УДК 004.93'1:629.7.014-519(045) DOI: 10.18372/2073-4751.77.18658

Лукаш Ю.В.,
orcid.org/0009-0003-1824-8936,
e-mail: 8391003@stud.nau.edu.ua

АНАЛІЗ ПРОДУКТИВНОСТІ АЛГОРИТМУ ДЕТЕКЦІЇ ОБ'ЄКТІВ YOLOv8n НА МІКРОКОМП'ЮТЕРАХ RaspberyPi TA NVidia Jetson Nano

Національний авіаційний університет

Вступ

В останні роки набув поширення розвиток штучного інтелекту, зокрема і розпізнавання образів та детекція об'єктів. Ці технології допомагають пришвидшити та полегшити аналіз зображень та відеопотоків. З сучасними обчислювальними можливостями розпізнавання образів можливе навіть в реальному часі.

З розвитком сфери розпізнавання образів – цю технологію почали використовувати для автоматизації певних процесів в самих різноманітних сферах.

Цю технологію використовують у військовій сфері, у аграрний сфері [6], безпековій, зокрема для аналізу та гасіння лісових пожеж [7], у сфері медицини. Також і в цивільній сфері на базі алгоритмів розпізнавання образів впроваджують різні функції для автомобілів та БПЛА. В тому числі розроблюють автопілоти, системи, які приймають самостійно рішення по керуванню на базі алгоритмів розпізнавання образів.

Дане дослідження актуальне в рамках розробки комплексу для автопілоту БПЛА, який базується на результаті обробки оптичного відеопотоку. Який в залежності від обставин та оброблених зображень самостійно приймає рішення по керуванню самим БПЛА для вирішення цільових задач.

Мета дослідження

Метою дослідження є розгляд можливих архітектурних побудов БПЛА з обробкою відеопотоку та прийняттям рішень по керуванню БПЛА. Зокрема варіанту повної автономії літального апарату без втручання оператора.

Задачею являється проведення експерименту з запуску однієї з популярних сучасних моделей розпізнавання та детекції образів *YOLOv8n* на недорогих одноплатних комп'ютерах.

Зробити аналіз і оцінку можливості та доцільності використання таких комп'ютерів безпосередньо на борту ЛА, базуючись на отриманих експериментальних даних.

Виявити можливості по покращенню швидкодії. Цільова швидкість – успішне розпізнавання об'єктів на відео в реальному часі.

Основна частина

Останні роки технології в БПЛА розвиваються дуже швидко. Існує величезна кількість різноманітних БПЛА, які побудовані для вирішення різноманітних задач, мають різну будову, різний запас по автономності, вантажопідйомності, вартості. Їх класифікують за абсолютно різними ознаками та параметрами [5].

Розглянуто БПЛА, який має бути відносно недорогим для масового виробництва але водночас мати обчислювальні можливості для розпізнавання образів з відеопотоку зі швидкістю реального часу для побудови автопілота на борту, що приймає рішення на базі розпізнавання.

Високорівнево архітектуру такого комплексу запропоновано розбити на 3 складові: камера, яка знімає відео, модуль обробки інформації – тобто розпізнавання об'єктів, модуль прийняття рішень на основі розпізнаних об'єктів і формування команд керування до БПЛА та сам польотний контролер, який отримує команди та змінює напрямок руху літального об'єкту.

З цих трьох складових – камера та польотний контролер завжди знаходяться безпосередньо на апараті, а от частина, яка відповідає за обробку даних та прийняття

рішень – може бути вирішена наступним чином.

1) ГраундСтейшен, або наземна станція – комп'ютер, ноутбук чи просто потужний блок керування, який знаходиться на землі у оператора БПЛА. В такому випадку – відеопотік з БПЛА відправляється на наземну станцію, локально проводяться всі обрахунки, детекції та прийняття рішень по командам управління – і відправляється назад сформовані команди управління та керування. Схематично архітектура зображена на рис. 1.

