

AUTOMATION AND ROBOTICS АВТОМАТИЗАЦІЯ ТА РОБОТОТЕХНІКА

УДК 004.8:681.5

DOI:10.30837/rt.2024.4.219.05

О.В. ЗУБКОВ, канд. техн. наук, О.С. ЯКОВЕНКО, С.В. СТАРОКОЖЕВ, М.В. СКОРБАТЮК

РОЗРОБКА ТА ДОСЛІДЖЕННЯ АЛГОРИТМУ АВТОМАТИЗОВАНОГО РОЗПІЗНАВАННЯ ПОКАЗАНЬ ГАЗОВИХ ЛІЧИЛЬНИКІВ

Вступ

Останні роки характеризуються швидким розвитком систем комп'ютерного зору, які активно використовуються для контролю якості продукції, в медицині, в системах дорожнього руху, робототехніці і т.д. [1 – 4]. Завдяки методам та алгоритмам комп'ютерного зору вдається автоматизувати багато процесів та реалізувати прямий зв'язок із базами даних для логування стану систем чи параметрів. В Україні в комунальних системах використовують в основному механічні лічильники або лічильники без автоматичного передавання показань споживання населенням: газу, світла, води. Для зняття показань використовують контролерів, які переписують ці показання. Такий підхід може призводити до помилок фіксації внаслідок людського фактора чи навмисних викривлень результатів перевірки. Аналогічна ситуація із метрологічною атестацією вимірювальних лічильників, коли треба фіксувати результати протоку через лічильник газу або води. Тому для України і інших країн актуальна задача створення алгоритмів та програмного забезпечення, що дозволять автоматично розпізнавати та зберігати результати розпізнавання показань лічильників при метрологічній повірці чи при знятті показань контролером [5]. Огляд світових напрацювань з цього напрямку показав, що існуючі методи потребують вдосконалення для покращення точності розпізнавання. Метою даного дослідження є створення високоточного алгоритму розпізнавання показань лічильників газу та дослідження його ефективності.

Аналіз зображень для розпізнавання та сучасних методів розпізнавання зображень

Існує багато типів лічильників газу за принципом вимірювань: турбінні, мембранні, роторні. Зовнішній вигляд декількох екземплярів лічильників наведено на рис. 1.



Рис. 1. Зовнішній вигляд лічильників газу різних типів

Аналіз зображень зовнішнього вигляду лічильників показує, що показання відображаються у вигляді п'ятизначного числа, що відповідає кількості кубометрів та тризначного числа, що відповідає кількості літрів. Також існують лічильники із електронним дисплеєм та відображенням кількості літрів у вигляді чотиризначного числа. Фактично показання лічильника – це текстова інформація (символи від 0 до 9). Крім показань на зображенні є ще багато різної текстової інформації про тип лічильника, виробника, номер і т.д.

Для розпізнавання текстової інформації існує три основних підходи: класичний, на основі нейронних мереж глибокого навчання, на основі нейронних мереж та трансформерів.

Класичний підхід [6, 7] передбачає фільтрацію зображення для зменшення впливу шуму при поганому освітленні, адаптивне порогове перетворення для чіткого виділення контурів написів, використання контурного аналізу для виділення символів та оптичне розпізнавання символів для перетворення контурів у текстову інформацію. Хоча класичний підхід забезпечує високу швидкодію, але має низку недоліків: чутливість до умов зйомки та бліків, якості зображень, необхідність другого етапу обробки для відокремлення показань від інших написів.

