КОМП'ЮТЕРНІ НАУКИ

УДК 004.8

РОЗРОБКА ПРОТОТИПУ СИСТЕМИ ОПТИЧНОГО РОЗПІЗНАВАННЯ ТЕКСТУ ДЛЯ ЗОБРАЖЕНЬ НИЗЬКОЇ ЯКОСТІ

М. Баранов[1], С. Іванов[1], Я. Соколовський[2], Ю. Юрченко[1]

[1] *Львівський національний університет імені Івана Франка, вул. Університетська, 1, м. Львів, 79000, e-mail: mykola.baranov@lnu.edu.ua, serhii.ivanov@lnu.edu.ua, yuliana.yurchenko@lnu.edu.ua,*
[2] *Національний університет "Львівська політехніка", вул. С. Бандери, 12, м. Львів, e-mail: yaroslav.i.sokolovskyi@lpnu.ua*

Розглянуто проблеми оптичного розпізнавання символів, коли вхідні зображення низької якості з високим рівнем цифрового шуму, розмиття, спотворень цифрової обробки тощо. Ці чинники, а також тип документа, шрифту тощо суттєво впливають на кількість помилок при розпізнаванні тексту.

Для емулювання спотворень зображень тексту було створено набір зображень шляхом накладання тексту на білий фон; до згенерованих зображень застосовані функції зміни розмірів, функції накладання шуму, оператори розмиття та повороту. Усі застосовані спотворення мають випадковий рівень інтесивності. Створення даних у такий спосіб дало змогу отримати розмічені зображення тексту низької якості у будь-якій кількості. З використанням бібліотеки Keras побудовано архітектуру CRNN, для якої визначена функція втрат CTC. Для збільшення точності результатів OCR ми запропонували етапи попередньої обробки даних: алгоритм горизонтального вирівнювання зображення тексту та алгоритм розрізання зображення багаторядкового тексту на декілька зображень однорядкового тексту. З використанням цього підходу задача розпізнавання багаторядкового тексту зводиться до перевикористання моделі, що працює лише з одним рядком символів. Для порівняння символьних послідовностей міток і передбачень моделі визначено метрику похибки, що дорівнює відношенню відстані Левенштейна між міткою та передбаченням моделі до довжини мітки. Значення цієї метрики при розпізнаванні тексту тестової частини зображень становить близько 0.02. Проведене тестування для сучасних OCR моделей Tesseract OCR v 5.0.0. (alpha) та Google Cloud Vision API доводить, що описані алгоритми попередньої обробки зображень і побудована архітектура нейронної мережі є ефективними для розпізнавання тексту зображень низької якості.

Ключові слова: оптичне розпізнавання символів, цифровий шум, розмиття зображень, спотворення зображень, згорткові нейронні мережі, рекурентні нейронні мережі.

1. Вступ

Оптичне розпізнавання символів (Optical Character Recognition, OCR) – це технологія, яка дає змогу перетворювати різні типи документів, такі як засканковані документи, PDF-файли або фото з цифрової камери, в електронний формат, що легше розпізнається і редагується комп'ютерами та програмами. Системи OCR отримали широке застосування у різних сферах сучасності: зчитування даних бланків та анкет, візитних карток, паспортних даних; автоматичне розпізнавання номерного знака; створення цифрових версій друкованих і рукописних документів;

технологія для допомоги сліпим і людям зі слабким зором. Для використання найпопулярніших систем розпізнавання тексту рекомендується використовувати скануюче обладнання, яке немобільне, часто недоступне або дороге, а також непридатне до зчитування документа нестандартного формату (номерні знаки, білборди, інформаційні вивіски і т.д.). У повсякденному житті люди частіше використовують камери мобільних пристроїв та цифрові фотоапарати. Порівняно зі сканерами, оптична схема камери більш складна і вносить більше спотворень внаслідок аберацій, відблисків і віддзеркалень усередині оптичної системи. Використання фотосенсорів і аналогової електроніки пристроями для реєстрації зображень неминуче призводить до появи спотворень зображень – цифрового шуму, який відображається на зображення у вигляді накладеної маски з пікселів випадкового кольору та яскравості. Цифровий шум посилюється в умовах недостатньої освітленості, внаслідок руху документа або камери під час експозиції. На відміну від сканерів при зйомці камерою сам документ розташований у довільній площині стосовно площини сфокусованого зображення, що може призводити до проєктивного спотворення зображення документа. Ще одне джерело спотворень – алгоритми стиснення зображень, особливо характерні для кадрів відеопотоку.

