

УДК 004.032.26:004.93

ДОСЛІДЖЕННЯ ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ З ФУНКЦІЄЮ ЛОКАЛІЗАЦІЇ НА ПЛАТФОРМІ RASPBERRY PI

Аязов О.А.

email: oleksandr.aiazov@nure.ua

Харківський національний університет радіоелектроніки, каф. МІРЕС
м. Харків, Україна

The paper provides a review of computer vision methods that perform classification and localization of objects in an image. Neural network architectures have been selected for embedding in the Raspberry PI single-board computer. The performance of SSDLite and YOLOv8n was tested on two versions of Raspberry PI with the Google Coral hardware accelerator. Practical recommendations are given for choosing the image size at the input of a neural network and the type of network to ensure real time when processing video camera frames.

Вбудовані системи використовуються в різних сферах – від розумних будинків до автономного транспорту. Перспективним напрямом у вбудованих системах є технології комп'ютерного зору, що дають змогу пристроям аналізувати візуальну інформацію та приймати рішення. Камери відеоспостереження з алгоритмами розпізнавання об'єктів можуть автоматично виявляти підозрілі дії, розпізнавати обличчя відстежувати переміщення людей [1]. Головною проблемою ефективного вбудовування методів комп'ютерного зору в доступні за ціною апаратні платформи є обмеженість апаратних ресурсів, що не дозволяє швидко розпізнавати об'єкти та приймати рішення. Одноплатні комп'ютери, такі як Raspberry Pi [2], є компромісним рішенням, що поєднує доступну ціну та швидкодію.

Для вирішення завдань класифікації об'єктів у зображенні, а також локалізації їх положення використовують нейронні мережі глибокого навчання, що спроможні декласифікувати об'єкти. Найбільшого поширення останніми роками набули такі двопрхідні методи, як RCNN, Fast RCNN, Faster RCNN [3] і однопрхідні SSD (Single Shot Detector) [4], YOLO (you only look once) [5]. В основному існуючі архітектури нейронних мереж працюють на персональних комп'ютерах. Так для алгоритму YOLO існує 7 основних популярних версій з 3 до 10 і для кожної версії існує 4 модифікації архітектури s,m,l,x з кількістю параметрів. Без використання GPU час розпізнавання зображень на ПК з Core I7 одинадцятого покоління для архітектур s,m,l,x YOLOv8 [5] та розміру вхідного зображення 640 пікселів становить від 130 до 800 мс. Для алгоритму SSD час обробки зображень з аналогічною роздільною здатністю становить від 110 до 700 мс. Також слід враховувати, що кількість параметрів мережі YOLOV8 знаходиться в межах від 10 до 70 мільйонів і для запуску її роботи потрібна наявність 8 Гбайт оперативної пам'яті. Апаратні платформи типу Raspberry PI, Jetson

Nano, не мають такого обсягу оперативної пам'яті. Ядра CUDA для виконання паралельних обчислень має лише обмежена кількість платформ, таких як Jetson Nano. Тому запуск стандартних архітектур нейронних мереж на платформах Raspberry PI або Jetson Nano неможливо, або робота таких мереж буде надто повільна для виконання основного завдання автоматизованої системи. Тому створюються спрощені архітектури популярних нейронних мереж для вбудованих систем. Так, для YOLO існує спрощена модифікація n, яка має від 2 до 3.2 мільйонів параметрів і зменшує час обробки одного кадру до 38 до 60мс. MobileNet-SSDLite, що є спрощеною версією алгоритму SSD, показує час обробки кадру від 30 до 50мс. Саме ці версії нейронних мереж були використані для тестування на Raspberry PI. Метою роботи стало дослідження можливості детектування об'єктів у реальному часі на обмеженій апаратній платформі Raspberry PI.

Для проведення досліджень використовувалися 2 найбільш популярні моделі Raspberry PI: 3B+ з об'ємом оперативної пам'яті 1Гбайт, а також PI4B з об'ємом пам'яті 2Гбайт. Також був використаний USB-прискорювач Google Coral для виконання апаратних обчислень на спеціалізованому процесорі. До Raspberry PI була підключена відеокамера OV5647 з роздільною здатністю 5 MP через спеціалізований інтерфейс відеокамери.

