентифікації кожного з M^D вхідних випадків, що описують кожен окіл, а потім асоціацію кожного околу з відповідним вихідним пікселем. Можна показати, як представити будь-який КА за допомогою одного згорткового шару, а потім і 1×1 згорткових шарів. Прийнятні ваги можна знайти аналітично, використовуючи аналіз самої КА, а не шляхом алгоритмічного навчання на вхідних даних. Таким чином, один із можливих варіантів визначає мережу, яка однозначно зіставляє кожен з M^D входів σ із шаблоном, у той час як інший варіант розглядає шари мережі як рівні деревоподібного пошуку, який ітеративно звужує кожен з входів σ до бажаного виходу m .

Ключовим аспектом такого підходу є використання лише одного неоднорічного згорткового шару, який є першим прихованим шаром в мережі. Поле цих конволюційних нейронів еквівалентно околу D КА. Усі наступні шари складаються з 1×1 згорток, які не консолідують жодної додаткової інформації про сусідів. Використання згорток 1×1 для реалізації логіки таблиці правил КА має підвищити ефективність мережі та зменшити обчислювальні витрати. Більше того, для такої реалізації КА мережа не потребує об'єднувальних шарів, що робить мережу еквівалентною установці невеликого конволюційного багатошарового перцептрон (« mlpconv ») на КА [8].

Розглянемо структуру та функції цього різновиду ЗНМ. Багатошаровий перцепційний згортковий шар (mlpconv) є комбінацією низькорівневих функцій за допомогою узагальненої лінійної моделі побудови ЗНМ. Втім реальні дані часто є нелінійними, тому доцільним є використання комбінації $\text{Conv} + \text{MLP}$ (згортка + багатошаровий перцептрон) замість операції лінійної згортки, щоб поліпшити здатність мережі отримувати абстрактні функції та здійснювати узагальнення. Mlpconv можна розглядати як мініатюрну багатошарову згорткову мережу, що міститься в локальному рецептивному полі кожної згортки. Каскадний процес інтеграції картки крос-функцій може дозволити мережі вивчати складні функції. Крос-канальний параметризований шар сприйняття еквівалентний згортці 1×1 .

На рис. 6 та 7 наведено для порівняння загальні схеми побудови лінійного згорткового шару ЗНМ та шару mlpconv .

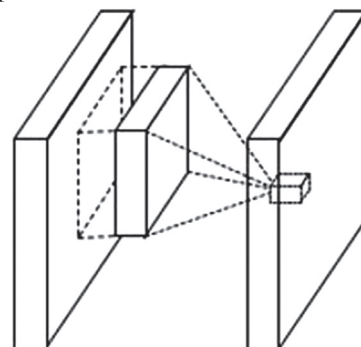


Рис. 6. Загальна структура лінійного згорткового шару

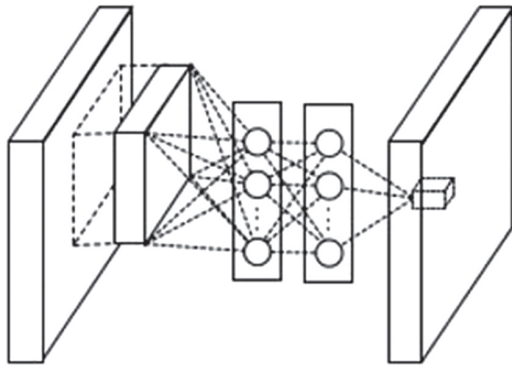


Рис. 7. Загальна структура шару mlpconv

Причина вибору MLP полягає в тому, що ця загальна функція апроксимації сумісна з базовою архітектурою ЗНМ, може бути поглиблена сама по собі (повторне використання функцій) і може бути навчена з використанням стандартного алгоритму зворотнього поширення помилки

Такий підхід до гібридного використання можливостей КА та конволюційного багат шарового перцептрон можна порівняти з використанням базових ЗНМ для розпаралелювання простих операцій двійкової арифметики, але він відрізняється від спроб використання менш поширених типів мереж з блоками сигма-пі, в яких кожен елемент у вхідному векторі розширюється попарною комбінацією помножених входів, а окремі вхідні біти можуть блокувати один одного [9].

4. Архітектура та принцип дії пропонованої системи розпізнавання зображень

Насамперед розглянемо алгоритмічну послідовність представлення довільного КА за допомогою ЗНМ.

Покажемо, як стандартний багат шаровий mlpconv перцептрон з активацією ReLU здатний представити довільний КА з M станами КА з кінцевою глибиною та обмеженою кількістю нейронів [8].

Розглянемо алгоритм зіставлення таблиці правил КА з мережею ЗНМ, що містить шар mlpconv.