Підхід на основі нейронних мереж використовує навчені нейронні мережі, що дозволяють класифікувати об'єкти на зображеннях та знаходити їх локалізацію, тобто координати обмежувальних рамок [8 – 12]. До найбільш популярних архітектур нейронних мереж з такими можливостями відносяться: Mask R-CNN, Fast R-CNN, Faster R-CNN, SSD, YOLO (You Only Look Once), DETR. Найбільшу точність розпізнавання забезпечує Mask R-CNN бо це двохпрохідний метод, що знаходить контури об'єктів, а потім їх розпізнає. Але цей метод працює дуже повільно і при використанні звичайних процесорів потребує до десятків секунд на обробку зображень. Архітектури Fast R-CNN та Faster R-CNN оптимізовані для збільшення бистротії, мають високу точність розпізнавання, але все одно не забезпечують обробки у реальному часі в силу додаткового часу на реалізацію обробки регіонів (Region Proposals). Найбільшу швидкодію забезпечують архітектури SSD та YOLO. При цьому YOLO також дає баланс між точністю та швидкістю і має версії архітектури адаптовані до роботи на апаратних платформах з обмеженою швидкістю.

Прикладом реалізації підходу на основі трансформерів є алгоритм DETR [13]. В цьому алгоритмі пріоритетом є точність розпізнавання, а не швидкість. Хоча, при цьому, він має непогані показники швидкості – кращі, ніж у Fast R-CNN, Faster R-CNN.

Ефективність розпізнавання зображень сучасними нейронними мережами характеризується рядом параметрів: TP – кількість вірно розпізнаних об'єктів, FP – кількість помилкових передбачень, FN – кількість об'єктів, які модель не знайшла, Precision – частка правильно детектованих об'єктів серед усіх передбачених, Recall – частка знайдених правильних об'єктів серед усіх справжніх об'єктів, IoU – міра порівняння області передбаченого об'єкта з реальною анотацією, mAP – середня точність для різних значень порогів Intersection over Union (IoU) [14, 15]. При навчанні Precision та Recall розраховуються для різних значень порогів IoU за формулами (1) та (2) та будується PR крива:

$$Precision = \frac{TP}{TP + FP}, \quad (1)$$

$$Recall = \frac{TP}{TP + FN}. \quad (2)$$

Усереднене значення Precision розраховується, як площа під PR кривою. Значення mAP розраховується, як середнє значення AP для всіх класів за формулою (3). Розрахунок зазвичай ведеться для діапазону порогів IoU від 0,5 до 0,95.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i, \quad (3)$$

де N – кількість класів.

Аналіз багатьох наукових публікацій і Internet ресурсів з оцінок ефективності роботи різних нейронних мереж дозволив створити об'єднану таблицю показників ефективності та характеристик мереж (табл. 1).

Таблиця 1

Архітектура	Швидкодія на RTX3090, FPS	Розмір моделі, Мбайт	Точність mAP ₅₀₋₉₅ , %	Кількість параметрів x 10 ⁶
Mask R-CNN	5-7	180	50-60	44
Fast R-CNN	6-9	150	45-60	23
Faster R-CNN	7-10	160-190	40-60	41
SSD512	20-40	13-140	30-45	24
YOLOv8-11	30-120	4-190	40-58	1.85-68
DETR	5-10	140-200	40-60	41

У табл. 1 наведена інформація про швидкодію роботи нейронних мереж у кількості зображень за секунду (FPS) для відео карти RTX3090, розмір файлу моделі, кількості параметрів в моделі та точності розпізнавання моделі після навчання на стандартних датасетах COCO. Аналіз даних табл. 1 показує, що архітектура SSD512 значно програє іншим архітектурам по точності, а кращі архітектури за точністю (Mask R-CNN, Faster R-CNN) працюють достатньо повільно, мають великий об'єм моделі та потребують для реалізації значних апаратних ресурсів. Найкращим варіантом для практичної реалізації є застосування алгоритма YOLO останніх версій починаючи із восьмої. Архітектура YOLO є достатньо гнучкою і має 5 різновидів: n,s,m,l,x, що орієнтовані, як на прості апаратні платформи типу Raspberry PI (моделі n та s) чи мобільні телефони (моделі s та m), так і на використання GPU (моделі s-x). Ці модифікації відрізняються кількістю ядер в кожному із загорткових шарів та складністю інших блоків. Невелика кількість параметрів моделей, від 2 мільйонів, дозволяє реалізувати мережу на невеликому об'ємі оперативної пам'яті, а висока точність дає можливість ефективного розпізнавання. Аналізуючи інформацію компанії розробника архітектури YOLO з сайту <https://ultralytics.com/> для подальших досліджень була обрана архітектура YOLOv10s. При найменшій кількості параметрів (7.2 мільйона) ця архітектура забезпечує точність розпізнавання mAP=47 %.

