УДК 004.93 http://doi.org/10.15407/pp2024.04.014

Ю.М. Бердник, А.О. Скотаренко

ДО ПРОБЛЕМИ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ПРИСТРОЯХ З ОБМЕЖЕНИМИ РЕСУРСАМИ

Розглянуто проблему реалізації розпізнавання об'єктів на мікроконтролерах з обмеженими ресурсами. Зокрема, виклики, пов'язані з їхньою обчислювальною потужністю, обсягом пам'яті та енергоспоживанням. Запропоновано використання компактної архітектури нейронної мережі MobileNet, яка базується на алгоритмі зі зниженим обчислювальним навантаженням і забезпечує достатню продуктивність у середовищах із обмеженими ресурсами. Такий підхід дає змогу виконувати задачі класифікації зображень на недорогих мікроконтролерах. Хоча технології, які розглядаються, є відомими самостійно, їхнє комплексне застосування для конкретних задач класифікації на мікроконтролерах, таких як розпізнавання об'єктів, залишається мало дослідженим. У статті наведено детальний опис усіх етапів розробки: підготовки даних, налаштування параметрів моделі, а також застосування спеціалізованих методів масштабування зображень для зменшення обчислювального навантаження. Практичну частину статті присвячено вирішенню прикладної задачі розроблення системи розпізнавання полуниць із визначенням ступеня їхньої стиглості на основі мікроконтролера ESP32. Результати дослідження демонструють ефективність підходу для різних прикладних задач і підтверджують можливість інтеграції технологій комп'ютерного зору у пристрої з обмеженими ресурсами. Загалом робота доводить, що сучасні технології машинного навчання стають доступними навіть для найменш потужних апаратних платформ, розширюючи їхні можливості та сфери використання.
Ключові слова: розпізнавання об'єктів, класифікація, нейронна мережа, архітектура MobileNet, мікроконтролер.

Y.M. Berdnyk, A.O. Skotarenko

ON THE PROBLEM OF OBJECT RECOGNITION ON DEVICES WITH LIMITED RESOURCES

The problem of implementing object recognition on microcontrollers with limited resources, particularly the challenges related to their computational power, memory capacity, and energy consumption is addressesed. The use of the compact MobileNet neural network architecture is proposed, which is based on an algorithm with reduced computational load and ensures sufficient performance in resource-constrained environments. This approach enables image classification tasks to be performed on low-cost microcontrollers. Although the technologies discussed are well-known individually, their comprehensive application for specific classification tasks on microcontrollers, such as object recognition, remains underexplored. The article provides a detailed description of all development stages, including data preparation, model parameter tuning, and the use of specialized image scaling methods to reduce computational load. The practical part of the article is devoted to developing a strawberry recognition system to determine their ripeness level using the ESP32 microcontroller. The research results demonstrate the effectiveness of the approach for different applied tasks and confirm the feasibility of integrating computer vision technologies into resource-limited devices. Overall, the work proves that modern machine learning technologies are becoming accessible even to the least powerful hardware platforms, expanding their capabilities and areas of application..
Key words: object recognition, classification, neural network, MobileNet architecture, microcontroller.

Вступ

У сучасних реаліях нейронні мережі відіграють ключову роль у розвитку технологій. Розпізнавання об'єктів є важливою функцією у сучасних IoT-застосунках, автономних системах і робототехніці. Успішна інтеграція цієї технології дозволяє автоматизувати виявлення, класифікацію та відстеження об'єктів у реальному часі. Однак інтеграція моделей нейронних мереж на пристроях із обмеженим обсягом пам'яті та обчислювальними ресурсами може стикатися з певними труднощами.

Існує два основні підходи до розпізнавання об'єктів залежно від використовуваних типів датчиків: системи з візуалізацією навколишнього середовища (каме-

ри/відео) та системи без візуалізації навколишнього середовища (ІЧ-датчики, лазерні, ультразвукові, Bluetooth або GPS) [1]. У цій статті розглянуто візуальне розпізнавання об’єктів на мікроконтролерах з обмеженими ресурсами.