Довільний двовимірний КА з M станами має спочатку бути перетворений у бінарне представлення. Для заданого $L \times L$ зображення сформуємо двійковий стек $L \times L \times M$ зображення для активації системи розпізнавання з використанням наступних дій, що реалізуються багат шаровим згортковим перцептроном mlpconv:

- згорнути вхідний шар за допомогою M різних згортки 1×1 фільтрів з одиничними вагами та зміщеннями, заданими як: $1, 0, -1, \dots, -(M-1)$; застосувати далі активацію ReLU;

- згорнути отримане зображення за допомогою згортки M 1×1 фільтрів з нульовими зміщеннями. Кожен з перших $(M-1)$ згорткових фільтрів перевіряє послідовну пару $[1, -b, 0, \dots, 0], [0, 1, -b, 0, \dots, 0]$,

$[0, 0, 1, -b, 0, \dots, 0], \dots, [0, \dots, 0, 1, -b]$, де b – позитивна константа $b = M/(M-1)$; останній згортковий фільтр $[0, \dots, 0, 1]$; застосувати знову активацію ReLU.

Таким чином, після виконання зазначених дій сформовано масив $L \times L \times (M-1)$, що відповідає одноразовому кодуванню значення кожного пікселя зображення в сітці $L \times L$. Далі шаблон зіставляє кожен із M^D можливих входів зі своїм відповідним вихідним значенням. Відзначимо, що кількість активних стани в КА має бути задані заздалегідь і дорівнювати λM^D (λ – параметр Ленгтона) [17]. Далі оцінювання кількості шаблонів для відповідності (тобто кількості параметри) можна зменшити в λ раз. тому що тільки активні правила, які дають ненульовий результат, мають бути враховані.

Після формування масиву кодування значень пікселів необхідно:

- побудувати блок M^D , що містить $S \times S \times (M-1)$ згортки фільтрів, де S відповідає розміру околу двовимірного КА ($S = 3$ для стандартного КА з околом Мура). Кожен із фільтрів M^D просто відповідає зображенню можливого вхідного стану, що дорівнюють одиниці для кожного ненульового елементу зображення і великі від’ємні значення (більше $D(M-1)$) для кожного нульового елементу. Для випадків, коли $M > 2$, глибина кожного згорткового ядра дозволяє здійснити точне зіставлення різних ненульових значень.

- призначити зміщення кожному з M^D фільтрів на основі таблиці правил КА. Для входів $S \times S \times (M-1)$, які відповідають ненульовим значенням q , призначити зміщення $(q-1) - (L-1)$, де L – кількість ненульових вузлів в околі L $D(M-1)$. Це гарантує, що тільки точні збіги з правилом сформуєть додатні значення для елементів згортки мережі. Для вхідних даних, які мають відобразитися на нуль, слід призначити будь-який зсув $L = D(M-1)$;

- застосувати функцію активації ReLU.

Наведений алгоритм обумовлює пропонований варіант архітектури mlpconv нейромережевої реалізації розпізнавання зображень, навченої на бінарних КА, яка може бути відображена такою схемою:

```
INPUT → [CONV3x3(1LAYER, 100FILTERS)] →
[CONV1x1(11LAYER, 100FILTERS)] → RELU →
[CONV1x1(LAYER, 1FILTER)] → RELU →
FC(SOFTMAX) → OUTPUT.
```

Мережа складається з базової архітектури mlpconv, що відповідає одному згортковому шару 3×3 , за яким слідує змінна кількість згорткових шарів 1×1 [28]. Об’єднувальні шари при цьому не використовуються, а параметри шарів 3×3 і 1×1 навчаються разом. Останній шар здійснює зважене підсумовування, яке генерує передбачене значення наступного стану ділянки сітки КА. Включення останнього шару

з класифікатором softmax прискорює навчання на бінарних КА, зменшуючи залежність збіжності від початкових ваг нейронів.

Мережа навчається за допомогою одного з оптимізаторів (зокрема, оптимізатора Адама з функцією втрат L_2 , з гіперпараметрами (швидкість навчання, початкові ваги і т.д.), оптимізованими за допомогою пошуку через сітку КА.

5. Розробка та тестування програмного додатку

Згідно зі схемою запропонованої нейромережевої технології розпізнавання зображень було здійснено її програмну реалізацію з використанням мови програмування Python і бібліотек TensorFlow та Keras, що дозволяють швидко побудувати і натренувати нейронну мережу. Програма передбачає можливість її інсталяції на мобільні пристрої.

Навчання програми оптимізатором Adam та її подальше тестування відбувалися з використанням навчальних вибірок, завантажених з датасету для розпізнавання продуктів Products-10K (dataset <https://products-10k.github.io/>).

Зокрема, одна з навчальних вибірок (Fruits) складалася з фотографій фруктів, розподілених за певними категоріями: яблуко (apple), банан (banana), апельсин (orange) тощо. Вона була поділена на множини train і valid (250 та 70 об'єктів відповідно).