Для розпізнавання показань газових лічильників необхідно підготувати датасет зображень різних лічильників та навчити нейронну мережу YOLOv10s з використанням цього датасету. Відповідно до рекомендацій, викладених в [8 – 11, 14, 15], для якісного навчання необхідно мати не менше 300 зображень кожного класу. В нашому випадку є 10 класів об'єктів – цифри від 0 до 9. Тоді загальна кількість зображень цифр на лічильниках повинна бути не менше 3000. Виходячи з того, що на одному зображенні лічильника є 8 цифр, в датасеті повинно бути не менше 375 фото лічильників. Існує багато Інтернет ресурсів готових датасетів. Серед них можна відмітити: Kaggle, RoboFlow, TensorFlow Datasets, Google Dataset Search і т.д. У датасетах ресурса Kaggle є лише зображення окремих цифр, що не дозволяє при навчанні та розпізнаванні тренувати мережу з локалізації положення цифр на зображенні лічильника. Датасети ресурсу RoboFlow повністю задовольняють потребам алгоритму YOLO при навчанні: містять безпосередньо зображення лічильників різних типів та виробників, зроблених в різних умовах зйомки (різне освітлення, світлові блики), розмір кожного зображення 416x416 пікселів. В результаті об'єднання знайдених датасетів, було створено датасет, що містить 1142 зображення лічильників відомих світових виробників для навчання, 342 – для валідації та 171 – для тестування. Частина датасету для навчання також була доповнена 52 зображеннями лічильників українських виробників. Далі всі зображення були анотовані з використанням програми LabelImage. При анотуванні кожна цифра показань виділяється обмежувальною рамкою, на кожне зображення лічильника створюється текстовий файл з переліком класів об'єктів та координатами обмежувальних рамок.

Навчання архітектури YOLOv10s та аналіз результатів навчання

Для навчання використовувалась бібліотека ultralytics та переднавчена модель yolov10s, в якій вихідний шар із 80 нейронами замінений на шар із 10 нейронів, кількість яких відповідає кількості класів об'єктів, що треба розпізнавати. Навчання запускається викликом метода train. При виклику цього метода були задані наступні параметри: кількість епох навчання –

300, розмір батча – 16 зображень, можливі нахили зображень перед подачею на нейронну мережу – до $\pm 8^\circ$, оптимізатор – RMSProp. При навчанні кожна епоха завершується тестуванням на валідаційному датасеті для оцінювання оптимальності параметрів моделі та розрахунку параметрів результатів навчання, що обчислюються за формулами (1) – (3). При досягненні найбільшого значення mAP_{50-95} зберігається досягнута оптимальна сукупність параметрів моделі. По завершенню 300 епох навчання було досягнуто високу ефективність розпізнавання: Precision=99,5 %, Recall=99,7 %, $mAP=99,3$ %. Також на рис. 2, а наведена матриця помилок результатів навчання, що демонструє по діагоналі імовірності вірного розпізнавання 10 класів, а також імовірності помилок розпізнавання класів та фону.