Сьогодні існує безліч систем для розпізнавання тексту, однак лише незначна частина здатна ефективно працювати з зображеннями низької якості. Ця недосконалість і непристосованість сучасних OCR систем до роботи з зображеннями низької якості призводить до неохідності розробки моделі оптичного розпізнавання тексту, яка працюватиме ефективно.

2. Сучасні підходи вирішення задачі OCR

Дослідження оптичного розпізнавання тексту досягли значного прогресу за останні роки. Серед традиційних методів відомі численні висхідні підходи, де окремі символи спочатку визначаються за допомогою ковзаючого вікна [11, 12], зв'язаних компонент [7], голосування за Хафом [13] тощо. Після цього виявлені символи інтегруються в слова за допомогою динамічного програмування, пошуку у лексиконі [11] тощо. Залежно від складності сформульованої цілі та специфіки даних задача OCR може виконуватися шляхом розв'язання незалежних підзадач: виявлення окремих смужок тексту; просторове перетворення смужок тексту в прямі лінії тексту; поділ ліній тексту на окремі символи; розпізнавання символів; семантичне доповнення або виправлення тексту.

Інші роботи використовують наскрізні підходи, де текст розпізнається безпосередньо з цілого вхідного зображення, та доводять їх ефективність [9]. Наприклад, у [1] пропонується прогнозувати вектори міток з вхідних зображень. У [6] задачу розпізнавання тексту зіставлено з задачею 90000-класової класифікації, де кожен клас визначає англійське слово. У [5] побудовано CNN зі структурованим вихідним рівнем для розпізнавання тексту без сталого лексикону та без відомої довжини слова. Деякі роботи моделюють проблему OCR як проблему розпізнавання послідовності, а саме послідовності символів. У праці [10] автори витягують послідовне подання зображення, яке є послідовністю дескрипторів HOG [2], і прогнозують відповідну послідовність символів за допомогою RNN. У [8] вперше комбінує CNN та RNN, щоб отримати послідовні візуальні ознаки заданого текстового зображення, а потім безпосередньо передати їх у декодер CTC.

3. Створення набору даних

Ми штучно створили 11000 зображень шляхом накладання тексту з книги "Війна і мир" Льва Толстого (перекладеної англійською мовою) на білий фон. Характеристики цих зображень наведені у табл. 1.

Наступним завданням було створення набору зображень тексту низької якості з існуючих чітких зображень тексту. Для цього розглянемо чинники зниження якості зображень, що діють в реальному житті, та їхні математичні апроксимації.

Таблиця 1

Характеристики згенерованих чітких зображень тексту

Розмір зображення	256x256
Формат зображення	PNG
Шрифт тексту	Arial
Розмір тексту	22px – 26px
Відстань між рядками	1.5px

1. Поворот зображення. Нехай W та H – ширина та висота зображення, відповідно; α – кут повороту; c_x та c_y – координати центру повороту зображення. Класично поворот зображення досягається застосуванням матриці перетворення (1)

$$M = \begin{pmatrix} \cos\alpha & \sin\alpha & (1-\cos\alpha)\cdot c_x - \sin\alpha \ \cdot c_y \\ -\sin\alpha & \cos\alpha & \sin\alpha \cdot c_x + (1-\cos\alpha)\cdot c_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{1}$$

Для створення датасету ми використовували змінену матрицю перетворення (3) й обчислювали ширину W' та висоту H' результуючого зображення за формулою (2). Перевагою такого підходу є уникнення втрати частини зображення внаслідок обрізання

$$\begin{aligned} W' &= H \cdot |\sin\alpha| + W \cdot \cos\alpha, \\ H' &= H \cdot \cos\alpha + W \cdot |\sin\alpha|. \end{aligned} \tag{2}$$

$$M = \begin{pmatrix} \cos\alpha & \sin\alpha & \frac{W'}{2} - \cos\alpha \cdot c_x - \sin\alpha \cdot c_y \\ -\sin\alpha & \cos\alpha & \frac{H'}{2} + \sin\alpha \cdot c_x + -\cos\alpha \cdot c_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{3}$$

2. Цифровий шум зображення. *Гаусівський шум* – шум, густина розподілу ймовірностей якого дорівнює густині p нормального розподілу випадкової величини z

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}, \tag{4}$$

де z відображає сірий рівень; μ – середнє значення; σ – стандартне відхилення.