Оскільки генерування нових навчальних даних не потребує великих обчислювальних витрат, для кожного етапу настроювання гіперпараметрів генерувався новий набір даних для валідації. Крім того, валідація проводилася з використанням випадково вибраних наборів правил КА, щоб гарантувати, що гіперпараметри мережі не були налаштовані на конкретні правила. Під час навчання того ж набору правил КА генерувався другий валідаційний набір даних, розмір якого становив 20% від розміру навчальних даних. Для контролю того, як вибір оптимізатора та функції втрат впливає на навчання, були створені та навчені реплікативні ансамблі мереж mlrconv, які мали ту саму форму мережі (12 шарів по 100 нейронів кожен), але іншу функцію втрат та оптимізатор, для яких були обрані різні оптимальні гіперпараметри. (таблиця 2).

Таблиця 2

Точність навченої мережі (mlrconv + CA) в залежності від типу оптимізатору

Оптимізатор	Точність (accuracy), %	Помилка (loss)
Adam	96,74	0.1521
SGD	82,65	0.2787
Rmsprop	94,35	0.1698

Дослідним шляхом було визначено оптимальну кількість епох навчання. Було проведено навчання

мережі при різній кількості епох на тестовій вибірці (з оптимізатором Adam). Результати експерименту відображено у таблиці 3. Оптимальна кількість епох навчання склала 40.

Таблиця 3

Точність навченої мережі (mlrconv + CA) в залежності від кількості епох навчання

Кількість епох навчання	20	30	40	50
Точність (accuracy), %	91.55	92.68	96.74	95.12

Навчання успішно завершилося для всіх розглянутих наборів правил КА. Відзначимо, що використання архітектури згорткової мережі спрощує навчання повної таблиці правил. У результаті всі навчені мережі були працездатними, проте тривалість і динаміка навчання суттєво відрізнялися залежно від набору правил КА. Оскільки КА має тільки M^D можливих пар «вхід-вихід», а не необмежений простір входів, то навчання зупинялось, коли модель обробляла всі входи. Слід відзначити, що мережа mlrconv+CA може кодувати попереднє очікування для довільного входу, якщо вона була навчена на КА певного типу.

Зауважимо, що реальні мережі, навчені оптимізатором Adam, можуть знайти інші варіанти застосування mlrconv для реалізації довільних КА.

Для розробки мобільного додатку, що реалізує інсталяцію та застосування розглянутої технології розпізнавання зображень, було використано бібліотеку react-navigation.

Для отримання доступу до камери мобільного пристрою (в експериментах використовувалася Apple iPhone 10) була використана бібліотека exocosmetica. Після отримання зображення з камери, перед тим, як відправити його на вхід до моделі нейромережі, передбачена обрізка зображення згідно з рамками, для чого використано бібліотеку expo-image-manipulator.

Під час практичного тестування розробленого мобільного додатку інколи виникали проблеми якості розпізнавання зображень, які обумовлені такими причинами:

- відсоток правильної класифікації знижується, коли освітлення є недостатнім, а на зображенні присутні тінюві плями;

- відсоток правильної класифікації знижується, коли в полі зору камери присутні однотипні об'єкти з близьких класів на недостатній відстані один від одного.

В умовах, коли мережа не була навчена на необхідних прикладах та не може точно класифікувати деякий об'єкт, то додаток його ідентифікує як найбільш подібний об'єкт, або оголошує про неможливість розпізнавання.

На рис. 8 наведено приклад розпізнавання зображення об'єкту з класу «Fruits».