Дана стаття пропонує підхід до вирішення проблеми реалізації розпізнавання об'єктів на пристроях із обмеженими ресурсами за допомогою технології MobileNet — компактної архітектури нейронної мережі, оптимізованої для роботи на пристроях із низькою продуктивністю. Для спрощення розробки використовується платформа Edge Impulse, яка дозволяє автоматизувати тренування, оптимізацію та розгортання моделей на мікроконтролерах. Обраний підхід демонструє, як можна подолати основні обмеження апаратного забезпечення, зберігши прийнятну точність і швидкість обробки.

Попри те, що кожна з цих технологій є самостійно відомою, їхнє поєднання для вирішення конкретної задачі класифікації зображень на пристроях із обмеженим апаратним потенціалом, таких як IoT-системи, є рідкісним у літературі.

Стаття може бути корисною для інженерів і розробників, оскільки включає детальний опис процесу підготовки даних, налаштування параметрів моделей і використання конкретних методів масштабування зображень.

Описаний підхід також підкріплюється прикладом вирішення прикладної задачі. Такий приклад не лише ілюструє ефективність підходу, а й демонструє його прикладну цінність для вузькоспеціалізованих завдань.

Основна мета — продемонструвати, що навіть у середовищах з обмеженими ресурсами можна реалізувати ефективні рішення для розпізнавання об'єктів.

1. Вибір способу реалізації розпізнавання об’єктів на мікроконтролері

Обмежені ресурси мікроконтролерів створюють багато технічних викликів, які змушують розробників шукати компроміс між продуктивністю та точністю. Насамперед мікроконтролери, які мають дуже обмежений обсяг оперативної та постійної пам’яті. Типовий обсяг оперативної пам’яті у найпродуктивніших чіпах становить лише кілька мегабайтів [2], тоді як сучасні моделі для розпізнавання об’єктів, такі як MobileNet або YOLO, вимагають десятків чи навіть сотень мегабайтів для зберігання ваги та виконання обчислень. Через це навіть компактні моделі важко завантажити чи виконувати на таких пристроях, не кажучи вже про обробку зображень у реальному часі. Додатково низька тактова частота мікроконтролерів (зазвичай до 240 МГц) та відсутність спеціалізованих обчислювальних блоків, таких як GPU або TPU, призводять до тривалого часу виконання завдань навіть для невеликих моделей [2]. Це особливо критично для завдань у реальному часі, де навіть затримка в кілька секунд може бути неприйнятною.

Ще одним обмеженням є низьке енергоспоживання, яке є ключовим для автономних пристроїв. Розпізнавання об’єктів є завданням з високою обчислювальною складністю, що може значно зменшити тривалість роботи від батареї. Використання навіть оптимізованих моделей може швидко розрядити пристрій, особливо якщо задачі виконуються постійно. Крім того, обмежена пропускна здатність введення/виведення, характерна для мікроконтролерів, уповільнює передачу даних із сенсорів, таких як камери, до процесора, додаючи затримки до загального процесу розпізнавання.

Існує кілька підходів до вирішення цього завдання, кожен з яких має свої переваги і недоліки.

Одним із популярних підходів є серверна обробка, за якої мікроконтролер використовується лише для збору даних, таких як зображення, а всі обчислення виконуються на потужному сервері або в хмарному середовищі. Це дозволяє застосовувати складні моделі глибокого навчання, які забезпечують високу точність і можливість обробки великих обсягів даних. Однак серверна обробка має серйозні обмеження. По-перше, вона вимагає стабільного підключення до мережі, що не завжди можливо в автономних системах, як-от, у відда-

лених регіонах або в пристроях, які працюють у польових умовах. По-друге, передача даних на сервер і назад може викликати значні затримки, що робить цей підхід непридатним для завдань реального часу, таких як виявлення небезпечних ситуацій у системах безпеки чи управління робототехнікою [3]. Нарешті підтримка серверної інфраструктури чи оплата хмарних послуг може значно збільшити витрати, що не завжди виправдано у випадку компактних і недорогих пристроїв.