Шум солі та перцю – випадково виникаючі чорні та білі пікселі.

Спекл-шум – шум, який виникає внаслідок впливу навколишнього середовища на фотоматрицю під час отримання зображення. Такий шум залежить від багатьох

Таблиця 2

Параметри зниження якості зображень

Поворот	$\alpha \in [-5; 5]$ градусів	
Шум	Гаусівський шум	$\mu \in [0.5; 0.9], \sigma \in [0.05; 0.09]$
	Шум солі та перцю	к-сть пікселів $\in [0; 0.02]$
	Спекл-шум	$\mu = 0, \sigma = 0.001$ або $\mu = 1.0, \sigma = 0$
Розмиття	Гаусівське розмиття	радіус ядра $\in \{0, 1\}$
	Розмиття по рамці	радіус ядра $\in \{0, 1\}$
	Фільтр мінімуму (максимуму)	розмір ядра $\in \{1, 3\}$
Зміна розмірів	білінійна або бікубічна інтерполяція	

обставин, неможливо однозначно виразити його математичними формулами. Ми використали шум, значення якого для кожного пікселя дорівнює добутку значення кольору цього пікселя та випадкової величини, густина розподілу якої дорівнює густині p нормального розподілу.

3. Розмиття зображення. *Гаусівське розмиття* – тип фільтрів розмиття зображення, що використовує функцію Гауса для обчислення перетворення, яке застосовують до кожного пікселя на зображенні. Формула функції Гауса для двовимірного випадку

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{5}$$

де σ – стандартне відхилення. Формула (5) породжує поверхню, контурами якої є концентричні кола з гаусівським розподілом від центральної точки. Значення цього розподілу використовують для побудови ядра згортки, яке застосовують до вихідного зображення.

Розмиття по рамці – лінійний фільтр просторової області, в якому кожен піксель результуючого зображення має значення, що дорівнює середньому значенню сусідніх пікселів у вхідному зображенні. Більш формально, фільтр розміру 3×3 може бути поданий у вигляді матриці $\frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$.

Фільтр мінімуму (максимуму). Фільтр мінімуму (6) (максимуму (7)) визначає значення пікселя $f(x, y)$ результуючого зображення як найменше (найбільше) серед значень пікселів оригінального зображення g, що належать області S_{xy}

$$f(x, y) = \min_{(s,t)\in S_{xy}} \{g(s, t)\} \tag{6}$$

$$f(x, y) = \max_{(s,t)\in S_{xy}} \{g(s, t)\}. \tag{7}$$

4. Спотворення зображень шляхом цифрової обробки. Зміна розмірів зображення. *Білінійна інтерполяція* замінює кожен відсутній піксель на середнє зважене значень найближчих пікселів.

Бікубічна інтерполяція менше розмиває краї та спотворює зображення, ніж білінійна інтерполяція, але має більшу обчислювальну складність. Бікубічна інтер-

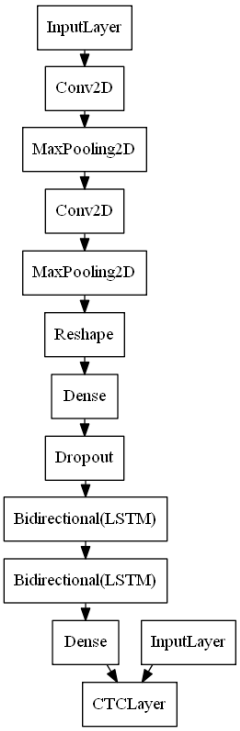

Рис. 1. Використана архітектура OCR моделі

поляція обчислюється шляхом підгонки кубічних поліномів до значень пікселів, що оточують відсутній піксель.

Для створення набору даних параметри описаних вище алгоритмів зниження якості зображень були обрані випадково з діапазонів значень, які наведені у табл. 2.

