УДК 004.412:004.8

# ЗАСТОСУВАННЯ АЛГОРИТМІВ КОМП'ЮТЕРНОГО ЗОРУ ДЛЯ РОЗРОБКИ ПРОГРАМНОЇ СИСТЕМИ ПОШУКУ ЗОБРАЖЕНЬ ЗА НАЯВНИМ ТЕКСТОМ НА ANDROID

Поцілуйко В.А.[1]; Прокоп Ю.В., к.і.н., доц.[2]
[1,2]Національний університет «Одеська політехніка»
[1,2]Україна, Одеса;
[1]potsiluiko.8864821@stud.op.edu.ua; [2]prokop.y.v@op.edu.ua

***Анотація.*** *Виконано порівняння алгоритмів комп'ютерного зору з боку балансу між точністю, адаптованістю, швидкодією, здатністю до розгортання та економічною раціональністю для розробки програмної системи пошуку зображень за наявним текстом на пристроях Android. На основі аналізу, визначено отимальний шлях виконання задачі та особливості його впровадження.*
***Ключові слова:*** *розпізнавання тексту; машинне навчання; великі мовні моделі.*

***Вступна частина.*** Середньостатистичний користувач сучасного смартфона зберігає у пам'яті свого пристрою тисячі зображень [1]. Пошук у такому об'ємному масиві графічних файлів методом ручного перебору елементів множини є вкрай трудомістким завданням. Для варіантів використання, де необхідно знайти, скан-копію документа чи квитанції, слайд презентації або інше зображення, яке містить текст, доцільно ввести можливість пошуку зображень за наявним на них текстом [2]. Для реалізації цього функціоналу пропонується використати алгоритми комп'ютерного зору, що здатні локалізовувати та розпізнавати текст на зображеннях [3].

Актуальність теми пояснюється тим, що на цей момент не існує рішення, яке дозволяє виконувати автоматичну індексацію та пошук зображень за наявним на них текстом на пристроях під керуванням операційної системи Android автономно, без застосування послуг хмарних сервісів. Сервіси на основі хмарних технологій потребують стабільного інтернет-з'єднання та вимагають прямої оплати при перевищенні ліміту на загальний розмір оброблених файлів, створюють значні ризики безпеки та втрати конфіденційності. Таким чином, система, яка виконувала б описане вище завдання без залучення послуг хмарних сервісів могла б мати значну практичну новизну та цінність для кінцевого користувача.

***Метою роботи*** є підвищення часової ефективності та рівня захищеності виконання команди з пошуку зображень за наявним на них текстом для користувачів пристроїв під керуванням операційної системи Android.

***Основна частина.*** Ідея про те, що команди над текстом, такі як, зокрема, пошук, мають бути доступними незалежно від того як зберігається текст: у вигляді символьного документа чи растрового зображення, була вперше висловлена Джефом Раскіним, видатним фахівцем із розробки комп'ютерних інтерфейсів користувача, який стояв за розробкою одного із найперших комерційно успішних персональних комп'ютерів, Macintosh.

Новатор запропонував ідею про те, що команди призначені для даних одного типу, можна використовувати для даних іншого типу, за умови, що доступний трансформатор для перетворення даних з початкового типу, в такий, якого потребує команда. Нехай команда C має наступний вигляд:

$$C: A \to R,$$

де A – вхідні дані для команди; R – результат виконання команди.

Щоб застосувати команду C до типу даних B, потрібно, щоб програмне середовище комп'ютера містило трансформатор:

$$\tau: B \to A,$$

де B – дані типу, які не підтримує команда C; A – дані типу, який підтримує команда C.

Тоді, застосування команди C до екземпляра даних $b \in B$ є можливим і виглядає наступним чином:

$$(C \circ \tau)(b) = C\big(\tau(b)\big) \in R.$$

У контексті розроблюваної системи таким трансформатором є спеціалізовані методи, які дозволяють розв'язати задачу з розпізнавання тексту на зображеннях. На цей момент ринок пропонує три основних види таких методів: традиційний, заснований на машинному навчанні (МН) та заснований на використанні великих мовних моделей (LLM).

Традиційні методи розпізнавання тексту на зображеннях були вперше розроблені в епоху, коли апаратне забезпечення навіть найпередовіших комп'ютерів мало вкрай обмежену обчислювальну потужність, окрім того, підходи, керовані даними, були значно менш розвинуті [4]. Їхня робота ґрунтується на використанні таких підходів, як зіставлення зі зразками та розпізнавання на основі правил. Обидва підходи вимагають попереднього розбиття вхідного зображення на сегменти, кожен із яких представляє окремий символ. Через це точність розпізнавання традиційних систем оптичного розпізнавання тексту часто різко падає, коли текст на зображенні має хаотичний макет, символи розміщені надто близько один до одного, використовуються нестандартні шрифти або текст є рукописним: подібні недоліки викликають помилки при сегментації, які, своєю чергою, пропагують помилки розпізнавання.