Іншим підходом є використання додаткових апаратних рішень, таких як спеціалізовані модулі для прискорення обчислень. Ці пристрої дозволяють виконувати складні нейронні мережі на периферійних пристроях, забезпечуючи високу швидкість і точність обробки даних. Вони особливо ефективні для роботи у реальному часі та автономних системах. Проте цей підхід також має недоліки. Використання додаткового обладнання збільшує загальну вартість системи, що може бути неприйнятним для недорогих пристроїв або масового виробництва. Крім того, інтеграція таких рішень вимагає додаткових ресурсів для налаштування, розробки та тестування. Апаратні модулі також збільшують струм споживання та розміри пристрою, що може бути критичним для компактних пристроїв, таких як дрони чи переносні пристрої. Енергоспоживання цих рішень, хоча й оптимізоване, все одно часто перевищує можливості стандартних акумуляторів, примушуючи шукати компроміс між розмірами батареї і часом автономної роботи.

Третій підхід — використання оптимізованих моделей глибокого навчання, таких як MobileNet, Tiny-YOLO чи SqueezeNet. Ці архітектури спеціально розроблені для пристроїв із невеликими ресурсами для виявлення об'єктів з хорошою продуктивністю в реальному часі та високою точністю [4, 5, 6]. Основними перевагами цього підходу є можливість обробки даних безпосередньо на пристрої, що усуває залежність від підключення до мережі та наявності затримки.

MobileNet — це спрощена архітектура, яка використовує згортки, що розділяються по глибині, для побудови легких глибоких згорткових нейронних мереж і забезпечує ефективну модель для мобільних і вбудованих систем [7].

Структура MobileNet базується на фільтрах, що розділяються по глибині, як показано на рис. 1.

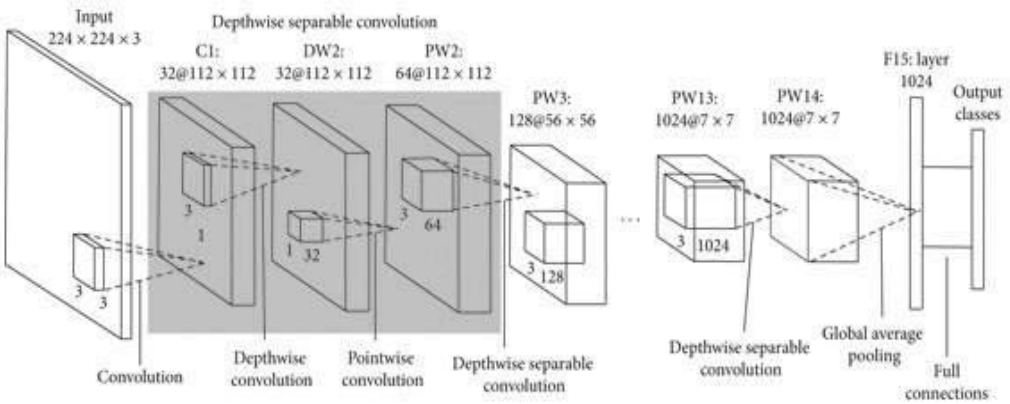

Рис. 1. Структура нейронної мережі MobileNet [8]

MobileNetV2 є значним удосконаленням у порівнянні з MobileNetV1 і впроваджує найсучасніші технології мобільного візуального розпізнавання, включаючи класифікацію, виявлення об’єктів і семантичну сегментацію. Архітектура MobileNetV2 заснована на перевернутій залишковій структурі, де вхід і вихід залишкового блоку є тонкими вузькими шарами, протилежними традиційним моделям залишків, які використовують розширені представлення у вхідних даних, а MobileNetV2 використовує легкі глибокі згортки для фільтрації функцій у проміжному шарі розширення. Загалом архітектура MobileNetV2 містить початковий повністю згорнутий рівень із 32 фі-

льтрами, за якими слідують 19 залишкових рівнів вузьких місць [9]. У роботі далі використовуватиметься зсув на 2 (stride=2), адже його основна ідея використання — це зменшення розмірів зображення (зменшення ширини та висоти), що допомагає зменшити обчислювальні витрати.

Проте навіть оптимізовані моделі вимагають спеціалізованих фреймворків і ретельного налаштування для ефективної роботи на мікроконтролерах.