4. Архітектура нейронної мережі

Ми використовуємо CRNN для задачі OCR. Така модель охоплює переваги DCNN та RNN [8]:

- для навчання може використовувати мітки, які є послідовностями (наприклад, слова), не вимагаючи додаткової анотації (наприклад, літер);
- має властивість DCNN вивчати інформативні подання безпосередньо з оригінальних зображень, не потребуючи ручного відбору інформативних признаків і препроцесингу;
- має властивість RNN запам'ятовувати послідовності;
- не обмежена щодо довжини об'єктів, які вивчають та мають вигляд послідовностей, потребує тільки нормалізації висоти цих об'єктів;
- досягає високої точності для задач OCR;
- має набагато менше параметрів, ніж звичайна DCNN модель, використовує менше місця для зберігання.

Особливістю побуваної моделі також є використання функції втрат CTC, яку здебільшого застосовують:

- для sequence-to-sequence (Seq2seq) моделей;

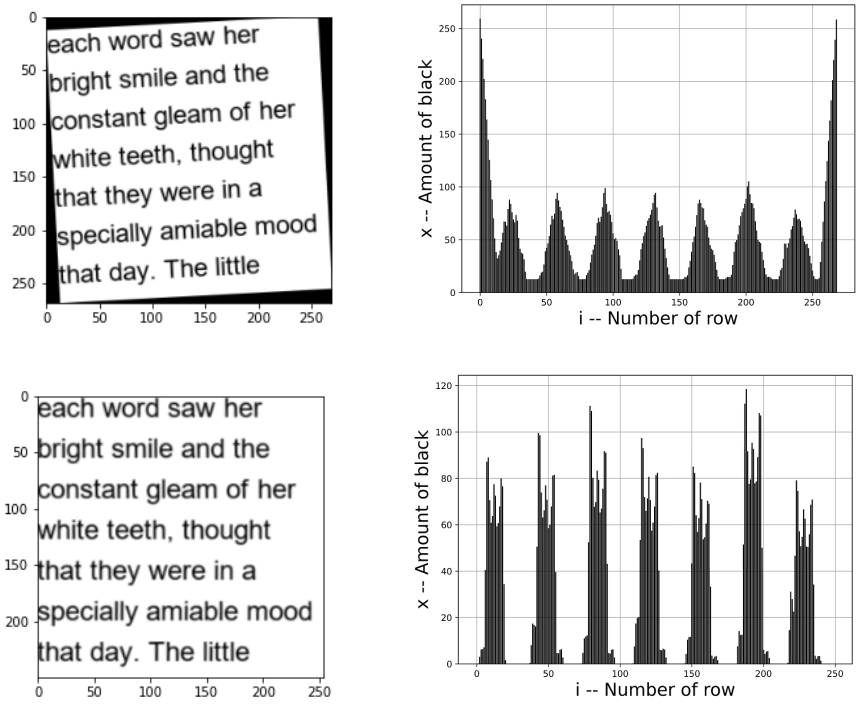

Рис. 2. Зображення тексту у відтінках сірого та графік ознаки $\mathbf{x}$ початкового (*а*) та вирівняного (*б*) зображень

– якщо мітки впорядковані, але немає взаємно однозначної відповідності між вхідними даними та міткою.

Архітектура побудованої моделі зображена на рис. 1. З'ясувалось, що вона не забезпечує необхідної якості для зображень багаторядкового тексту і потребує застосування додаткових алгоритмів; один з можливих пропонується у наступному розділі.

5. Попередня обробка зображень

Зображення приводимо до вигляду одноканального у відтінках сірого. Далі до зображення застосовуємо розроблений алгоритм розрізання зображення багаторядкового тексту на декілька зображень однорядкового тексту. Застосування цього алгоритму дає змогу значно поліпшити кінцеву точність моделі. Опишемо його детальніше.

Розглянемо зображення тексту у відтінках сірого. Значення пікселя $p_{i,j} \in [0; 256)$, $i = \overline{1,n}$, $j = \overline{1,m}$; n, m – кількості рядків і стовпців зображення, відповідно. Нехай $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, де

$$x_i = \sum_{j=1}^{m}(1 - \frac{p_{i,j}}{255}), \quad i = \overline{1,n} \tag{8}$$

чисельно виражає кількість чорного кольору в кожному рядку.

