

УДК 004.93

Довгаль Д.Ф.¹, Миронова Н.О.², Пархоменко А.В.²

¹ студ. гр. КН-113м НУ «Запорізька політехніка»

² доц. НУ «Запорізька політехніка»

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ВИЗНАЧЕННЯ ОБ'ЄКТІВ ДЛЯ НАЗЕМНОГО ДРОНУ

Актуальність роботи обумовлена зростаючим попитом на автономні рішення для визначення об'єктів в реальному часі, що забезпечують економічність та функціональність. Використання доступних мікроконтролерів з алгоритмами комп'ютерного зору дозволяє створювати компактні, бюджетні системи для застосувань у промисловості, безпеці, моніторингу навколишнього середовища та інших сферах. Такий підхід є перспективним для впровадження в робототехнічних системах, де необхідно зменшити навантаження на обчислювальні ресурси й одночасно забезпечити високу мобільність і точність роботи.

Мета роботи – розробити прототип системи визначення об'єктів для наземного дрону, що дозволяє ідентифікувати об'єкти у реальному часі з використанням алгоритмів комп'ютерного зору.

У рамках проектування системи визначення об'єктів для наземного дрону, було виконано порівняння мікроконтролерів, які мають функцію відео камери та можливість дистанційного керування, що дозволить вибрати оптимальний мікроконтролер для реалізації проекту. Були розглянуті наступні рішення: ESP32-CAM, Raspberry Pi, Arduino Nano RP2040 Connect, NVIDIA Jetson Nano. Було виявлено, що вибір мікроконтролера залежить від специфічних вимог проекту, таких як необхідна продуктивність, якість камери, можливість бездротового з'єднання та область застосування. Для простих проектів, наприклад відеоспостереження, підходить ESP32-CAM, тоді як для складніших завдань, пов'язаних із штучним інтелектом, краще використовувати Raspberry Pi 4 або Jetson Nano. Для реалізації прототипу системи було вирішено обрати ESP32-CAM, оскільки цей мікроконтролер є оптимальним вибором для бюджетних проектів, які потребують базової обробки зображень та функцій віддаленого керування.

Для визначення та розпізнавання об'єктів ESP32-CAM можна використовувати наступні технології, враховуючі можливості та обмеження цієї плати через невеликий обсяг пам'яті і процесорної потужності: TensorFlow Lite Micro, ESP-WHO, каскадні класифікатори Naag,

TensorFlow Lite Micro дозволяє запускати моделі глибокого навчання на мікроконтролерах; підходить для завдань класифікації та розпізнавання простих об'єктів. Модель має бути квантована для зменшення розміру та використання пам'яті. Підходить для складніших завдань, але вимагає спрощення моделі для оптимальної роботи на ESP32-CAM.

ESP-WHO – бібліотека від Espressif для виявлення та розпізнавання осіб, що спеціально оптимізована для ESP32-CAM. Дозволяє додавати особи до бази даних для ідентифікації. Легко використовувати для систем відеоспостереження та Розумного будинку, але обмежена завданням розпізнавання облич.

Каскадні класифікатори Naag, що можуть виявляти об'єкти на основі контурів. Використання Naag-каскадів потребує інтеграції бібліотеки OpenCV або спеціалізованих збірок, що можливо, але обмежено ESP32. Застосовні для базових завдань детекції, але не універсальні щодо всіх типів об'єктів.

YOLO (You Only Look Once) – існують полегшені версії YOLO, наприклад Tiny YOLO, або спеціальні модифікації, які можна адаптувати для ESP32-CAM. Потребує оптимізації та квантизації моделі, але може забезпечити гарну продуктивність на обмежених пристроях. Продуктивність та точність YOLO на ESP32-CAM будуть обмеженими. Ці моделі часто використовують для класифікації зображень з низькою роздільною здатністю та споживають менше пам'яті. Ці методи можуть комбінуватися в залежності від задачі, що дозволяє реалізувати ефективне розпізнавання на малопотужних пристроях, таких як ESP32-CAM.

В результаті роботи розроблено прототип системи визначення об'єктів для наземного дрону (Рис.1) на основі мікроконтролера ESP32-CAM, що дозволяє ідентифікувати об'єкти в реальному часі з використанням алгоритмів комп'ютерного зору та програмне забезпечення для ESP32-CAM, яке реалізує обробку зображень та алгоритми машинного навчання.



Рисунок 1 – Прототип наземного дрону