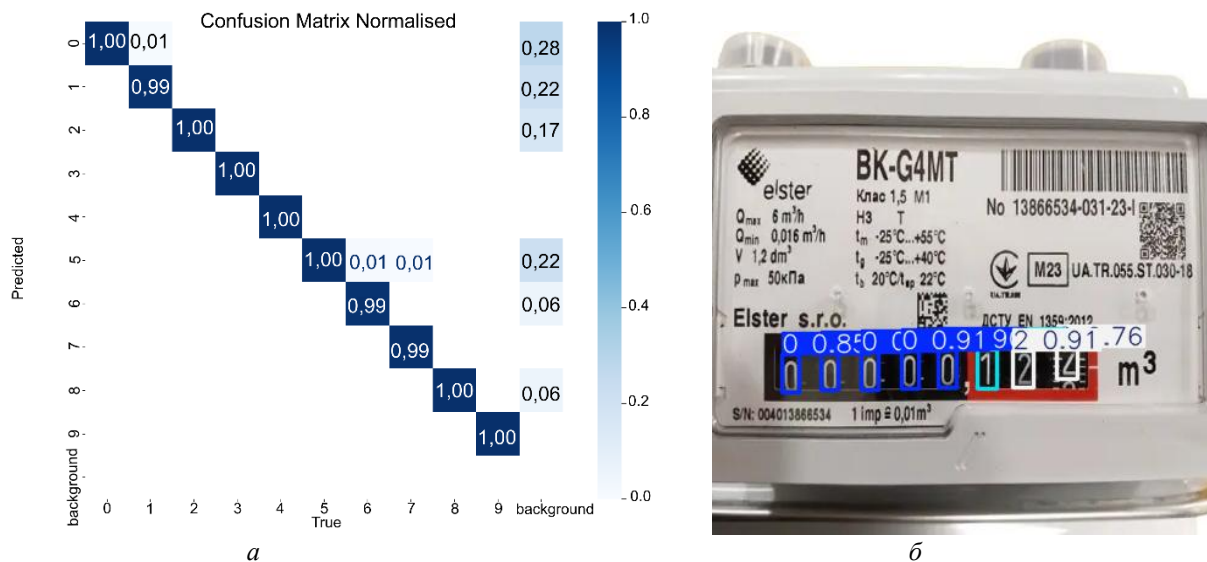


Рис. 2. Матриця розпізнавання цифр та приклад розпізнавання показань лічильника

В результаті навчання формується сукупність обмежувальних рамок, що завдана координатами цих рамок, номери класів розпізнаних цифр, що співпадають із самими цифрами та значення конфіденційності, що характеризують достовірність розпізнавання та локалізації. На рис. 2, б наведено приклад результату розпізнавання з обмежувальними рамками.

Як можна побачити з даних рис. 2, а імовірності розпізнавання 10 класів достатньо високі від 0,99 до 1. Однак також можна побачити високі імовірності розпізнавання фрагментів фону, як цифр показань лічильника. Це пов'язано із наявністю значної кількості цифрової інформації на лицьовій панелі лічильників. Аналіз результатів розпізнавання, наприклад, рис. 2, б, показує, що сформований масив обмежувальних рамок та класів об'єктів, що відповідають цим рамкам, є не відсортованим за горизонтальними координатами обмежувальних рамок. Тому, в прикладі рис. 2, б, послідовність даних в масиві містить не реальну кодову комбінацію, а код 20001200. Аналіз результатів розпізнавання інших зображень лічильників показує, що кількість обмежувальних рамок може бути більше 7, коли фрагменти фону розпізнаються помилково, як цифри лічильника. Також існують випадки помилкового розпізнавання цифр лічильника. В результаті аналізу тестової частини датасету була отримана імовірність вірного розпізнавання цифр лічильника 0,95.

Вдосконалення алгоритму розпізнавання

Отримані результати показують, що використання стандартного алгоритму YOLOv10s недостатньо для отримання достовірних результатів розпізнавання і потрібно додати вторинну обробку результатів розпізнавання. У вторинній обробці було реалізовано аналіз координат обмежувальних рамок. На першому етапі виконувалось сортування масиву обмежувальних рамок за їх горизонтальними координатами. Таким чином, наприклад, для рис. 2, б, була отримана реальна послідовність символів 00000122. Також було додано аналіз вертикальних

координат обмежувальних рамок. Як можна побачити з рис. 2, б, центри усіх обмежувальних рамок знаходяться вздовж однієї лінії з невеликим відхиленням кожного з центрів. Аналіз результатів розпізнавання валідаційної та навчальної частин датасету показала, що середньоквадратичне відхилення вертикальної координати центрів обмежувальних рамок від лінії дорівнює $\sigma=1,9$ пікселя. На основі цих даних з масиву обмежувальних рамок видаляються ті, у яких вертикальне відхилення центрів від лінії перевищує 3σ , тобто 6 пікселів. Після введення сортування та видалення зайвих рамок вдалось підвищити імовірність вірного розпізнавання до 0,99. Але це, все одно, означає, що помилки можливі при невірному розпізнаванні цифр і є необхідність зменшити імовірність помилок.