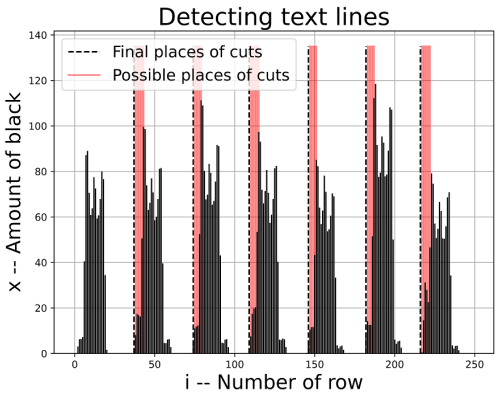

Рис. 3. Графік ознаки **x** і результатів виявлення текстових рядків зображення

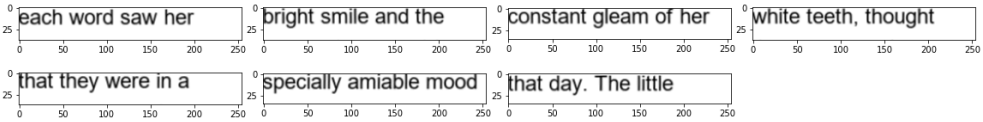

Рис. 4. Результат роботи алгоритму розрізання зображення багаторядкового тексту на декілька зображень однорядкового тексту

Для горизонтального вирівнювання зображення повернемо його на кут ϕ, для якого у результуючого зображення коефіцієнт варіації ознаки **x** максимальний.

Тепер визначимо місця можливих розрізань зображення після рядків i, для яких

– $\{x_{i-\alpha}, ..., x_{i-2}, x_{i-1}\}$ утворюють область низької кількості чорного кольору

$$\overline{x_\alpha} = \frac{1}{\alpha} \sum_{j=i-\alpha}^{i-1} x_j \quad < \quad \overline{x} - \frac{1}{2}\sigma \tag{9}$$

– $\{x_i, x_{i+1}, ..., x_{i+\beta}\}$ утворюють область високого значення приросту кількості чорного кольору

$$d_\beta = \frac{\sum\limits_{j=i}^{i+\beta-1} x_j - \sum\limits_{j=i-\beta}^{i-1} x_j}{\beta} \quad > \quad \delta \cdot \sigma. \tag{10}$$

Тут α, β та δ – регульовані гіперпараметри. Ці гіперпараметри були визначенні як $\alpha = 0.04$, $\beta = 0.02$, $\delta = 0.3$.

На рис. 3 місця можливих розрізань позначені червоним кольором. З кожної групи таких індексів вибирається перший (лівий край на рис. 3 позначений пунктиром).

Останніми кроками обробки зображень є приведення їх до однакового розміру 32×256 і нормалізація значень пікселів до діапазону $[0; 1]$.

6. Результати роботи OCR моделі та метрики

Дані для тренування, валідації та тестування поділені у відношенні 9 : 1 : 1 зображень, відповідно. Тренування відбувалося протягом 80 епох (рис. 5).

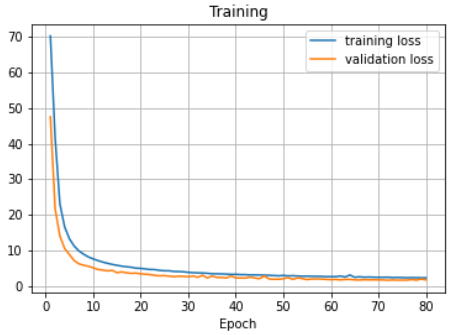

Рис. 5. Значення функції втрат при тренуванні та валідації

Як метрику порівняння взято відношення відстані Левенштейна [14] між передбаченням моделі та міткою до довжини мітки. Для порівняння використовували Tesseract OCR v5.0.0. (alpha) [3] та Google Cloud Vision API [4] (цей тільки для 50 зображень відповідно до квоти сервісу). Значення метрики порівняння наведені в табл. 3.

Таблиця 3

Середні значення метрики порівняння на тестовому наборі даних для нашої моделі, Tesseract та Google Cloud Vision API

API	**Наша модель**	Tesseract	Cloud Vision
50 тестових зображень	**0.022326**	0.742395	0.190283
1000 тестових зображень	**0.019783**	0.726493	–

Зазначимо також, що обчислення були виконані на процесорі Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz.

