

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-53-05>

УДК 004.415.3

Пех Петро Антонович, к.т.н., доцент

<https://orcid.org/0000-0002-6327-3319>

Мельник Катерина Вікторівна, к.т.н., доцент

<https://orcid.org/0000-0002-9991-582X>

Шепелюк Дмитро Леонідович, магістр

Шепелюк Леонід Дмитрович, магістр

Луцький національний технічний університет, м. Луцьк, Україна

ДОСЛІДЖЕННЯ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ ЗА ДОПОМОГОЮ МОБІЛЬНИХ ДОДАТКІВ

Пех П.А., Мельник К.В., Шепелюк Д.Л., Шепелюк Л.Д. Дослідження технології розпізнавання номерних знаків за допомогою мобільних додатків. В статті досліджуються технології розпізнавання номерних знаків засобами сучасних смартфонів.

Ключові слова: система розпізнавання номерних знаків, оператор Кенні, оператор Собеля, оператор Лапласа

Pekh P., Melnyk K., Shepelyuk D., Shepelyuk L. Study of license plate recognition technology using mobile applications. The article examines license plate recognition technologies using modern smartphones.

Key words: Number plate recognition system, Kenny operator, Sobel operator, Laplace operator

Постановка задачі. В час бурхливого розвитку комп'ютерних технологій усе більш актуальною задачею є розробка та використання систем розпізнавання образів для різних потреб: визначення і розпізнавання обличчя, знаходження конкретних образів на відео та зображеннях з метою оброблення отриманих шляхом розпізнавання даних. Ймовірно, що найбільш широким напрямом цієї тематики є розпізнавання текстової інформації, і саме тому оцифрування та оброблення текстових даних мають велике значення. Однією з підзадач розпізнавання тексту є розпізнавання автомобільних номерів.

Транспортні засоби мають унікальний ідентифікаційний номер в якості основного розпізнавального знаку. В них можуть використовуватися українські літери, літери латиниці, цифри, а також малюнки [1]. В залежності від виду, номерні знаки можуть мати різний розмір, колір тла та символів.

«Сьогодні в Україні існує 15 типів номерних знаків (більше двадцяти підтипів). На номерних знаках використовуються українські літери, що мають графічні відповідники у латиниці (усього 12: А, В, С, Е, Н, І, К, М, О, Р, Т, Х) та літери української та латинської абеток: А, В, Г, Е, І, К, М, Н, О, Р, С, Т, Ф, Х, Ч, Ю, Я» [2].

На сьогоднішній день існує велика кількість різноманітних систем розпізнавання автомобільних номерів, серед яких вирізняють два типи – апаратні та програмні. Апаратні системи розпізнавання автомобільних номерів, це системи, які визначають номер за допомогою вбудованих у камери засобів.

Розрізняють системи розпізнавання номерів з різним рівнем якості розпізнавання, величиною швидкодії і спектром послуг додаткових функцій [3, 4]. Сучасні системи (LPR-системи) характеризуються високою швидкістю і точністю розпізнавання (97-98%), і, як правило, дуже дорогі та вимагають стаціонарної установки та відповідної підтримки (спеціального сервера, який і виконує всі необхідні операції оброблення зображень). Висока вартість LPR-систем не дозволяє використовувати їх на побутовому рівні. В даний час існує досить велика кількість мобільних систем визначення номерних знаків [3,4]. В основному це платні продукти, і алгоритм їх функціонування засекречений і є комерційною таємницею.

Альтернативою дороговартісним камерам є застосування мобільних додатків смартфонів[5,6]. Цей підхід може бути більш вдалим, якщо мова йде про заощадження коштів, оскільки для розгортання такої системи не потрібне додаткове дороге обладнання. Характеристики камери в сучасних моделях смартфонів наближаються до професійних камер. Смартфони завжди під рукою, надають можливість робити знімки автотранспорту у звичайних умовах “на льоту”, та використовувати програмну систему розпізнавання номерів.

Метою дослідження було дослідити різні технології розпізнавання номерних знаків, з'ясувати, які з цих технологій можна реалізувати за допомогою сучасних смартфонів і експериментально перевірити ефективність таких систем.