Зіставлення зі зразками передбачає порівняння виділених сегментів вхідного зображення з наперед підготовленими зразками. Бібліотека таких зразків зіставляє кожен символ алфавіту зі зразками, які його можуть представляти у графічному вигляді [5]. Алфавіт, тобто набір літер, цифр та символів, які здатна розпізнати система, визначається жорстко вручну, відповідно до специфікації вимог конкретної системи.

Розпізнавання на основі правил передбачає застосування вручну визначених інструкцій, а не пошук візуальних збігів. Наприклад, система може містити таку логіку: якщо сегмент містить одну петлю та вертикальну риску зліва від петлі, то це літера «P».

Для того щоб адресувати недоліки традиційних систем розпізнавання тексту на зображеннях, індустрією інформаційних технологій було розроблено методи розпізнавання тексту на зображеннях засновані на МН. Вони не потребують попереднього розбиття вхідного зображення на окремі символи. Натомість процес розпочинається з виявлення регіонів тексту на зображенні та їх розбиття на окремі лінії [6]. Кожна лінія обробляється згортковою нейронною мережею (ЗНМ) [7, с. 78].

ЗНМ мають численну кількість окремих згорткових шарів. Вони обробляють зображення, застосовуючи до його послідовних регіонів фільтри. Кожен фільтр, або ядро – це числова матриця, яка визначається під час процесу навчання моделі, щоб виділяти певну графічну ознаку [5]. На початкових етапах застосовуються фільтри, які виявляють прості ознаки (краї фігур, лінії). На наступних етапах фільтри виявляють вже складніші ознаки, такі як окремі частини символів або навіть форми, схожі на цілі символи. Між етапами згортання застосовується процедура об'єднання шарів. Вона зменшує просторову розмірність вхідної інформації для наступного згорткового шару, зберігаючи при цьому найрелевантніші дані, виявлені під час попереднього згортання. Отримані результати збираються у так звану «карту ознак», яка є багатовимірними числовим масивом, кожне число у якому вказує на наяв-

ність у сегменті *ij* вхідних даних ознаки, яку виявляє фільтр *k*. Ця карта ознак зберігає просторову структуру тексту (порядок символів у тексті, їхню орієнтацію зліва направо чи навпаки), тому її довгу сторону можна розглядати як чергу вузьких вертикальних відрізків.

Кожен вертикальний відрізок може містити частини декількох розпізнаваних символів, але це не є проблемою. Замість того щоб жорстко приймати один відрізок як такий, що має відповідати одному символу, модель обробляє всю послідовність відрізків засобами рекурентної мережі (такої, де нейрони наступного шару можуть мати зворотні зв'язки з нейронами попередніх) [8, с. 872].

Останнім етапом є декодування результатів рекурентної мережі. На кожному кроці мережа визначає розподіл ймовірностей для всіх символів алфавіту, які містилися в навчальних даних, та спеціального «порожнього» символу, який вказує на відсутність символу на відрізку [8, с. 875]. Можлива ситуація, коли один і той самий символ розпізнано на декількох відрізках, тому результат може містити численні повтори. Метод CTC (Connectionalist Temporal Classification) перетворює вивід моделі, який є результатом приблизних прогнозів, у цілісну лінію тексту, шляхом видалення порожніх символів та повторів літер.

За останнє десятиріччя технологія великих мовних моделей (LLM) стала однією з найбільш відомих та впливових у сфері МН та штучного інтелекту. Їхня поява суттєво підвищила якість результатів для завдань автоматизованого комп'ютерами аналізу, інформації, що має формат природної людської мови, порівняно з попередніми розробками. Великі мовні моделі базуються на трансформаторній архітектурі, яка обробляє вхідні дані у декілька кроків. На початковому етапі текст розбивається на токени, тобто на слова, або їхні окремі фрагменти, які часто зустрічаються [9]. Кожен токен перетворюється у векторне представлення. Для цього вектор спочатку ініціалізується випадковими числовими значеннями. Під час тренування, модель обробляє масив тренувальних даних (текстів), роблячи припущення про те, яке слово буде йти після попереднього. Коли модель робить помилку, значення коригуються таким чином, щоб слова які з'являються у схожих контекстах ставали ближчими за векторною відстанню. Разом вектори для кожного виділеного токену моделі формують векторний простір. Натренована модель здатна визначати залежності в тексті шляхом впровадження механізму самоуваги, який передбачає використання цього простору для контекстуалізації та пріоритезації окремих одиниць тексту [10].