На Raspberry PI було встановлено дві основні бібліотеки комп'ютерного зору: tensorflow і ultralytics. Ці бібліотеки використовувалися для завантаження моделей MobileNet-SSDLite та YOLO відповідно. Для доступу до відеокамери та захвату зображення застосовувалася вбудована в Raspbian бібліотека PiCamera2, для перетворення формату захоплених кадрів до формату бібліотеки ultralytics використовувалась додаткова бібліотека OpenCV. Відеозахват з відеокамери здійснювалося за допомогою методу capture_array() бібліотеки PiCamera2. Захват та розпізнавання виконувались при двох значеннях розміру вхідного зображення 320 та 640 пікселів. Вони представлені у вигляді таблиці 1

Таблиця 1 – Результати вимірювання часу обробки кадрів з камери у мережах MobileNet-SSDLite та YOLOv8n

	Середній час обробки кадра з камери, ms			
	Без прискорювача		З прискорювачем	
	Raspberry PI 3 B+			
Розмір кадра	SSDLite	YOLOv8n	SSDLite	YOLOv8n
320x320	730	540	68	81
640x640	2100	1620	314	390
	Raspberry PI 4 B			
320x320	510	540	32	65
640x640	1570	1620	227	286

При виконанні вимірювань розраховувався усереднений час обробки кадрів. Відхилення від середнього значення становило до $\pm 6\%$, що пов'язано з кількістю об'єктів, що розпізнаються.

Висновки. Апаратні ресурси Raspberry PI достатні для роботи сучасних спрощених архітектур нейронних мереж, які дозволяють класифікувати об'єкти на зображеннях та виконувати їх локалізацію у вигляді обмежувальних рамок. Найбільш доцільно обробляти зображення з роздільною здатністю 320x320 пікселів. У цьому випадку вдається досягти обробки трьох кадрів за секунду. Зі збільшенням розмірів зображення у 2 рази, час обробки збільшує у 3 рази. Такої швидкодії достатньо для систем безпеки, які мають лише виявити об'єкт у зоні огляду. Для роботів даної швидкодії недостатньо, оскільки вони повинні керувати маніпулятором та рухом у реальному масштабі часу. Застосування апаратних прискорювачів типу Google Coral вирішує цю проблему та дозволяє підвищити швидкість обробки зображення до 15-30 кадрів/с при розмірі зображення 320x320 пікселів. За рахунок наявності 128 ядер прискорювач дозволяє виконувати більшість обчислень паралельно із роботою центрального процесора. Однак прискорювач має USB3.0 інтерфейс. Тому при його підключенні до Raspberry PI 3 B+ не вдається досягти високої швидкості обробки за рахунок повільної передачі між прискорювачем і Raspberry PI. Спрощена архітектура алгоритму MobileNet-SSDLite дозволяє виконувати розпізнавання швидше у порівнянні з YOLOv8n. Тому можна її рекомендувати для практичного використання.

Список використаних джерел:

1. AA Syahidi, K. Kiyokawa, F. Okura. Computer Vision в Smart City Application: A Mapping Review. 6th IEEE International Conference on Applied Computational Intelligence in Information Systems. Bandar Seri Begawan, Brunei Darussalam. 2023. pp.253-267
2. M. Abdulhamid, O. Odondi, M. Al-Rawi. Комп'ютерна позиція базується на raspberry pi system. Applied Computer Science. Vol.16, No.4. pp.85-102
3. Хассам Т., Muhammad SK, Muhammad OT Performance Analysis і Comparison of Faster R-CNN, Mask R-CNN і ResNet50 для виявлення і 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.
4. J. Cao and oth. Front Vehicle Detection Algorithm для Smart Car Based на Improved SSD Model. Sensors.Vol. 20. 2020. pp.1-21
5. Mukaram S., Nizar Z., Mahmoud M. An Improved YOLOV8 до Detect Moving Objects. IEEE Access. 2023. Vol. 11. P. 1-25.