

УДК 004.42

РОЗРОБКА СИСТЕМИ АВТОМАТИЧНОГО ПОШУКУ ТА РОЗПІЗНАВАННЯ ТЕКСТОВИХ СИМВОЛІВ ІЗ ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

Пелех Г.П.,

Булатецький В.В. (ORCID: 0000-0002-9883-4550).

Волинський національний університет імені Лесі Українки, Луцьк, Україна

DEVELOPMENT OF A SYSTEM FOR AUTOMATIC DETECTION AND RECOGNITION OF TEXTUAL SYMBOLS USING NEURAL NETWORKS

Pelekh H.P., Bulatetskyi V.V.

Lesya Ukrainka Volyn National University, Lutsk, Ukraine

Abstract. The aim of the study is to develop software for the automatic detection and recognition of symbols in images using neural networks. The system employs two neural networks: the first analyzes the full-size image, identifies symbols, and determines their coordinates in the QUAD format. The second network processes the extracted image fragments and classifies the symbols based on the training dataset. Text recognition is performed character by character, making the CRAFT model an optimal choice for symbol detection. The symbol recognition model is built using a classical classifier architecture that includes two convolutional layers with pooling and four fully connected layers. The results demonstrate that the combination of the two models ensures a recognition accuracy of 83.87% on the test dataset. Testing confirmed that the main factors influencing accuracy are the quality of the training dataset, learning parameters, and the method of symbol localization. The practical significance of the results lies in the potential application of the developed system for automating text data processing in various fields, such as document management, computer vision, and sign recognition.

Keywords: Symbol recognition, neural networks, CRAFT, machine learning, classification.

Вступ.

Розробка систем розпізнавання тексту є одним із ключових напрямів у сфері комп'ютерного зору. Їхня актуальність зумовлена потребою автоматизації процесів обробки текстової інформації з різноманітних джерел, зокрема документів, вивісок, цифрових зображень тощо. Використання нейронних мереж, таких як CRAFT (Character Region Awareness for Text Detection), дозволяє досягти високої точності в локалізації та розпізнаванні символів навіть у складних умовах.

У цій роботі представлено підхід до створення програмного забезпечення для пошуку та розпізнавання символів із застосуванням двох нейронних мереж. Особливу увагу приділено вибору інструментів, організації навчання моделей та аналізу точності отриманих результатів.

Результати дослідження та їхнє обговорення.

Інструменти та технології для розробки проєкту на основі машинного навчання. Для розробки проєкту з виявлення та розпізнавання символів було обрано мову програмування Python, яка є найпоширенішою для створення програмного забезпечення з використанням машинного навчання та нейронних мереж. Мова Python відрізняється легкістю вивчення та використання через простоту синтаксису та відсутність специфічних особливостей, що ускладнюють її використання у порівнянні з іншими мовами. На цій мові написана велика кількість бібліотек, що полегшують роботу із нейронними мережами та алгоритмами обробки візуальних даних, у нашому випадку – зображень. Ще однією з переваг використання мови програмування Python є віртуальні середовища, що дозволяють уникати конфліктів залежностей між встановленими бібліотеками та вести паралельну розробку проєктів, що використовують різні версії тієї ж бібліотеки.

Для реалізації проєкту були обрані бібліотеки PyTorch та OpenCV, які забезпечують необхідний функціонал для побудови нейронних мереж і роботи із зображеннями. PyTorch – це бібліотека машинного навчання з відкритим кодом, створена на основі мови програмування Python [1]. Вона пропонує динамічний і гнучкий підхід до побудови нейронних мереж і широко використовується в дослідницьких і виробничих середовищах. OpenCV – це бібліотека функцій програмування, в основному для комп'ютерного зору в реальному часі [2]. Вона містить значну кількість інструментів для роботи із зображеннями.

У якості середовища розробки використали редактор коду Visual Studio Code. Серед його переваг – можливість підключення різноманітних плагінів та розширень, розумне доповнення коду на основі контексту, додаткові інструменти налагодження, підсвічування помилок, аналіз в залежності від обраної мови програмування тощо. VSCode поєднує в собі простоту редактора коду з тим, що потрібно розробникам для основного циклу редагування-складання-налагодження.

Концептуальна структура проєкту для виявлення та розпізнавання символів. В якості головних компонент розробки були використані дві нейронні мережі. Перша приймає на вхід повнорозмірне зображення із трьома кольоровими каналами та повертає координати всіх знайдених символів у форматі QUAD (координати кутів чотирикутника, описаного навколо символу). Використовуючи ці координати, можна виділити з початкового зображення трансформовані фрагменти із зображеннями знайдених символів та передати їх на обробку другій нейронній мережі, яка виконує розпізнавання.