7. Висновки

Досліджено підходи машинного навчання, які дають змогу розпізнавати текст зображень низької якості. Були отримані такі результати.

1. Проведено огляд, вивчення та аналіз наявних OCR систем, сучасних алгоритмів і підходів до вирішення задач оптичного розпізнавання тексту.
2. Математично сформульовано задачі пониження якості зображень для досягнення спотворень, які виникають у реальному житті. Створено набір даних зображень низької якості.

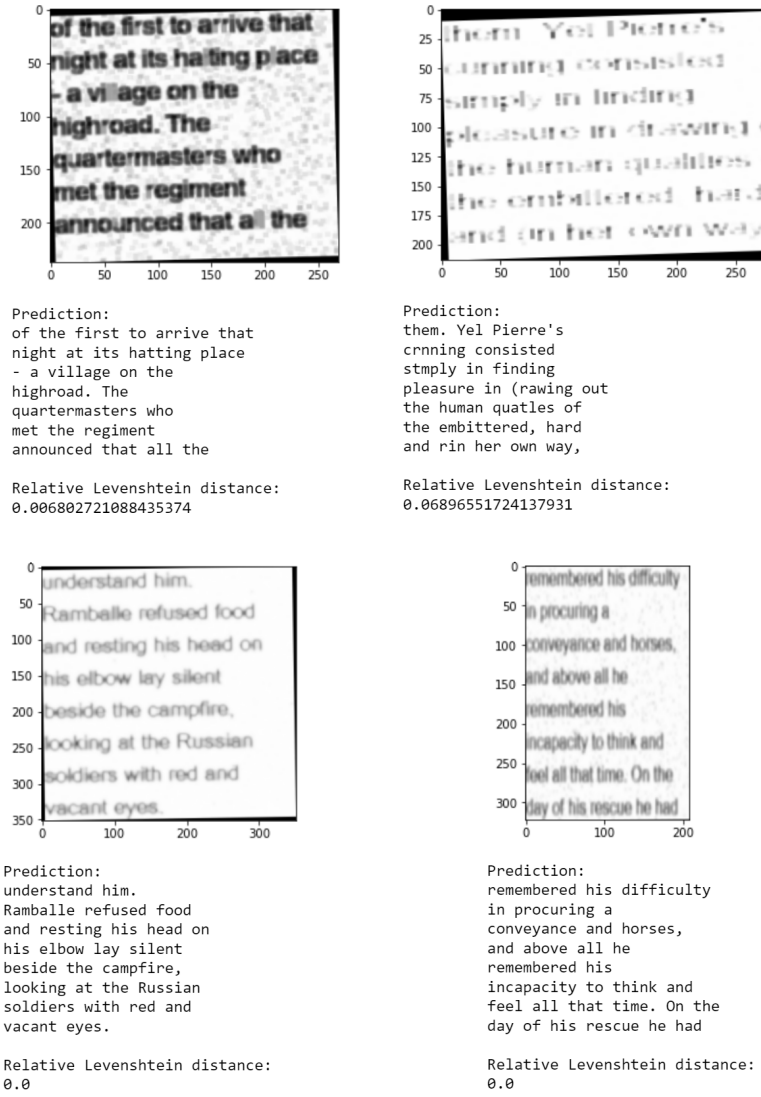

Рис. 6. Результати передбачення нашої моделі: приклади розпізнавання тексту для зразків тестового набору даних і чисельні значення метрики порівняння

3. Реалізовано попередню обробку зображень для забезпечення максимальної ефективності нейронної мережі на створеному наборі даних.

4. Побудовано нейронну мережу, архітектура якої пристосована до вирішення задач оптичного розпізнавання тексту низької якості.

5. Означено порівняльну метрику для оцінки ефективності роботи OCR систем та досліджено цю метрику для результатів створеної моделі, Tesseract OCR v5.0.0. (alpha) та Google Cloud Vision API. Наша модель виявилася ефективною для розпізнавання тексту зображень низької якості та мала кращий результат.