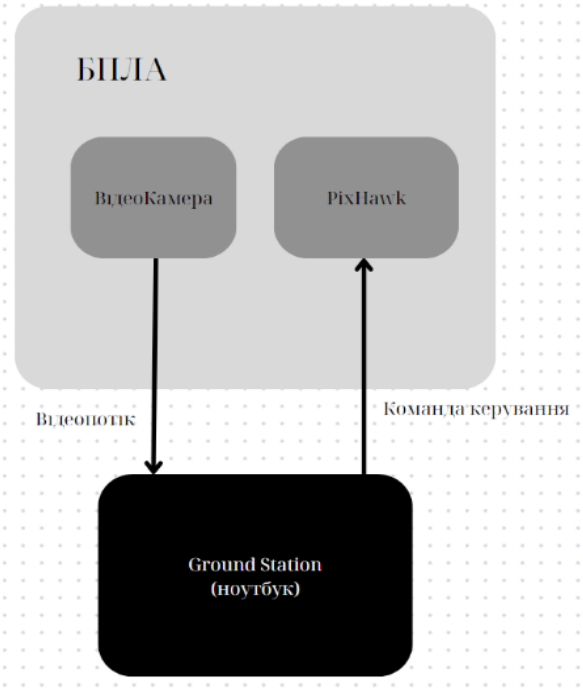

Рис. 1. Архітектура з обробкою відео на наземній станції

2) Клауд компьютинг, зображений на рис. 2 – якщо на самому БПЛА забезпечити стійкий зв'язок з мережею інтернет, що цілком можливо в умовах міста, де дрони виконують задачі, наприклад, доставки – тоді відправка відео може здійснюватися в *Cloud*, проводитися розрахунки та відправлятися команди керування назад. Викликом у такому підході може стати задача серіалізації та десеріалізації команд в протоколи зрозумілі для польотного контролера, і тоді може з'явитися необхідність мікрокомп'ютера на БПЛА, що буде забезпечувати комунікацію з мережею та перетворення прийнятих рішень в *Cloud*-середовищі у протокол, наприклад *MAVLink* і його пряму відправку в польотний контролер. Також в такому підході треба враховувати затримки на передачу даних та синхронізацію.

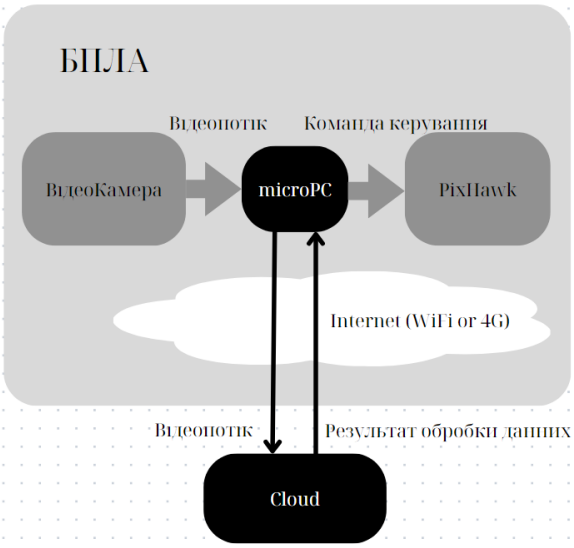

Рис. 2. Архітектура з обробкою відео в *Cloud*

3) Розміщення процесорних потужностей, одноплатного комп'ютера на самому БПЛА (рис. 3). В такому підході з'являються обмеження по вазі, потужності, живленню мікрокомп'ютера. Проте – цей варіант дозволяє реалізувати повну автономність, коли є небезпека відсутності каналів зв'язку між ЛА та оператором, проблем на певних частотах. Також вирішується питання затримок на передачу сигналу в дві сторони.

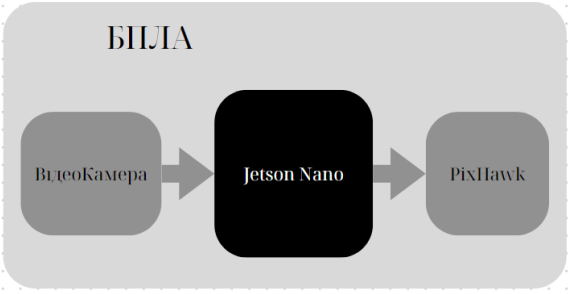

Рис. 3. Архітектура з обробкою відео у мікрокомп'ютері на БПЛА

Архітектура автопілота

Задачами автопілотування на базі результатів аналізу оптичного пристрою можуть бути наступні елементи: слідування за об'єктом, зближення з об'єктом, корегування руху у відповідності до навігаційних символів на землі.

Для вирішуванні перелічених задач запропоновано таку архітектуру програми (рис. 4), яка відповідає за аналіз відеопотоку в реальному часі та формує команди керування. Саме таке розбиття архітектури зроблено для подальшого аналізу швидкодії кроків обробки даних , щоб було можливо оцінити доцільність використання конкретних моделей мікрокомп'ютерів на борту БПЛА.