Саме тому ще один підхід, який активно використовується в поєднанні з оптимізованими моделями, — це фреймворки для обробки на мікроконтролерах. Сучасні фреймворки, такі як TensorFlow Lite for Microcontrollers, CMSIS-NN і Edge Impulse, забезпечують інструменти для створення, оптимізації та розгортання моделей машинного навчання (ML) на пристроях із обмеженими ресурсами. Їхня мета — забезпечити ефективне виконання моделей машинного навчання на пристроях із малою пам’яттю та низькою обчислювальною потужністю.

Edge Impulse — це платформа повного циклу розробки, яка робить процес створення, розгортання та масштабування вбудованих програм з підтримкою машинного навчання простішим і швидшим, надаючи можливість розробникам створювати та оптимізувати рішення з реальними даними [10].

У статті запропоновано реалізацію третього і четвертого підходів — використаня оптимізованих моделей глибокого навчання та фреймворку Edge Impulse. Такий вибір пояснюється низкою факторів. MobileNet забезпечує високу точність і компактність, що робить її придатною для мікроконтролерів із обмеженими ресурсами. Edge Impulse, у свою чергу, автоматизує складні етапи оптимізації та розгортання моделі, що значно прискорює процес розробки та мінімізує ризик помилок. Крім того, обробка даних на пристрої дозволяє забезпечити автономність, низькі затримки та підвищену безпеку, оскільки дані не потрібно передавати в мережу. Тож, запропоноване рішення поєднує в собі ефективність, простоту впровадження та відповідність вимогам до компактних пристроїв із низьким енергоспоживанням.

2. Розгортання моделі розпізнавання об’єктів на мікроконтролері ESP32

Для одного з можливих рішень проблеми розпізнавання об’єктів за мікроконтролер було обрано обчислювальний модуль ESP32-CAM на основі мікроконтролера ESP32.

ESP32-CAM — це доступний і компактний модуль із можливостями Wi-Fi та Bluetooth, який також має вбудовану камеру [2]. Однак його апаратні обмеження ставлять значні виклики для виконання нейронних мереж. Основні обмеження можна виділити такі: незначний обсяг оперативної пам’яті, малий обсяг flash-пам’яті та невеликі обчислювальні потужності.

ESP32-CAM оснащений 520 KB внутрішньої оперативної пам'яті та додатковими 8 MB зовнішньої PSRAM [2]. Хоча PSRAM може використовуватися для зберігання більших обсягів даних, вона повільніша за внутрішню RAM і не завжди підходить для обчислювально-інтенсивних завдань. Цей обсяг пам'яті є дуже обмеженим для завантаження та виконання більшості моделей нейронних мереж, які часто потребують десятків або сотень мегабайтів для обробки великих масивів даних.

За замовчуванням ESP32-CAM має 32 Мбіт (4 МБ) флеш-пам'яті, де зберігаються машинні коди програми [2]. Це створює обмеження для завантаження моделей нейронних мереж, особливо великих. Флеш-пам'ять не придатна для виконання динамічних обчислень, тому обмеження її обсягу є критичним, якщо потрібно зберігати модель нейронної мережі на самому пристрої.

ESP32-CAM використовує процесор із двома ARM-ядрами і тактовою частотою до 240 МГц, який не призначений для інтенсивних обчислень, таких як виконання великих нейронних мереж [2].

Відсутність спеціалізованих обчислювальних блоків, таких як GPU чи TPU, обмежує можливість виконання обчислень із плаваючою точкою, які часто використовуються в нейронних мережах.

Тим не менше, такий модуль дуже зручний для простого розпізнавання об’єктів. Він має вбудовану камеру OV2640 та підтримує стиснений формат вихідного зображення JPEG [2].