Друга нейронна мережа отримує на вхід фрагмент початкового зображення, що відповідає окремому символу та повертає на виході закодоване значення, що відповідає одному із символів набору даних, на якому було навчено мережу.

Модель пошуку тексту на зображенні на основі CRAFT. Розпізнавання тексту відбувається посимвольно, тому в якості основи для моделі пошуку було обрано модель CRAFT. Модель CRAFT була запропонована у 2019 році. CRAFT – це згортова нейронна мережа, яка повертає на виході оцінки областей і оцінки спорідненості. Оцінка областей використовується для локалізації окремих символів на зображенні, а оцінка спорідненості використовується для групування кожного символу в один екземпляр. Щоб компенсувати відсутність анотацій на рівні символів, CRAFT використовує слабо-контрольоване машинне навчання, яке оцінює еталонні значення на рівні символів із існуючих реальних наборів даних на рівні слів. Використовуючи розпізнавання регіону на рівні символів, CRAFT легко знаходить тексти різних форм (рис.1). [3]

CRAFT використовує архітектуру повністю згорткової нейронної мережі на основі VGG-16 [4] із нормалізацією пакетів як основної бази. Модель оснащена пропускними з'єднаннями в частині декодування, що нагадує архітектуру U-net [5], завдяки здатності агрегувати низькорівневі ознаки. Остаточний вихід представлений у вигляді двох каналів, які виступають картами оцінок: регіональної оцінки та оцінки спорідненості. Схема архітектури мережі зображена на рис. 2. [3]

Для кожного навчального зображення ми генеруємо мітки ground truth для оцінки регіону та спорідненості за допомогою межових рамок на рівні символів. Оцінка регіону представляє ймовірність того, що заданий піксель є центром символу. Оцінка спорідненості відображає ймовірність центру простору між суміжними символами.

На відміну від бінарної карти сегментації, яка визначає кожен піксель окремо, тут визначається ймовірність центру символу за допомогою гауссового теплового зображення. Теплові карти використовуються для навчання як оцінки регіону, так і оцінки спорідненості.