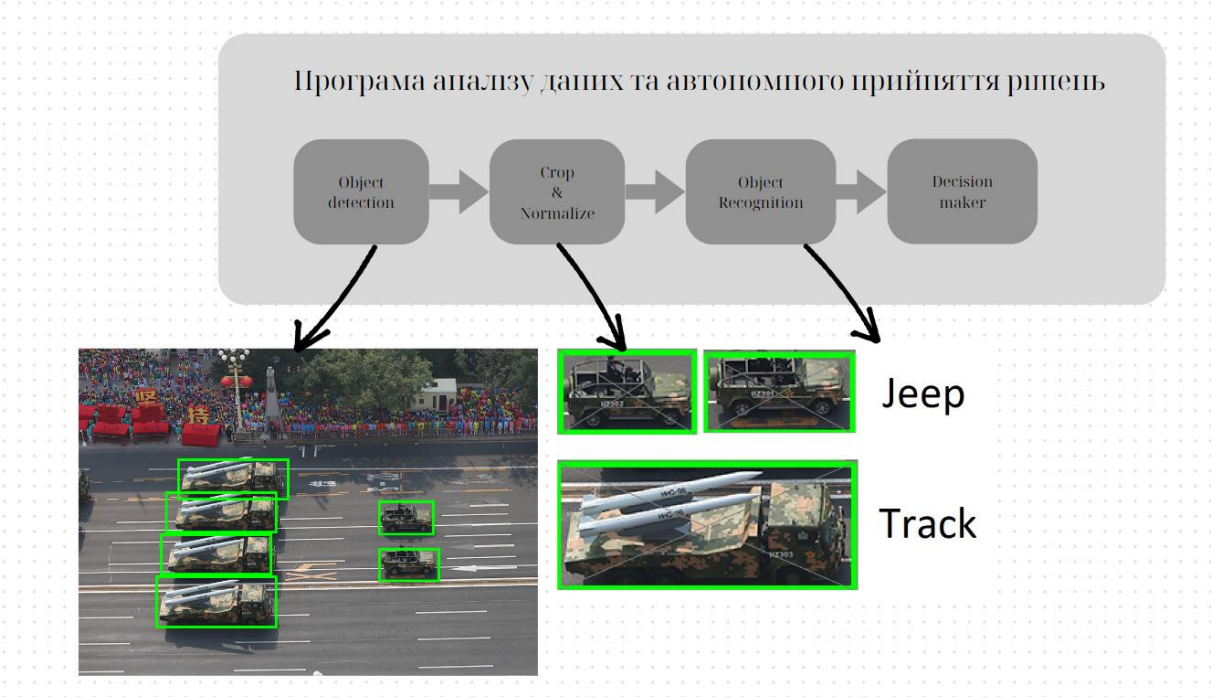

Рис. 4. Архітектура програми, що реалізує автономне прийняття рішень керування БПЛА

У запропонованій архітектурі програми, що наведено вище відносно швидкими етапами можна вважати *Crop & Normalize* та *Decision Maker* етапи. Водночас затратними за ресурсами та часу виконання будуть *Object Detection* та *Object Recognition*.

Для етапу *Object detection* існує досить багато різних алгоритмів, які можна використовувати. Для використання під конкретну задачу, звичайно, необхідно проводити додаткове навчання моделі на ретельно підібраному датасеті з класами об'єктів, які нас цікавлять. Проте, навчання здебільшого впливатиме на якість детекцій, на показники точності та похибки. А задачею цієї роботи є дослідити швидкодію, наскільки взагалі доцільно використовувати загальноприйняті алгоритми для детекції об'єктів в реальному часі в умовах обмежених ресурсів, потужності, що може бути розміщена на самому літальному апарату.

Для детекції можна використовувати такі відомі алгоритми як:

- *YOLOv*8 (*You Look Only Once*)
- *PaddleDetection*
- *mmdetection*
- *detectron*2
- *RetinaNet*
- *Viola-Jones Detector*
- *SIFT*
- *HOG Detector*

Вище наведені і сучасніші алгоритми і більш давні. Кожен з них має свої переваги та недоліки, зокрема щодо швидкодії та точності.