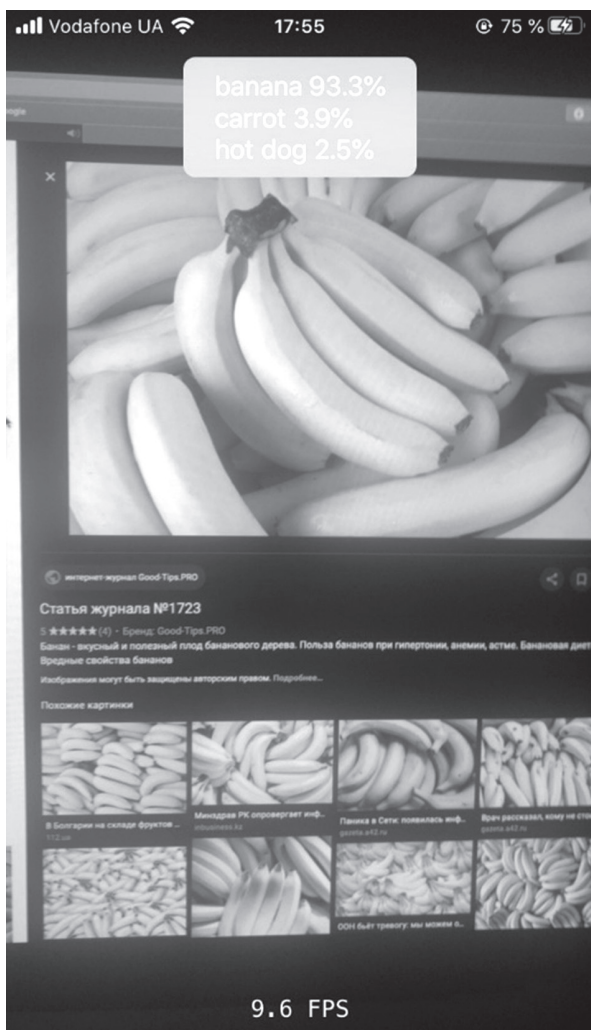


Рис. 8. Приклад розпізнавання зображення об'єкту з класу «Fruits».

Оскільки додаток орієнтовано на користувачів з обмеженими можливостями зорового сприйняття, то в ньому передбачено формування відповідного голосового повідомлення за результатами розпізнавання.

Висновки

До важливих завдань розпізнавання статичних та динамічних зображень слід, зокрема, віднести завдання динамічного пошуку та детекції об'єкта у часі та просторі з використанням наявних відеокамер

Перспективним напрямом вирішення цих завдань є застосування методів та засобів очислювального інтелекту. У статті наведено результати розробки та програмної реалізації нейромережевої технології розпізнавання зображень з використанням згорткових нейронних мереж та клітинних автоматів.

Побудову комбінованої нейронної мережі здійснено з частковим використанням бібліотеки Keras для мови програмування Python. Для формування наборів зображень в роботі використовувався датасету для розпізнавання продуктів Products-10K. Для

розробки мобільного додатка, що реалізує встановлення та застосування розглянутої технології розпізнавання зображень, була використана бібліотека react-navigation.

Результати проведеного тестування запропонованої системи розпізнавання зображень побутових об'єктів підтверджують її працездатність (зокрема, на тестовій вибірці середня точність прогнозу дорівнює 94.6%, що є цілком задовільним результатом для згорткових нейронних мереж з урахуванням малої кількості елементів вибірки). Розроблена система розпізнавання зображень з використанням мобільних пристроїв є гнучкою та розширюваною, що робить можливим її подальше удосконалення у напрямку підвищення точності розпізнавання та розширення функцій. Для підвищення точності системи доцільно збільшити навчальну вибірку, розширити діапазон представлених у ній зображень та додати у створену систему додаткові дані про особливості її застосування (наприклад, скорегувати набір гіперпараметрів мережі відповідно до побажань користувачів з обмеженим зоровим сприйняттям).

Список літератури:

- [1] Schmidhuber, J. (2015). "Deep Learning in Neural Networks: An Overview". *Neural Networks*. 61: 85–117. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003. PMID 25462637. S2CID 11715509.
- [2] Torlai, G., Mazzola, G., Carrasquilla, J. et al. Neural-network quantum state tomography. *Nature Phys* 14, 447–450 (2018). <https://doi.org/10.1038/s41567-018-0048-5>
- [3] Жежерун О.П., Калітовський Б.В. (2019). Використання клітинних автоматів для вирішення задач фільтрації шумів та виявлення контурів зображень. *Наукові записки НаУКМА. Комп'ютерні науки*. Том 2. 66 – 72.
- [4] Shukla, A. P. (2016). Training Cellular Automata for Image Edge Detection. *Romanian Journal of Information Science and Technology*, 19, 4, 338–359.
- [5] A review of cellular automata models for crowd evacuation / Li Yang et al. *Physica A*: (2019). *Statistical Mechanics and its Applications*. Vol. 256. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0378437119303528> (accessed 05.05.2021).
- [6] Diosan, Laura & Andreica, Anca & Enescu, Alina. (2017). The Use of Simple Cellular Automata in Image Processing. *Studia Universitatis Babeş-Bolyai Informatica*. 62. 5-14. 10.24193/subbi.2017.1.01.
- [7] Rosin, Paul & Sun, Xianfang. (2011). Cellular Automata as a Tool for Image Processing. 10.1142/9789814343008_0012.
- [8] William Gilpin. Cellular automata as convolutional neural networks. *Physical Review E*, 100(3), Sep 2019. ISSN 2470-0053. doi: 10.1103/physreve.100.032402. URL <http://dx.doi.org/10.1103/PhysRevE.100.032402>.
- [9] Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin (2020). Growing neural cellular automata. doi: 10.23915/distill.00023. URL <https://distill.pub/2020/growing-ca>.

Надійшла до редколегії 16.02.2022