Рис. 1. Мапи областей та спорідненості у мережі CRAFT

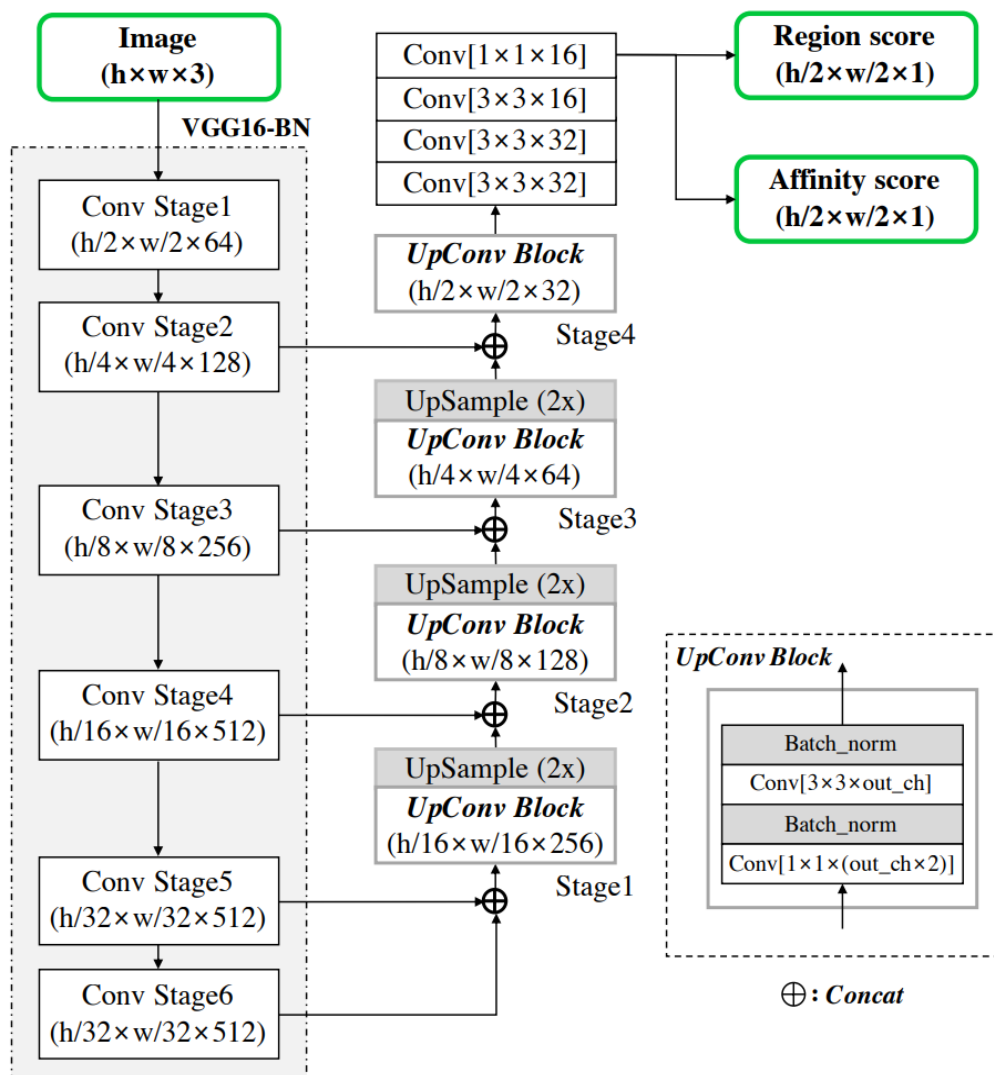


Рис. 2. Структура мережі CRAFT [3]

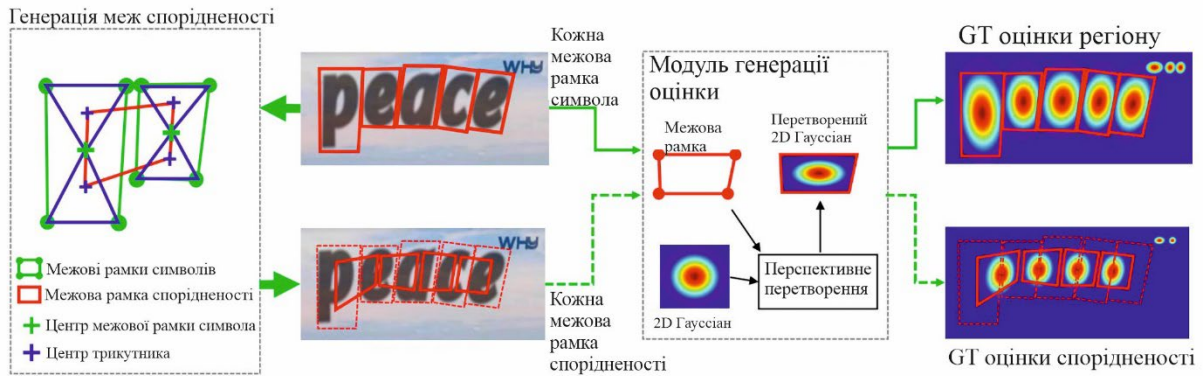


Рис.3. Ілюстрація процедури генерації еталонних значень [3]

Рис. 3 ілюструє процес генерації міток для синтетичного зображення. Обчислення значень гауссового розподілу для кожного пікселя в межах рамки символу є дуже затратним за часом. Оскільки межові рамки символів на зображенні зазвичай спотворюються через перспективні проєкції. Тут використано наступні кроки для апроксимації та генерації міток ground truth як для оцінки регіону, так і для оцінки спорідненості:

1. підготовка ізотропної 2D-гауссової карти;
2. обчислення перспективного перетворення між областю гауссової карти та рамкою кожного символу;
3. перетворення гауссової карти до області рамки символу.

Для міток ground truth оцінки спорідненості рамки спорідненості визначаються за допомогою суміжних рамок символів, як показано на рис. 3. Провівши діагональні лінії для з'єднання протилежних кутів кожної рамки символу, ми можемо створити два трикутники – верхній та нижній. Потім для кожної пари суміжних рамок символів рамка спорідненості створюється шляхом встановлення центрів верхнього та нижнього трикутників як кутів рамки. Запропоноване визначення міток ground truth дозволяє моделі достатньо виявляти великі або довгі текстові екземпляри. [3]

Модель складається із двох основних частин: видобування ознак (extractor) та їх об'єднання (merge). Через відсутність апаратної можливості тренування мережі CRAFT власноруч були використані значення параметрів моделі, навченої на наборах даних ICDAR2013 [6], ICDAR2017[7] та SynthText [8]. Це забезпечило універсальність мережі та здатність працювати із зображеннями різної візуальної структури.

Видобування ознак здійснюється послідовними шарами згортки та нормалізації. Велика кількість шарів згортки дозволяє мережі враховувати ознаки різного масштабу та складності. Клас extractor за основу використовує модель VGG16_BN. Для її ініціалізації використовуються допоміжні функції, які допомагають згенерувати шари нейронної мережі з визначеного списку та налаштовують значення шарів видобування ознак зображення. Клас VGG також використовує допоміжні функції для ініціалізації частини нейронної мережі.