Сьогодні можна знайти роботи по оцінці швидкості роботи попередніх версій *Yolo* (*v*3, *v*4) на пристроях з обмеженим ресурсах, які зачасту використовуються в *IoT* задачах, робототехніці, БПЛА [1].

В даному експерименті випробована швидкодія роботи новішої версії з популярних моделей *Yolo* для *object detection*, полегшеної версії, а саме *YOLOv8n* – це найлегша модель серед *Yolo v8*, то ж очікування, що вона потенційно може працювати на одноплатному комп'ютері з прийнятною швидкістю.

YOLO (*You Only Looks Once*) – це особливий підхід по роботі з детекцією об'єктів на зображенні, в основі якого лежить аналіз всього зображення за раз, а не проходження ковзним вікном і окремий багаторазовий аналіз частини зображення. Перша версія цієї моделі була представлена у 2016 році [4] і з тих пір було багато покращень та модифікацій. Архітектура моделі доступна дослідникам, тому розвиток став стрімким. То ж якщо ще 3-5 років тому була актуальною *Yolo-v3-tiny*, детальні заміри по її роботі доступні в [1], то в 2023 році актуальною покращеною версією вже була *Yolo-v8*.

Щодо одноплатних комп'ютерів, які можуть бути поміщені на БПЛА невеликого розміру та за помірною ціною для можливого масового виробництва – в дослідженні розглянуто *RaspberryPi* та *NVidiaJetson Nano* мікрокомп'ютери.

Для таких задач нерідко порівнюють потужність та швидкодію наведених вище одноплатних комп'ютерів. Наприклад в роботі [2] проводився порівняльний аналіз для цих комп'ютерів, проте для інших задач, запускалися інші моделі штучного інтелекту. То ж на отримані результати можна спиратися для базового розуміння, проте в роботі проведено заміри для інших параметрів, корисних для задач автопілотування.

На рис. 5 зображено зовнішній вигляд розглянутих в експерименті одноплатних комп'ютерів. А в табл. 1 наведено порівняння їх характеристик.

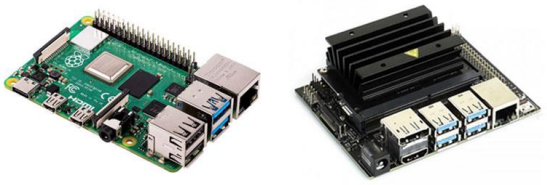

Рис. 5. *Raspberry Pi VS Jetson Nano*

Таблиця 1. Порівняння характеристик Jetson Nano та Raspberry Pi

Features	***Jetson Nano 4GB***	***Raspberry Pi 4 B***
CPU	*ARM Cortex-A57 (quad-core) @ 1.43GHz*	*Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz*
GPU	*128-core NVIDIA Maxwell @ 921MHz*	*Broadcom VideoCore VI*
DL	*NVIDIA GPU support (CUDA, cuDNN, TensorRT)*	
Memory	*4GB 64-bit LPDDR4 @ 1600MHz | 25.6 GB/s*	*2GB, 4GB or 8GB LPDDR4-3200 SDRAM with on-die ECC*
Vision	*NVIDIA GPU support (CUDA, VisionWorks, OpenCV)*	*OpenGL ES 3.0 Graphics*
Encoder	*4Kp30 | 4x 1080p30 | 9x 720p30 (H.264/H.265)*	*H.264 (1080p30)*
Decoder	*4Kp60 | 2x 4Kp30 | 8x 1080p30 | 18x 720p30 (H.264/H.265)*	*H.265 (4Kp60) | H.264 (1080p60)*
Camera	*2x MIPI CSI-2 connectors*	*1× MIPI CSI Connector*
Display	*HDMI and DP*	*2× micro-HDMI (4kp60) | 1× MIPI DSI*
USB	*4x USB 3.0 A | 1x USB 2.0 Micro-B*	*2x USB 3.0 A | 2x USB 2.0 A*
Misc IO	*UART, SPI, I2C, I2S, GPIOs*	*UART, SPI, I2C, GPIO*
Power	*5V / 5-10W*	*5V / 10-15W*

Очевидно, що *Jetson Nano* значно перевершує *RaspberryPi*, проте вартість *RaspberryPi* на сьогодні в 3-4 рази менша. А це важливий фактор, коли ми говоримо про недорогі БПЛА, або навіть БПЛА, які допустимо втрачати.