На платформі розробки Edge Impule можна одразу обрати під яку апаратну платформу буде виконуватися розробка. Серед запропонованих варіантів є і популярний ESP32-CAM. Далі було обрано версію мережі MobileNet, а саме MobileNetV2 0.35 на базі алгоритму зі зниженим обчислювальним навантаженням FOMO. Edge Impulse FOMO (швидші об’єкти, більше об’єктів) — це новий алгоритм машинного навчання, який забезпечує виявлення об’єктів на пристроях із жорсткими обмеженнями, що дозволяє підраховувати кілька об’єктів і знаходити їхнє розташування на зображенні в режимі реального часу, використовуючи до 30 разів менше процесорної потужності та пам’яті, ніж MobileNet SSD або YOLOv5 [11].

Для прикладу було виконано розпізнавання ягоди полуниці (рис. 2) та її класифікація на стиглу та недозрілу, що може бути використано для створення недорогих агророботів для збору полуниць. Для створення навчального набору даних було використано існуючі зображення ягоди полуниці з відкритих джерел. Загалом для створення наборів було використано 120 зображень, які були поділені на 90 і 30 (75%/25%) на тренувальні та тестові набори відповідно. Всі вони були вручну проіндексовані та промарковані відповідно до стиглості ягід на зображеннях.

Рис. 2. Робота алгоритму класифікації полуниць

У процесі обробки зображень, вони були приведені до розміру 96 x 96. На основі отриманих даних було здійснено їх перетворення у набір спільних атрибутів. Після маркування наступним кроком стало створення та навчання моделі. На цьому етапі визначається тип даних для обробки, задаються параметри розміру зображень та метод їх масштабування. Водночас зберігається або змінюється співвідношення сторін залежно від потреб проєкту. У Edge Impulse передбачено три способи досягнення необхідних розмірів: Squash, Fit shortest axis і Fit longest axis. Для розробки цієї моделі було обрано останній метод Fit longest axis, який додає чорні смуги (так званий "letterbox") вздовж коротшої осі зображення для досягнення потрібного співвідношення сторін. Потім зображення масштабується до визначеного розміру шляхом інтерполяції, забезпечуючи збереження пропорцій. На основі цих налаштувань модель виконує розпізнавання об'єктів і класифікує їх у дві категорії: «стиглі» та «нестиглі» [12].

Результати створення характеристик наведені на рис. 3.

Як можна побачити з рис. 3, немає двох чітко виокремлених класів із тренувальних зображень. Причиною цього є наявність одночасно і стиглих, і не стиглих полуниць на одному зображенні.

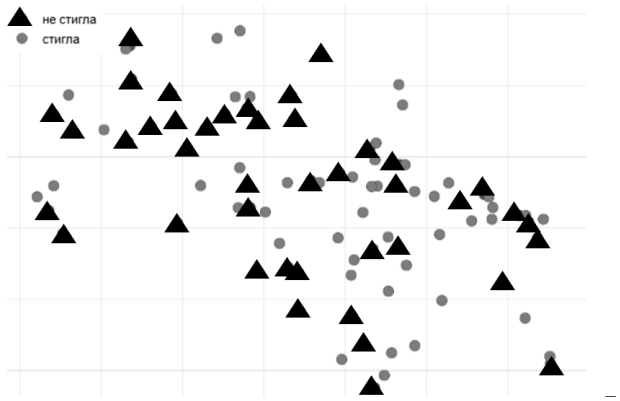

Рис. 3. Результати класифікації

Після налаштувань перед навчанням необхідно ще задати налаштування обробки нейронної мережі. Тут вказується кількість тренувальних циклів, чи використовувати навчений оптимізатор, навчальний крок (Learning rate), вид процесора для навчання, чи виконувати випадкове перетворення даних та інші додаткові умови. Для цієї системи були обрані такі параметри:

– number of training cycles: 70;
– use learned optimizer: ні (використовується стандартний оптимізатор);
– learning rate: 0.001;
– training processor: CPU (процес навчання буде здійснюватися на центральному процесорі);
– data augmentation: так (під час навчання будуть застосовуватися техніки аугментації даних, такі як обертання, зміна яскравості, масштабування тощо. Це допомагає покращити узагальнення моделі, зменшуючи перенавчання на тренувальному наборі даних) [12].