Інтеграція великих мовних моделей у ланцюжок інструментів розпізнавання тексту може дозволити автоматично виправляти некоректно розпізнані слова на основі смислового наповнення тексту та коригувати загальний текст, враховуючи загальні знання, якими володіє модель. Незважаючи на це, інтеграція LLM у процес розпізнавання супроводжується серйозними труднощами. Ключовим фактором є їх колосальні вимоги до обчислювальних ресурсів. Актуальні версії LLM налічують мільярди параметрів і, відповідно, вимагають для власної роботи спеціалізованих графічних процесорів або серверних кластерів, що робить їх розгортання на пристроях користувача практично нереальним.

Порівняння показало, що традиційні системи працюють найшвидше та мають найнижчі вимоги до обчислювальних ресурсів, однак можливість їх застосування значно обмежена високими вимогами до якості вхідних даних. Великі мовні моделі навпаки демонструють виняткову здатність до виправлення помилок та аналізу тексту, але надмірна ресурсомісткість робить їх непридатними у рамках цього проєкту. Найбільш збалансованим виявився підхід заснований на методах МН, оскільки він долає недоліки традиційних систем і водночас зберігає здатність до локального розгортання.

Оскільки обробка кожного зображення за допомогою алгоритмів МН на сучасному мобільному обладнанні хоча і є можливою, але може бути тривалою, вводиться механізм фонової індексації. Це передбачає, що зображення оброблюватимуться автоматично у міру їх збереження у спільній пам'яті пристрою, незалежно від того, чи користувач відкрив у цей момент вікно програми на передньому плані. Завдяки введенню процесу індексації, передбачається, що у момент коли користувач подасть запит до системи на пошук зображення за ключовим словом, результат можна буде визначити з мінімальною затримкою.

***Висновки.*** У роботі розглядалося застосування алгоритмів комп'ютерного зору для розробки програмної системи пошуку зображень за наявним текстом на Android. Проведене теоретичне дослідження показало, що оптимальним способом розпізнавання тексту на зображеннях для локального запуску на базі пристроїів під керуванням операційної системи Android є застосування методів машинного навчання. Експериментальне дослідження швидкодії цих методів допомогло виявити потребу у введенні механізму фонової обробки даних.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


[1] Mays, S. (2024, 21 березня). *How many pictures are there.* smays.com. https://www.smays.com/2024/03/how-many-pictures/

[2] Sanad.ai. (2024, 8 жовтня). *The benefits of OCR: Optimizing document scanning and recognition*. https://sanad.ai/the-benefits-of-ocr-optimizing-document-scanning-and-recognition/

[3] Milvus. (2025). *What is optical character recognition (OCR) in computer vision?* https://milvus.io/ai-quick-reference/what-is-optical-character-recognition-ocr-in-computer-vision

[4] Tripathi, P. (2025, 11 квітня). *A journey through history: The evolution of OCR technology.* Docsumo. https://www.docsumo.com/blog/optical-character-recognition-history

[5] Konovalchuk, N. (2024, 28 травня). *OCR algorithms: Types, use cases and best solutions.* Itransition. https://www.itransition.com/computer-vision/ocr-algorithm.

[6] Achache, N. (2022, 8 квітня). *OCR in natural images SOTA in text detection and recognition.* Theodo Data & AI. https://data-ai.theodo.com/en/technical-blog/ocr-text-detection-recognition

[7] Chaitra, Y. L., Roopa, M. J., Gopalakrishna, M. T., & Swetha, M. D. (2023). Text detection and recognition from the scene images using RCNN and EasyOCR. *IOT With Smart Systems*, 75–85.

[8] Khalkar, R. G., Dikhit, A. S., Goel, A., & Gupta, M. (2021). Handwritten text recognition using deep learning (CNN & RNN). *International Advanced Research Journal in Science, Engineering and Technology, 8(6),* 870–881.

[9] Trott, S. (2024, 2 травня). *Tokenization in large language models, explained.* The Counterfactual. https://seantrott.substack.com/p/tokenization-in-large-language-models

[10] Vaswani, A. (2023). Attention is all you need. *31st International Conference on Neural Information Processing Systems.*



Potsiluiko V.A., Prokop Y.V.


**Application of computer vision algorithms for the development of a software system for searching images by recognized text on Android**


***Abstract.*** *A comparison of computer vision algorithms has been carried out in terms of the balance between accuracy, adaptability, speed, deployability, and cost-effectiveness for the development of a software system for image search by available text on Android devices. Based on the analysis, the most optimal way of accomplishing the task and the features of its implementation have been determined.*



***Keywords:*** *text recognition; machine learning; large language models.*