Опис експериментальної частини

Для отримання достовірних результатів експерименту – на кожному одноплатному комп'ютері для певної розмірності було обрано по 3 тестових відео і проведено по 5 запусків на кожне відео. Результати отримані від 5 запусків усереднено. Значення для різних відео – також потім усереднено для отримання більш достовірних результатів та мінімізації ефекту випадковості.

Отже, для порівняння швидкодії – обрані відео розмірністю 360x360 та 480x480, модель *YOLOv8n* та написана програма на мові програмування *Python* з використанням бібліотеки *ultralytics*.

Таблиця 2. Результати експерименту

Video		*Raspberry Pi* 4		*Jetson Nano*	
Video-name	*Frame count*	Загальний час	*FPS*	Загальний час	*FPS*
*Video*1	734	278 сек	2.64	38 сек	19.47
*Video*2	842	300 сек	2.81	42 сек	20.11
*Video*3	503	184 сек	2.73	26 сек	19.35
Результат 360x360			**2.73**		**19.64**
*Video*4	489	264 сек	1.85	46 сек	10.56
*Video*5	687	350 сек	1.96	61 сек	11.20
*Video*6	711	406 сек	1.75	67 сек	10.65
Результат 480x480			**1.85**		**10.80**

Слід зазначити, що для моделі було проведено додаткове навчання для специфічних класів об'єктів. Проте, це впливає на точність визначення, а метою цього експерименту є дослідити саме швидкодію.

У табл. 2 наведені результати запусків по окремим відео Характеристики обраних відео також наведено у таблиці. Зокрема обрані розмірності 360x360 та 480x480. Для кожної розмірності – підібрані відео різної довжини. Це мало б знівеліювати піки у швидкодії рооті алгоритму чи інші аномалії під час досліду. Отримані показники в таблиці усереднені. Підсумкові значення також усереднені для двох розглянутих розмірностей відео.

Як видно з отриманих результатів – використання алгоритму *YOLO* в чистому вигляді на *Raspberry Pi* для задач визначення об'єктів в реальному часі є неефективним. Потужності процесора вистачає всього на 2 кадри в секунду. А цього недостатньо для впевненості в визначеному результаті та своєчасному прийнятті рішень. Отже, треба обирати більш потужний комп'ютер, або шукати простіші архітектури. Наприклад запуск на БПЛА лише алгоритму трекінгу. Він є швидшим, а задача визначення та захоплення цілі має виконуватися оператором.

Щодо *Jetson Nano* – показники отримані в результаті проведеного експерименту значно кращі. Швидкодія до 20 кадрів на секунду без оптимізацій для *GPU* модулю, який є суттєвою перевагою всіх одноплатних комп'ютерів від *NVidia* – це гарний показник, який дозволить з високою швидкістю аналізувати потокове відео з камери. На базі цього треба досліджувати етап прийняття рішень по керуванню БПЛА.

Компанія *NVidia* пропонує широку лінійку одноплатних комп'ютерів, і безумовно можна обрати більш потужний

комп'ютер. Порівняння інших, потужніших моделей можна знайти в статті [1]. Проте варто враховувати, що в задачі яка розглядається є фокус на мінімально достатньому одноплатному комп'ютері, тому що суттєвим критерієм при виборі є ціна готового БПЛА з комплексом автономного керування на борту.

Отже подальші дослідження доцільно проводити на однопалатному комп'ютері *Jetson Nano* від *NVidia*. Хоча, як вбачається, навіть на *Jetson Nano* є необхідність пропускати кадри у відеопотоку, а отже є потреба в покращенні алгоритмів, або більш детального вибору алгоритму під конкретну цільову задачу.

Висновки

Розглянуто різні підходи до побудови комплексу для БПЛА, що міг би забезпечити автопілотування ЛА на базі аналізу відеопотоку з оптичного каналу. Для архітектури, в якій комп'ютер має знаходитися безпосередньо на БПЛА проведено експеримент по оцінці потужностей одноплатних комп'ютерів від різних виробників. За результатами експерименту видно, що для описаних задач можна продовжувати використання *Jetson Nano*. Проте одноплатні мікрокомп'ютери невисокого цінового діапазону мають обмежені потужності і потребують суттєвої уваги різних способах оптимізації. Такими оптимізаціями, що можуть бути предметом розгляду в майбутньому – можна виділити:

- компресія вхідного зображення, щоб на вхід алгоритму потрапляли кадри меншого розміру
- використання одноканального, некольорового зображення. Це може бути від тепловізійної або інфрачервоної камери, або препроцесинг
- активне використання *NVidia GPU* модуля і специфічні оптимізації для нього
- більш детального вибору алгоритмів комп'ютерного зору для конкретних цільових задач
- комбінування різних алгоритмів і їх динамічну заміну в залежності від умов в яких знаходиться апарат

Для подальших досліджень в побудові комплексу БПЛА з елементами автопілотування доцільно сфокусуватися саме на обраному *Jetson Nano* від *NVidia*.