Новизна дослідження полягає у тому, що для побудови системи розпізнавання номерних знаків використовуються смартфони, що істотно зменшує на її вартість.

Основна частина. Розглянемо два підходи до розпізнавання образів.

Перший підхід — апаратний — ґрунтується на оптичному розпізнаванні образів, що дозволяє застосувати методи перебору вигляду об'єкта під різними кутами, масштабами, зсувами й т. д. Для розпізнавання символів потрібно перебирати різні шрифти, властивості шрифтів т. д. Даний метод складно реалізувати засобами смартфону, оскільки потребує кілька камер та деяку кількість знімків.

Другий підхід — програмний — передбачає розробку мобільного додатку здатного опрацювати знімок автотранспорту та розпізнавати номерні знаки.

Основним завданням програмного забезпечення знайти контур об'єкта — номерної рамки на знімку та розпізнати номерні знаки. Існує безліч алгоритмів для виділення контурів, такі як: оператор Собеля, оператор Лапласа, оператор Робертса, оператор Прюїтт і оператор Кенні[9]. Детектування контурів необхідно в тому випадку, коли є досить складне зображення Використовуючи інструменти бібліотеки комп'ютерного зору необхідно виділити об'єкти на цьому зображенні для подальшої роботи з ними.

Весь процес визначення автономеру можна записати як послідовність кроків [16] попередньої обробки зображення, вибірки даних, на основі зображення, та аналізу зібраних даних. Оскільки номерні знаки є комбінацією цифр і літер - то розпізнавання номера зводиться до алгоритмів розпізнавання тексту і цифр[15].

Алгоритм визначення номерних знаків із застосуванням бібліотек оптичного розпізнавання наведено на рисунку 1.

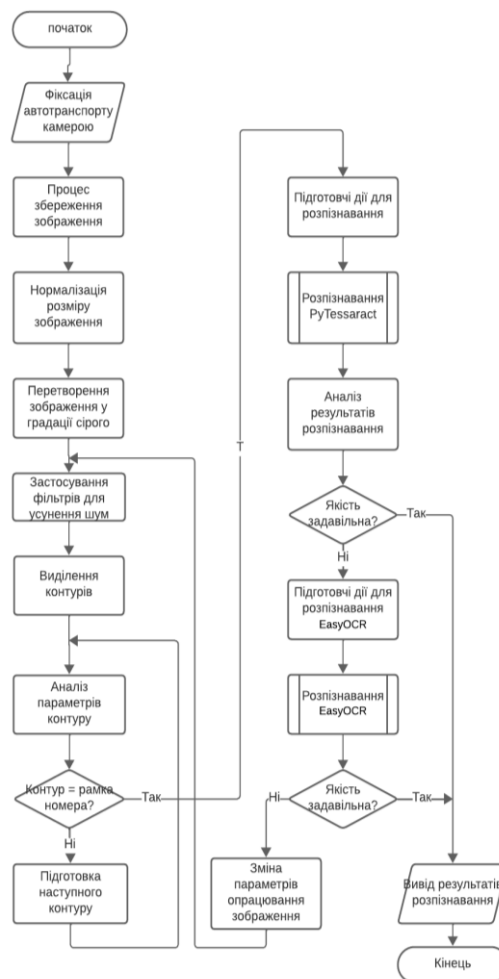


Рис. 1. Алгоритм визначення номерних знаків із застосуванням бібліотек оптичного розпізнавання

Варто відзначити, що всі наступні методи працюють з чорно-білими зображеннями. Тобто, спершу вихідна картинка з RGB переводиться у відтінки сірого. Приклад перетворення зображення показано на рис. 2.