З допомогою бібліотеки easydict реалізується файл конфігурації використання нейронної мережі для пошуку символів CRAFT. Файл містить параметри нормалізації вхідних зображень, використаних під час тренування мережі, шляхи запису вхідних та вихідних зображень, та обмеження для областей із символами та зв'язуючих областей. Коригування цих обмежень дозволяє працювати із зображеннями з різним відносним розташуванням символів та розділяти щільно розміщені області символів.



Рис. 4. Фрагмент тренувального набору даних після застосування випадкових трансформацій

Для роботи з будь-яким розміром зображення воно спочатку масштабується до розмірів, вказаних у конфігурації, записуючи відношення старих розмірів до нових. При завантаженні зображення у мережу воно перетворюється у спеціальний тип даних, з яким працює бібліотека Torch. Результати обчислень моделі необхідно перемістити назад з пам'яті обчислювального пристрою, де виконувались розрахунки.

Результатом обчислень мережі CRAFT є теплова мапа з областями символів та зв'язуючими областями (рис.3). У класичному варіанті використання мережі межі слів генеруються з суми значень цих мап. У даній модифікації для виділення окремих символів метод генерації меж віднімає від мапи символів зв'язуючу мапу. Це забезпечує відокремлення символів для їх легкого розпізнавання іншою нейронною мережею.

Взаємодія з наборами даних. Для генерації наборів даних для тренування мережі було використано зовнішній інструмент Text Recognition Data Generator [9]. У цій роботі застосовано набори даних із 62 класів: великі та малі літери латинського алфавіту, а також цифри від 0 до 9. Додатково створено окрему директорію з меншою кількістю екземплярів класів, яка не залучається до тренування, але використовується для формування тестового набору даних. Для підвищення здатності мережі до узагальнення зображення в наборі даних піддаються випадковим трансформаціям, зокрема поворотам, масштабуванню, зміщенням та додаванню шуму. Це сприяє створенню репрезентативного набору даних, що допомагає мережі навчитись розпізнавати зображення в неідеальних умовах, таких як нерівномірне освітлення або дефекти шрифтів (рис. 4). Для навчання мережі використовувався набір із 6200 зображень розміром 24x24x3 пікселі, які передавались на вхід моделі порціями по 24 зображення.

Модель-класифікатор для розпізнавання зображень символів. Модель для розпізнавання символів має класичну архітектуру класифікаторів. Вона складається із двох послідовних шарів згортки і об'єднання, та чотирьох повністю з'єднаних шарів.

Для підвищення ефективності процесу навчання мережі використовуються різноманітні методи оптимізації. Збереження імпульсу, тобто додавання до поточного градієнту зменшене попереднє значення, прискорює навчання мережі в цілому. Згасання значень ваг нейронів запобігає надмірному використанню певної частини мережі та забезпечує рівномірний вклад кожного її нейрону. Зниження швидкості навчання при досягненні плато у точності допомагає мережі навчитись враховувати дрібніші та менш помітні ознаки.

Обрані оптимізаційні параметри: збереження інерції 0.25, множник швидкості навчання 0.1, що домножується на 0.5 після 2-х епох без підвищення точності. Графік значень похибки на тренувальному та тестовому наборах даних при навчанні мережі на

згенерованому наборі даних відображає поточну точність похибку мережі кожної епохи (рис. 5).



Рис. 5. Графік похибки мережі у процесі тренування

Результатом навчання є мережа класифікації символів із точністю 83.9% на тестовому наборі даних.

Організація тестування та налагодження програмного засобу. Кожна ітерація навчання нейронної мережі супроводжувалась її тестуванням на окремому наборі даних. Це не дозволило мережі використовувати для класифікації символів ознаки, що присутні лише у тренувальному наборі даних.

Аналіз графіків точності мережі під час кожної епохи тренування допомагає правильно налаштувати параметри навчання та виявити достатню кількість епох, після якої зміна точності мережі не є суттєвою.

Для автоматичного створення графіків використовували AimStack – спеціалізований трекер метаданих процесів машинного навчання з відкритим вихідним кодом [10]. Для його використання достатньо передавати йому значення, які потрібно відслідкувати у ході тренування мережі. Ці значення автоматично обробляються та стають доступні для візуалізації засобами AimStack.