Список використаної літератури

1. Almazán J. Word spotting and recognition with embedded attributes / J. Almazán, A. Gordo, A. Fornés, E. Valveny // IEEE Trans. Pattern Anal. Mach. Intell. – 2014. – Vol. 36 (12). – P. 2552–2566.
2. Dalal N. Histograms of oriented gradients for human detection / N. Dalal, B. Triggs // In CVPR. – 2005.
3. GitHub repository for Tesseract OCR v5.0.0 (alpha). – Available from: https://github.com/UB-Mannheim/tesseract/wiki
4. Google Cloud Vision API. – Available from: https://cloud.google.com/vision
5. Jaderberg M., Deep structured output learning for unconstrained text recognition / M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman // In ICLR. – 2015.
6. Jaderberg M. Reading text in the wild with convolutional neural networks / M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman // Int. J. Comput. Vision. – 2015.
7. Neumann L. Real-time scene text localization and recognition / L. Neumann, J. Matas // In CVPR. – 2012.
8. Shi B. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition / B. Shi, X. Bai, C. Yao
9. Smith R. End-to-End Interpretation of the French Street Name Signs Dataset / R. Smith, C. Gu, D.-S. Lee, H. Hu, R. Unnikrishnan, Ju. Ibarz, S. Arnoud, S. Lin
10. Su B. Accurate scene text recognition based on recurrent neural network / B. Su, S. Lu // In ACCV. – 2014.
11. Wang K. End-to-end scene text recognition / K. Wang, B. Babenko, S. Belongie // In ICCV. – 2011.
12. Wang K. Word spotting in the wild / K. Wang, S. Belongie // In ECCV. – 2010.
13. Yao C. Strokelets: A learned multi-scale representation for scene text recognition / C. Yao, X. Bai, B. Shi, W. Liu // In CVPR. – 2014.
14. Левенштейн В.И. Двоичные коды с исправлением выпадений, вставок и замещений символов / В.И. Левенштейн // Докл. АН СССР. – 1965. – 163, 4. – С. 845–848.

Стаття: надійшла до редколегії 15.09.2021
доопрацьована 10.11.2021
прийнята до друку 24.11.2021

DEVELOPMENT OF A PROTOTYPE OF AN OPTICAL TEXT RECOGNITION SYSTEM FOR LOW-QUALITY IMAGES

M. Baranov[1], S. Ivanov[1], Ya. Sokolovskyy[2], Yu. Yurchenko[1]

[1]*Ivan Franko National University of Lviv,*
Universytetska str., 1, Lviv, 79000,
e-mail: mykola.baranov@lnu.edu.ua, serhii.ivanov@lnu.edu.ua,
yuliana.yurchenko@lnu.edu.ua,
[2]*National University "Lviv Polytechnic",*
Lviv, Stepan Bandera str., 12, e-mail: yaroslav.i.sokolovskyi@lpnu.ua

This paper considers the problems that arise while recognizing symbols for low-quality images with a high level of digital noise, blurring, distortion of digital processing.

In this work a new dataset proposed, which consists of synthetic images with overlaying text on a white background. A lot of distortions were obtained via applying resize functions,

noise functions, blurring and rotation operators. Applied transformations have random intensity uniformly distributed. It simulates the image distortions that occur in real life. Creating data in this way allows getting labeled low-quality images of text in any quantity. We used Keras library to build a CRNN and instantiate a custom "endpoint layer" for implementing CTC loss. A novel pipeline of data preprocessing is suggested for data preprocessing in order to increase the accuracy of OCR results: an algorithm for horizontal alignment of the text image and an algorithm for cutting a multi-line text image into several single-line text images. It allows us to reuse model that is suitable for single-line text and achieve similar accuracy score on the multi-line text images.

We defined an error metric to compare character sequences of labels and model predictions as the ratio of the Levenstein distance between the label and the model prediction to the label length. That score expose how often model mismatch single character. The value 0.02 of this metric was obtained for our model while recognizing the text of the test dataset. Testing was also performed for state-of-art OCR models – Tesseract OCR v 5.0.0. (alpha) and Google Cloud Vision APIs. The results prove that the built neural network architecture and the described image preprocessing algorithms are effective for recognizing text of low-quality images.

Key words: optical character recognition, image noise, image blur, image distortion, convolutional neural network, recurrent neural network.