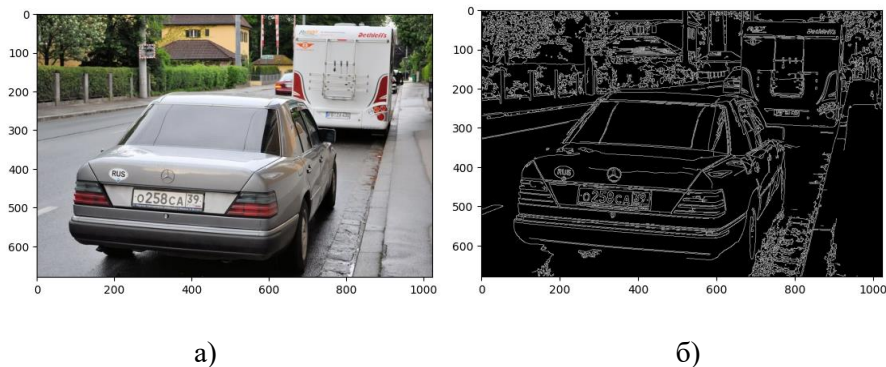


Рис. 2. Результат перетворення
а) вхідне зображення; б) виділення контурів застосування оператора Кенні

Наступним етапом алгоритму є компонентний аналіз: це послідовний аналіз виділених контурів з урахуванням їх площі, геометрії і т.д. Це доволі витратний за ресурсами етап, оскільки вимагає великої кількості обчислень: перевірка кожного контуру, а також child-контурів.

В даний час існує більше 10 алгоритмів, що вирішують завдання фільтрації контурів, ми перерахуємо основні 5: оператор Кенні, оператор Собеля, оператор Лапласа, оператор Прюїтт, оператор Робертса. Найчастіше використовується саме оператор Кенні (рис. 3), який виокремлює контури при правильних налаштуваннях фільтрів. Програмна реалізація попередньої обробки із застосування бібліотеки комп'ютерного зору OpenCV зображена на рис. 3

```
import cv2
import numpy as np
image = cv2.imread('cntr.jpg') #прочитати зображення
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) #відтінки сірого
blurred = cv2.GaussianBlur(gray, (5, 5), 0) #зменшення шуму
# Використовуємо поріг для виділення контурів
_, thresh = cv2.threshold(blurred, 128, 255, cv2.THRESH_BINARY)
# thresh = cv2.Canny(blurred, 30, 150) #або застосування оператора Canny
# виділення контурів на зображенні
contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
                               cv2.CHAIN_APPROX_SIMPLE)
#Сортування контурів за площею (вибір максимального)
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:1]
# Збереження виокремленого контуру у новий файл
cv2.imwrite('cntr.jpg', contour)
```

Рис. 3. Програмна реалізація етапів попереднього оброблення зображення

Далі зупинимося на задачі розпізнавання номерних знаків зі застосуванням бібліотек комп'ютерного зору.

Розпізнавання символів являється класичною задачею. Існує безліч рішень, та певні з них майже досконалі. Найкращими такими алгоритмами є алгоритм Tesseract OCR[7] (рис. 4) та його аналоги.

PyTesseract – це бібліотека, яка була розроблена у 90-тих роках минулого століття, та з 2006-го року використовується компанією Google. Бібліотека має підтримку юнікоду UTF-8, вміє розпізнавати більше 100 мов світу. Вихідний формат розпізнавання може бути різним: звичайний текст, html, pdf, tsv, тощо.

Одним із варіантів є застосування бібліотеки PyTesseract - розпізнавання текст на зображеннях або в графічних файлах.

```
!pip install pytesseract
!sudo apt install tesseract-ocr
!sudo apt-get install tesseract-ocr-ukr
import pytesseract
. . .
# Зчитуємо зображення
image = cv2.imread('path/to/image/cntr.jpg')
# Перетворюємо в відтінки сірого якщо це не було попередньо зроблено
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
# Розпізнаємо текст на зображенні
text = pytesseract.image_to_string(gray)
print(«Розпізнаний текст:», text)# Виводимо результат
```

Рис.4. Фрагмент коду розпізнавання тексту на зображенні із застосуванням бібліотеки PyTesseract