У зв'язку з тим, що для навчання використовувався великий датасет, втричі більший за мінімально потрібний для алгоритму YOLO, намагалися покращити модель за рахунок збільшення розміру датасету немає сенсу. Тому у вторинну обробку було додано альтернативне розпізнавання зображень під обмежувальними рамками за допомогою розробленої нами згорткової нейронної мережі [16, 17]. На вхід розробленої загорткової мережі подається зображення з висотою 32 пікселя та шириною 16. Такий розмір обрано на основі статистичного аналізу сукупності обмежувальних рамок цифр в зображеннях лічильників в тренувальній та валідаційній частинах датасету. Структура розробленої загорткової нейронної мережі показана на рис. 3.

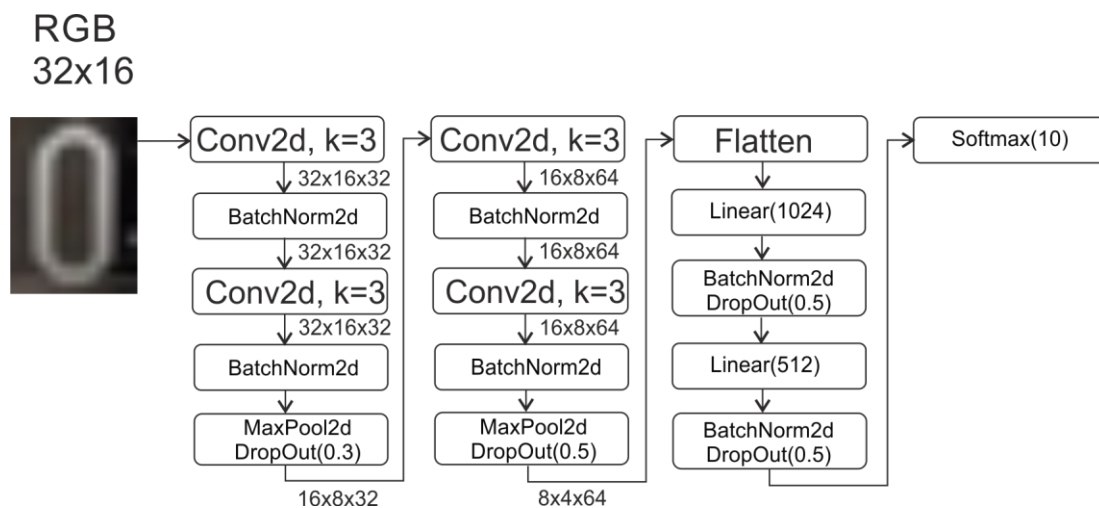


Рис. 3. Структура розробленої згорткової нейронної мережі

На вхід нейронної мережі подаються трьохкольорові зображення з розмірами 32x16 пікселів. В загортковому шарі Conv2d виконується згортка з 32 ядрами 3x3 і, в результаті, маємо 32 карти ознак. Використовується функція активації ReLU. Для прискорення навчання та стабілізації вагів мережі додано шар BatchNorm2d. Далі повторюються згортковий та нормалізаційний шари. Після них розміри карт зменшуються вдвічі завдяки MaxPool2d шару. Також для запобігання перенавчанню додано шар Dropout, що випадково обнуляє ваги мережі. В результаті маємо 32 карти ознак з розмірами 16x8. Далі описана послідовність блоків повторюється з 64 ядрами і ми маємо 64 карти ознак з розмірами 8x4. Після згорткових шарів йде повнозв'язна мережа, що складається із шарів Linear по 1024 і 512 нейронів, а також вихідний шар для класифікації Softmax, що має 10 виходів. Для переходу від двомірних карт ознак до векторної структури входів повнозв'язної мережі використовується шар Flatten. Реалізація мережі була виконана на PyTorch.