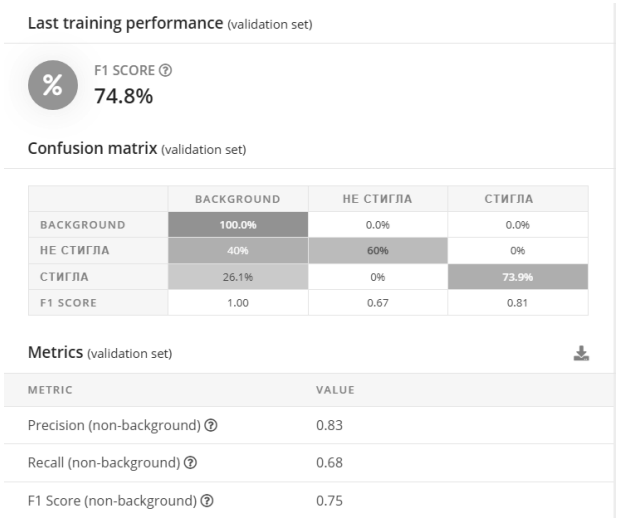

Рис. 4. Результати навчання моделі

3. Результати роботи

На рис. 4 наявна інформація про загальну продуктивність моделі на валідаційній вибірці, зокрема, F1 Score дорівнює 74.8%. F1 Score — це середньогармонічне значення точності (precision) та повноти (recall), яке показує баланс між цими двома метриками. Значення 74.8% вказує на хорошу продуктивність моделі [13]. Далі зображена матриця плутанини (confusion matrix) на валідаційній вибірці. Вона показує, як добре модель класифікує зразки в кожну з категорій. Рядки представляють фактичні класи, а стовпці — передбачувані класи [12].

На ній видно 100% зразків, які фактично є «Фоном» і були правильно класифіковані як «Фон».

Для рядка «Нестигла» 40% зразків, які фактично є «Нестигла», були неправильно класифіковані як «Фон». Однак ці неправильні результати є некритичними для таких систем як, наприклад, робот для збору полуниць, адже знайшовши нестиглу полуницю, система не має нічого робити. 60% зразків, які фактично є «Нестигла», були правильно класифіковані як «Нестигла». І найголовніше для даної системи — 0% зразків, які фактично є «Нестигла», були неправильно класифіковані як «Стигла».

Відповідні позитивні результати видно і в рядку «Стигла» у матриці плутанини: 26.1% зразків, які фактично є «Стигла», були неправильно класифіковані як «Фон»; 0% зразків, які фактично є «Стигла», були неправильно класифіковані як «Нестигла»; 73.9% зразків, які фактично є «Стигла», були правильно класифіковані як «Стигла». У цій же матриці вказані F1 Score для кожного класу: 1, 0.67, 0.81 до «Фон», «Стигла» та «Нестигла» відповідно.

Також результати оцінки містять такі метрики: Precision (non-background) або точність — відсоток правильних позитивних передбачень серед усіх позитивних передбачень [13]; Recall (non-background) або повнота — це відсоток правильних позитивних передбачень серед усіх фактичних позитивних зразків [13]; F1 Score (non-background) — це середньогармонічне значення точності та повноти для зразків, які не є фоном.

На рис. 5 наведені результати тестування моделі виявлення об'єктів. Там вказана загальна точність тестування 80%, точність Precision 94%, повнота Recall 87%, F1 Score 90%. Модель демонструє високу точність та повноту, що вказує на її здатність коректно виявляти більшість об'єктів і рідко видавати хибнопозитивні результати. У цілому, F1 оцінка 0.9 показує, що модель добре збалансована між точністю та повнотою. Проте загальна точність 80% вказує, що все ще є простір для поліпшення. Також зображена візуалізація ознак, де на графіку показано правильні і неправильні

передбачення моделі [12]. Приклад передбачення моделі наведено на рис. 6.