EasyOCR[12] (рис.5) - це бібліотека для розпізнавання тексту, яка базується на deep learning та використовує предобробку та архітектури нейронних мереж для ефективного розпізнавання тексту на зображеннях.

```
import imutils
import easyocr
#попередня обробка зображення
img = cv2.imread('/content/drive/MyDrive/Colab Notebooks/cars/4.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGRA2GRAY)
edges = cv2.Canny(img_filter, 30, 200)
cont = cv2.findContours(edges.copy(), cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
cont = imutils.grab_contours(cont)
cont = sorted(cont, key = cv2.contourArea, reverse= True)[:2]
crop = gray[x1:x2, y1:y2]
text = easyocr.Reader(['ru'])
text =text.readtext(crop)
print(text)
```

Рисунок 5 – Фрагмент коду розпізнавання тексту на зображенні із застосуванням бібліотеки EasyOCR

Експериментальні дослідження. Розроблення програмного забезпечення проводилася з використанням мови програмування Python 3 в середовищі Google Collaboratory[17], що надає можливість запускати та писати код Python в хмарному середовищі, не встановлюючи на своєму комп'ютері Python або інші програмні засоби.

Для автоматичного тестування використано датасет: Automatic Number Plate Recognition [13]

Для мануальної перевірки роботи програмного засобу була зібрана невелика кількість зображень автомобілів з номерами - скріншоти фотографій автомобілів [14] з середньою роздільною здатністю 1000x700, що наближена до роздільної здатності камери смартфона (1280x720), які можна розпізнати людським оком.

Результати експериментів з параметрами конфігурації та оптимізації зображень та використання різних методи підготовки зображень (наприклад, зменшення шуму, бінаризація тощо) відображені в таблиці 1.

Таблиця 1 – Результати розпізнавання номерних знаків. Вибірка нерозпізнаних номерів

Зображення	Результати розпізнавання PyTesseract	Результата розпізнавання EasyOCR	Номерний знак	Примітки
Датасет №1 Dataset Automatic Number Plate Recognition [15]				
247.jpg	TN078U5427	TN07BU5427	TN07BU5427	
244.jpg	KL 05AK3300	K105AK3300	KL	розпізнано

Зображення	Результати розпізнавання PyTesseract	Результата розпізнавання EasyOCR	Номерний знак	Примітки
			05AK3300	
242.jpg	MH 20DV2362	MH 20DV2362	MH 20DV2362	
238.jpg	H20EE1591	MH20EE7597	MH20EE7597	не чіткий
249.jpg	не розпізнано	не розпізнано	не розпізнано	не розпізнано
241.jpg	KV 57 CK0	KV 57 CK	KV 57 CKC	під кутом
246.jpg	LMH06AW8929	MH06AW8929	MH06AW8929	під кутом
240.jpg	TN 09CH7770	TN 090H77701	TN 09CH7770	не чітке
4.jpg	24KZT5	24KZT5	24KZT5	жовтий фон
7.jpg	не розпізнано	не розпізнано	77ZFK8	жовтий фон, додатковий напис
20.jpg	JPI481	JP548J	JP548J	жовтий фон "змазаний"
9.jpg	не розпізнано	не розпізнано	MH14TC206/AN	фон рамки та кузова -червоний
Датасет №2 Власна колекція [16]				
301.jpg	не розпізнано	не розпізнано	03EC0213	червона рамка, білі знаки
302.jpg	KA 0222BX	KA 0222BX	KA 0222BX	
303.jpg	не розпізнано	не розпізнано	E SKM LTD	
304.jpg	HH0095CC	HH0095OO	HH0095CC	

Висновки. В процесі дослідження було створено макет мобільної системи розпізнавання номерів автотранспортних засобів, яка реалізована засобами мови Python. Для аналізу результатів роботи програмного засобу було використано датасет від Kaggle та власна колекція зображень автомобілів з номерами, які можна розпізнати людським оком, та мануально перевірити роботу програмного засобу.

За результатами ручного тестування якості розпізнавання номерних знаків для датасет №1 складає 83%, для датасет №2 – 87%.

Причиною поганого розпізнавання є нечіткі або забруднені номерні рамки та знаки. Часто причиною поганого розпізнавання є червоне або жовте тло рамки, білий колір знаків. Підвищити якість розпізнавання можна, застосовуючи адаптивні фільтри, змінюючи порогове значення, за яким відбувається бінаризація. Експериментальні дослідження показали, що порогове значення кількості об'єктів сегменту для уникнення хибної детекції номерного знаку повинно бути не більшим за 20.