Головною проблемою у перших ітераціях тренування мережі була подібність певних символів при однаковому обтинанні зображення, зокрема символів “m” та “T” (рис. 6). Збільшення ширини символу “m” та подальше його горизонтальне стиснення допомогло мережі краще розрізнити ці символи.



Рис. 6. Подібність символів при неправильному обтинанні

Аналіз отриманих результатів дослідження. Створена програмна розробка з натренованими нейронними мережами здатна знаходити та розпізнавати символи з точністю 83,87%. Для оцінки точності використовувався тестовий набір даних, згенерований аналогічно до тренувального.

Натренована мережа розпізнавання найкраще працює із зображеннями, на яких символи добре відокремлені (рис. 7). Цей недолік можна частково компенсувати

масштабуванням нейронної мережі. Однак ефективнішим рішенням є використання рекурентних нейронних мереж для розпізнавання цілого слова.

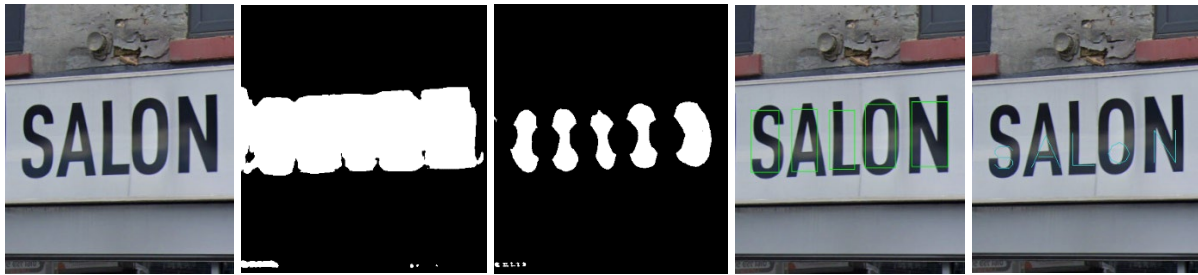


Рис. 7. Приклад допустимого вхідного зображення мапи регіонів символів, згенеровані мережею CRAFT, межі областей символів із результатами розпізнавання.

Нейронна мережа для пошуку тексту на основі CRAFT генерує мапу областей із символами (рис. 7). Точність знаходження тексту в специфічних умовах можна покращити, тренуючи модель на відповідному наборі даних. Коригування порогових значень при поділі областей на окремі символи може поліпшити результати у випадках незвичайного розташування символів у слові. Области з надто малою площею не передаються до другої нейронної мережі через недостатню роздільну здатність вхідних даних, що призводить до значного зниження точності розпізнавання (рис. 7)."

Висновки.

У роботі розроблено програмний інструмент для розпізнавання символів із використанням сучасних методів машинного навчання. Застосування двох нейронних мереж дозволило ефективно вирішити задачу локалізації та класифікації символів. Отримані результати демонструють точність розпізнавання символів на рівні 83,87%, що підтверджує ефективність запропонованого підходу. Тестування показало, що точність моделі значною мірою залежить від якості вхідних даних і способу підготовки тренувального набору.

Запропонований підхід є основою для подальших досліджень у сфері розпізнавання тексту, зокрема для задач, пов'язаних із аналізом текстів різних мов та шрифтів.

References

1. PyTorch documentation – PyTorch 2.5 documentation. *PyTorch*. URL: <https://pytorch.org/docs/stable/index.html>.
2. OpenCV: OpenCV modules. *OpenCV documentation index*. URL: <https://docs.opencv.org/4.x/index.html>.
3. Character Region Awareness for Text Detection. *arXiv.org*. URL: <https://doi.org/10.48550/arXiv.1904.01941>.
4. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
5. O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015.
6. Papers with Code - ICDAR 2015 Dataset. *The latest in Machine Learning | Papers With Code*. URL: <https://paperswithcode.com/dataset/icdar-2015> (date of access: 21.01.2025).
7. Papers with Code - ICDAR 2017 Dataset. *The latest in Machine Learning | Papers With Code*. URL: <https://paperswithcode.com/dataset/icdar-2017> (date of access: 21.01.2025).
8. GitHub - ankush-me/SynthText: Code for generating synthetic text images as described in "Synthetic Data for Text Localisation in Natural Images", Ankush Gupta, Andrea Vedaldi, Andrew Zisserman, CVPR 2016. *GitHub*. URL: <https://github.com/ankush-me/SynthText> (date of access: 21.01.2025).
9. GitHub - Belval/TextRecognitionDataGenerator: A synthetic data generator for text recognition. *GitHub*. URL: <https://github.com/Belval/TextRecognitionDataGenerator>.
10. Home | AimStack. *AimStack*. URL: <https://aimstack.io/>.