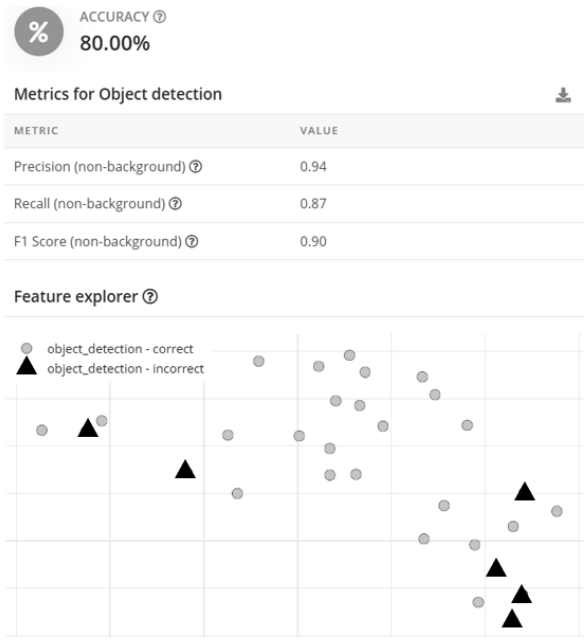

Рис. 5. Результати тестування моделі

Створена модель може розпізнавати полуницю в двох станах: стигла та нестигла. Наступним етапом став експорт моделі та конвертація її у вид, який можна завантажити на мікроконтролер. Для реалізації цього було конвертовано модель у бібліотеку та завантажено. Таким чином підключивши її, можна дописати потрібний код для подальшого керування периферійними пристроями та за потреби легко замінити модель на іншу [12].

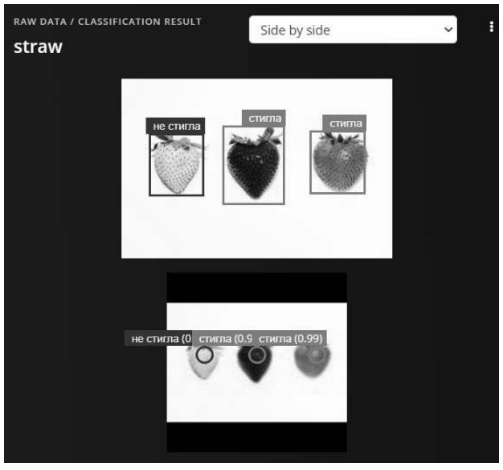

Рис. 6. Наочна демонстрація роботи моделі

Після імпортування моделі в середовище програмування плат Arduino IDE [14], було розроблено програму, що виводить поточну інформацію про розпізнавання об'єктів та їх класів у консоль послідовного монітору (рис. 7).

Рис. 7. Вигляд роботи системи

Висновки

У статті розглянуто ключові виклики, що виникають під час реалізації розпізнавання об'єктів на мікроконтролерах з обмеженими ресурсами, а також запропоновано ефективне рішення, яке поєднує використання оптимізованої моделі MobileNet на основі алгоритму зі зниженим обчислювальним навантаженням і платформи Edge Impulse. Це дослідження продемонструвало, що навіть у середовищах із низькою обчислювальною потужністю можна створювати ефективні ML-рішення, які відповідають вимогам щодо точності, продуктивності та автономності.

Використання MobileNet на основі алгоритму зі зниженим обчислювальним навантаженням дозволило подолати проблему обмежених ресурсів завдяки її компактній архітектурі, що зменшує кількість обчислень за рахунок глибинних розділених згорток. Такий підхід забезпечив можливість запускати модель навіть на пристроях із мінімальними апаратними характеристиками, водночас зберігаючи прийнятний рівень точності. До того ж інтеграція з Edge Impulse значно спростила процес розробки. Ця платформа забезпечила інструменти для автоматизації таких складних етапів, як оптимізація моделі, її квантоване навчання та інтеграція з мікроконтролером.

Завдяки цьому вдалося суттєво скоротити час розробки та уникнути багатьох потенційних помилок.

Ще однією важливою перевагою стала можливість локальної обробки даних безпосередньо на пристрої. Це рішення дозволило мінімізувати затримки, знизити залежність від зовнішніх мережевих ресурсів і забезпечити додаткову безпеку, оскільки дані не потребували передачі на сервери. Такий підхід особливо важливий для завдань реального часу, де затримка може суттєво вплинути на ефективність роботи системи.

Розроблена модель успішно виконала поставлену задачу — класифікацію об'єктів безпосередньо на мікроконтролері у дві категорії: «стиглі» та «нестиглі» з точністю 80%. Це підтвердило ефективність запропонованого підходу, особливо для прикладних застосувань.