Програмний засіб відмінно розпізнає номерні знаки та високоякісні зображення автомобіля з роздільною здатністю 1920x1080 та забрудненим номером.

Для чітких зображень причиною помилки в розпізнаванні було те, що автомобіль був такого ж кольору, як і тло автомобільного номеру, що ускладнює процес виявлення рамки номерного знаку.

Тимчасові номери, які мають червоне тло номеру і білі знаки, програмний засіб розпізнати не зміг. Така ситуація виникла під час навчання каскаду Хаара. Було використано лише декілька зображень з номерами не білого кольору, тому алгоритм в деяких випадках не знаходив рамки номеру.

Розроблений застосунок потребує інтелектуальної системи налаштування попередньої обробки зображень. Але для непрофесійного, побутового застосування така точність розпізнавання буде достатня.

Таким чином, у статті наведені технології та результати дослідження систем розпізнавання номерів транспортних засобів, які можуть бути застосовані під час розроблення нових та модернізації існуючих систем.

Список бібліографічного опису

1. Номерні знаки України. <https://avtonomera.net.ua/ua/nomerni-znaky-avtomobilni-nomera-ukrayiny.html>
2. AutoVu automatic license plate recognition (ALPR). <https://www.genetec.com/products/unified-security/autovu>
3. <https://www.nedapidentification.com/products/filter/license-plate-recognition-solutions/passive-uhf-solutions/>
4. Додатки в Google Play NPRReader (full version). <https://play.google.com/store/apps/details?id=com.imense.imenseANPRfull&hl=uk&gl=US>
5. Додатки в Google Play Carmen Mobile. <https://play.google.com/store/apps/details?id=com.arh.anprclientwithopencv&hl=uk&gl=US>
6. ДСТУ 4278:2019. Дорожній транспорт. Знаки номерні транспортних засобів. Загальні вимоги. Чинний від 16.03.2020. Київ: ДП "УкрНДНЦ", 2020. 31 с.
7. Подорожняк А.О. Нейромережева система розпізнавання автономера / А.О. Подорожняк, Н.Ю. Любченко, Г.В. Гейко // Системи управління, навігації та зв'язку. Полтава: ПНТУ ім. Ю. Кондратюка. Вип. 4 (62). 2020. С. 88 – 92. doi: <https://doi.org/10.26906/SUNZ.2020.4.088>.
8. Dataset: скріншоти автомобілів сайту <https://auto.ria.com/uk/>
9. Dataset Automatic Number Plate Recognition <https://www.kaggle.com/datasets/aslanahmedov/number-plate-detection>
10. Портал документації Open. eVision 2.5.1 .1107/ https://documentation.euresys.com/Products/Open_eVision/Open_eVision_2_5/en-us/Content/03_Using_Open_eVision/D1_EasyOCR_-_Reading_Texts/EasyOCR_-_Reading_Texts.html
11. Бібліотека сторонніх пакетів Python (PyPI). <https://pypi.org/project/easyocr/>

References

12. Tesseract documentation. Tesseract OCR. URL: <https://tesseract-ocr.github.io/> (date of access: 12.11.2023).
13. OpenCV: OpenCV-Python Tutorials. OpenCV documentation index. URL: https://docs.opencv.org/3.4/d6/d00/tutorial_py_root.html (date of access: 12.11.2023).
14. Document Analysis and Recognition/ ICDAR 2023 17th International Conference, San José, CA, USA, August 21–26, 2023, Proceedings, Part 3/ 2023
15. Lakhmi C. Jain, Beatrice Lazzarini. Knowledge-Based Intelligent Techniques in Character Recognition Lakhmi C. Jain, Beatrice Lazzarini. 2020
16. Matthew Rever. Computer Vision Projects with OpenCV and Python 3. 2018
17. Google Colaboratory. Google Colab. URL: <https://colab.research.google.com/> (date of access: 12.11.2023).