Розроблена нейронна мережа була навчена з використанням навчальної частини датасету. В результаті перевірки навченої моделі на валідаційній частині датасету були отримані наступні результати: Precision=99,1%, Recall=99,2%, mAP=98,8%. Ці результати дещо поступаються результатам навчання моделі YOLO, що пояснюється більш простою структурою нейронної мережі.

Вдосконалений алгоритм було використано на тесовій частині датасету. Рішення про вірний результат розпізнавання робився при умові, що обидві нейронних мережі показали однаковий результат класифікації і значення конфіденційності в алгоритмі YOLO не менше 0,75, а значення на виході розробленої нейронної мережі для цього класу не нижче 0,9. При цих параметрах вдалось виявлять усі невірні розпізнані цифри однією з мереж. У випадку неспівпадання результатів розпізнавання двох нейронних мереж пропонується робити новий фотознімок та повторювати процес розпізнавання для отримання достовірних даних.

Висновки

1. Проведено аналіз сучасних методів розпізнавання зображень для вирішення проблеми автоматизованого розпізнавання показань газових лічильників. Для розпізнавання обрано найновітнішу архітектуру YOLOv10s, що забезпечує компроміс між точністю розпізнавання та роботою у реальному часі, у порівнянні з іншими методами та архітектурами нейронних мереж.

2. Для навчання архітектури YOLOv10s розроблено датасет на основі існуючих Internet ресурсів та фото українських виробників лічильників. З використанням розробленого датасету було виконано навчання YOLOv10s та тестування навченої нейронної мережі. Результати навчання показали високу ефективність розпізнавання на тестовому датасеті (mAP=99,3 %), але не забезпечили 100 % розпізнавання показань.

3. Для покращення результатів розпізнавання було створено алгоритм, за яким на другому етапі обробки виконувалось селектування обмежувальних рамок за горизонтальними та вертикальними координатами. Це дозволило видалити рамки, що відповідали помилковому розпізнаванню фрагментів фону, як показань лічильника, а також дозволило отримати дійсну послідовність цифр показань.

4. Також у алгоритм вторинної обробки результатів розпізнавання було додано аналіз локалізацій цифр після YOLOv10s за допомогою альтернативної розробленої загорткової нейронної мережі. Ця мережа також була навчена за допомогою створеного навчального датасету і показала високу ефективність розпізнавання (mAP=98,8 %). Результати сумісного розпізнавання обома нейронними мережами дозволили визначити ті зображення, що містять недостовірні випадки розпізнавання. Запропоновано алгоритм з повторенням фотографування та розпізнавання показань в таких випадках.

Список літератури:

1. Shelke S., Pathak I.S., Sangai A.P., Lunge D. A Review Paper on Computer Vision // International Journal of Advanced Research in Science Communication and Technology. Deli. 2023. Vol.3, Iss.2. P.1–4. DOI:10.48175/IJARST-8901
2. Dong S., Wang P., Abbas K. A survey on deep learning and its applications// Computer Science Review. Amsterdam. 2021. Vol.40, Iss.C. P.1–8. <https://doi.org/10.1016/j.cosrev.2021.100379>
3. Mahadevkar S.V. A Review on Machine Learning Styles in Computer Vision – Techniques and Future Directions // IEEE Access. Piscataway. 2022. Vol.10. P.1-37. DOI:10.1109/ACCESS.2022.3209825
4. Chai J., Zeng H., Li A., Ngai E.W.T. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. Machine Learning with Applications. Amsterdam. Vol. 6. 2021. P.1–13. DOI:10.1016/j.mlwa.2021.100134
5. Tarhib S., Tanha J., Imanzadeh S., Hassanzadeh S. Multi Model CNN Based Gas Meter Characters Recognition // 13th International Conference on Computer and Knowledge Engineering (ICCKE). IEEE: Mashhad, Iran. 2023. P.1–7. DOI:10.1109/ICCKE60553.2023.10326299
6. Wang J. A Study of The OCR Development History and Directions of Development // Highlights in Science Engineering and Technology. Milan. 2023. Vol.72. P.409–415. DOI:10.54097/bm665j77
7. Kaur R., Singh E.R. Image Filtering Techniques-A Review // International journal of advance research in science and engineering. Deli. 2017. Vol.6, Iss. 8. P.1–6. DOI:10.23956/ijarcsse/V7I6/0105
8. Hassam T., Muhammad S. K., Muhammad O. T. Performance Analysis and Comparison of Faster R-CNN, Mask R-CNN and ResNet50 for the Detection and Counting of Vehicles // 2021 International Conference on Computing, Communication, and Intelligent Systems, Greater Noida, India, 19–20 February 2021. Piscataway, NJ: IEEE, 2021. P. 587–594. DOI:10.1109/ICCCIS51004.2021.9397079