Дане дослідження відкриває нові перспективи для створення доступних і енергоефективних IoT-рішень, що здатні працювати автономно. Запропонований підхід може знайти широке застосування в таких галузях, як розумне сільське господарство, промислова автоматизація, системи моніторингу та переносні пристрої. Загалом робота доводить, що сучасні технології машинного навчання стають доступними навіть для найменш потужних апаратних платформ, розширюючи їхні можливості та сфери використання.

Подальші дослідження потрібні для визначення зв’язку між точністю розпізнавання, потрібним обсягом пам’яті та тактовою частотою, необхідною для забезпечення достатньої швидкості обробки зображень. Також потрібно дослідити способи реалізації запропонованого підходу на інших мікроконтролерах, зокрема, STM32.

Література

1. Ashiq, F.; Asif, M.; Bin Ahmad, M.; Zafar, S.; Masood, K.; Mahmood, T.; Mahmood, M.T.; Lee, I.H. *CNN-Based Object Recognition and Tracking System to Assist Visually Impaired People*: IEEE Access, 2022. 14819-14821 с.
2. ESP32-CAM: Technical Specification. URL: https://docs.sunfounder.com/projects/galaxy-rvr/en/latest/hardware/cpn_esp_32_cam.html (Дата звернення: 17.11.2024)
3. What are the pros and cons of AI (compared to traditional computing). URL: https://www.ibm.com/think/insights/artificial-intelligence-advantages-disadvantages (Дата звернення 17.11.2024)
4. Wei Fang, Lin Wang, Peiming Ren. *Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments*: IEEE Access, 2019. 1935-1938 с.
5. Yahui Nan, Jianguo Ju, Qingyi Hua, Haoming Zhang, Bo Wang: *A-MobileNet: An approach of facial expression recognition*: Alexandria Engineering Journal, 2021. 4435-4437 с.
6. SqueezeNet: *The Key to Unlocking the Potential of Edge Computing*. URL: https://medium.com/sfu-cspmp/squeezenet-the-key-to-unlocking-the-potential-of-edge-computing-c8b224d839ba (Дата звернення 18.11.2024)
7. Howard A. G., Zhu M., Chen B. et al. *Mobilenets: efficient convolutional neural networks for mobile vision application*: Cornell University, 2017. 1-5с.
8. Wei Wang, Yutao Li, Ting Zou, Xin Wang, Jieyu You, Yanhong Luo: *A Novel Image Classification Approach via Dense-MobileNet Models*: Hindawi, 2020 3с.
9. MobileNetV2. URL: https://paperswithcode.com/method/mobilenetv2 (Дата звернення 18.11.2024)
10. EDGE IMPULSE. URL: https://edgeimpulse.com/about (Дата звернення 18.11.2024)
11. FOMO: Object detection for constrained devices. URL: https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks/object-detection/fomo-object-detection-for-constrained-devices (Дата звернення 18.11.2024)
12. Скотаренко А.О. Система автоматичного збору полуниць: бак.р. …бакалавра: Київ, 2024. 40-52 с.
13. f1_score. URL: https://scikit-learn.org/1.5/modules/generated/sklearn.metrics.f1_score.html (Дата звернення 19.11.2024)
14. Arduino IDE. URL: https://www.arduino.cc/en/software (Дата звернення 18.11.2024)

Одержано: 25.11.2024
Внутрішня рецензія отримана: 30.11.2024
Зовнішня рецензія отримана: 30.11.2024

Про авторів:

Бердник Юрій Михайлович,
аспірант, асистент кафедри
інформаційних систем та технологій
https://orcid.org/0000-0002-0008-4748

Скотаренко Анастасія Олександрівна,
студентка магістр
https://orcid.org/0009-0006-1429-5729

Місце роботи авторів:

НТУ України «Київський політехнічний
інститут імені Ігоря Сікорського»
Тел.: +38-044-204-86-10.
E-mail: y.berdnyk@kpi.ua,
skotarenko.anastasiia@lll.kpi.ua