Література

1. Feng H. et al. Benchmark analysis of yolo performance on edge intelligence devices. *Cryptography*. 2022. Vol. 6, iss. 2. 16. DOI: 10.3390/cryptography6020016.

2. Suzen A. A., Duman B., Sen B. Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN. *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* : proceedings, Ankara, Turkey, 26-28 June 2020 / IEEE. Danvers, 2020. 5 p. DOI: 10.1109/HORA49412.2020.9152915.

3. Pathak A. R., Pandey M., Rautaray S. Application of Deep Learning for Object Detection. *Procedia Computer Science.* 2018. Vol. 132, Iss. 6. P. 1706–1717. DOI: 10.1016/j.procs.2018.05.144.

4. Redmon J. et al. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* : proceedings, Las Vegas, NV, USA, 27-30 June 2016 / IEEE. Danvers, 2016. P. 779–788. DOI: 10.1109/CVPR.2016.91.

5. Mohsan S. A. H. et al. Unmanned aerial vehicles (UAVs): practical aspects, applications, open challenges, security issues, and future trends. *Intelligent Service Robotics.* 2023. Vol. 16, iss. 1. P. 109–137. DOI: 10.1007/s11370-022-00452-4.

6. Radoglou-Grammatikis P. et al. A compilation of UAV applications for precision agriculture. *Computer Networks.* 2020. Vol. 172. 107148. DOI: 10.1016/j.comnet.2020.107148.

7. Akhloufi M. A., Couturier A., Castro N. A. Unmanned aerial vehicles for wildland fires: Sensing, perception, cooperation and assistance. *Drones.* 2021. Vol. 5, Iss. 1. 15. DOI: 10.3390/drones5010015.

Лукаш Ю.В.

АНАЛІЗ ПРОДУКТИВНОСТІ АЛГОРИТМУ ДЕТЕКЦІЇ ОБ'ЄКТІВ YOLOv8n НА МІКРОКОМП'ЮТЕРАХ RaspberyPi ТА NVidia Jetson Nano

У науковій статті описані підходи для реалізації комплексу автономного управління на БПЛА. Зокрема виділено архітектуру з використанням одноплатного комп'ютера, що може бути встановлений безпосередньо на літальний апарат. Для такої архітектури побудови комплексу обрано потенційні одноплатні комп'ютери та досліджено їх швидкодію. Для оцінки описано та проведено експеримент з запуску алгоритму розпізнавання образів на базі нейромережі Yolo v8 nano. В статі наведені результати роботи алгоритму. На основі проведеного експерименту визначено, що для подібних алгоритмів доцільно використовувати Jetson Nano. А RaspberryPi не володіє достатньою потужністю для конкретної задачі з обраним алгоритмом.

Ключові слова: *БПЛА; розпізнавання образів; розпізнавання зображень; детекція об'єктів; нейромережа; YOLO; RaspberryPi; Jetson Nano; автопілот; автономне керування; одноплатний комп'ютер.*

Lukash Y.V.

PERFORMANCE ANALYSIS OF YOLOV8N OBJECT DETECTION ALGORITHM ON RASPBERRYPI AND NVIDIA JETSON NANO MICROCOMPUTERS

The scientific article describes approaches to the implementation of a complex of autonomous control of UAVs. In particular, the architecture with the use of a single-board computer that can be installed directly on the aircraft is highlighted. Potential single-board computers were selected for such an architecture of the complex construction and their performance was investigated. For evaluation, an experiment was described and conducted to run the pattern recognition algorithm based on the Yolo v8 nano neural network. The results of the algorithm are given in the article. Based on the conducted experiment, it was determined that it is advisable to use Jetson Nano for similar algorithms. And RaspberryPi does not have enough power for a specific task with the selected algorithm.

Keywords: *UAV; pattern recognition; image recognition; object detection; neural network; YOLO; RaspberryPi; Jetson Nano; autopilot; autonomous control; single board computer.*