9. Mukaram S., Nizar Z., Mahmoud M. An Improved YOLOv8 to Detect Moving Objects // IEEE Access. Piscataway. 2023. Vol. 11. P. 1–25. DOI:10.1109/ACCESS.2024.3393835
10. Muhammad H. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection // Machines. 2023. Vol. 11, Iss. 7. P. 1–25. DOI:10.3390/machines11070677
11. Yixiao Z., Yuanming G., Xiaolong C. Research on YOLOv5 Vehicle Detection and Positioning System Based on Binocular Vision // World Electric Vehicle Journal. Basel. 2024. Vol. 15, Iss.2. P. 1–15. <https://doi.org/10.3390/wevj15020062>
12. Cao J. and oth. Front Vehicle Detection Algorithm for Smart Car Based on Improved SSD Model // Sensors. Basel. Vol. 20. 2020. P.1–21. DOI:10.3390/s20164646
13. Choi H.K., Paik C.K. Recurrent DETR: Transformer-based Object Detection for Crowded Scenes // IEEE Access. Piscataway. Vol. 4. 2023. P. 1–21. DOI:10.1109/ACCESS.2023.3293532
14. Zubkov O., Sheiko S., Oleynikov V., Kartashov V., Babkin S. Investigation Of The Yolov5 Algorithm Efficiency For Drone Recognition // Telecommunications and Radio Engineering. Redding. 2024. Vol.83, Iss.1. P.65–79. DOI:10.1615/TelecomRadEng.2023048987
15. Jung H.K., Choi G.S. Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions // Appl. Sci. Basel. 2022. Vol. 12. P.1–16. DOI: 10.3390/app12147255
16. Illia Oleksiienko, Dat Thanh Tran, Alexandros Iosifidis. Variational Neural Networks implementation in Pytorch and JAX // Software Impacts. Amsterdam. 2022. Vol.14. P.1–8. <https://doi.org/10.1016/j.simpa.2022.100431>
17. Zebin T., Scully P.J., Peek N., Casson A. Design and Implementation of a Convolutional Neural Network on an Edge Computing Smartphone for Human Activity Recognition // IEEE Access. Piscataway. Vol. 8. 2019. P. 1–11 DOI:10.1109/ACCESS.2019.2941836

Надійшла до редколегії 02.10.2024

Відомості про авторів:

Зубков Олег Вікторович – канд. техн. наук, професор, Харківський національний університет радіоелектроніки, завідувач кафедри мікропроцесорних технологій і систем, Україна; e-mail: oleh.zubkov@nure.ua; ORCID: <https://orcid.org/0000-0002-8528-6540>

Яковенко Олександр Сергійович – Харківський національний університет радіоелектроніки, магістр кафедри мікропроцесорних технологій і систем, Україна; e-mail: oleksandr.iakovenko@nure.ua

Старокожев Святослав Валерійович – Харківський національний університет радіоелектроніки, магістр кафедри мікропроцесорних технологій і систем, Україна; e-mail: sviatoslav.starokozhev@nure.ua

Скорбатюк Максим Володимирович – Харківський національний університет радіоелектроніки, магістр кафедри мікропроцесорних технологій і систем, Україна; e-mail: maksym.skorbatiuk@nure.ua