

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційних радіотехнологій та технічного захисту інформації
(повна назва)

Кафедра мікропроцесорних технологій і систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Розробка та дослідження системи безпеки з функцією автоматизованого
розпізнавання рухомих об'єктів
(тема)

Виконав:

здобувач 2 року навчання,
групи ІМСм-23-1

Томас МАЧОНІС

(власне ім'я, прізвище)

Спеціальність 171 Електроніка
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Інженерія
мікропроцесорних систем
(повна назва освітньої програми)

Керівник проф. Олег ЗУБКОВ
(посада, власне ім'я, прізвище)

Допускається до захисту

В.о. завідувача кафедри МТС



(підпис)

Олег ЗУБКОВ

(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет інформаційних радіотехнологій та технічного захисту
інформації

Кафедра мікропроцесорних технологій і систем

Рівень вищої освіти другий (магістерський)

Спеціальність 171 Електроніка

(код і повна назва)

Тип програми освітньо-наукова

(освітньо-професійна або освітньо-наукова)

Освітня програма Інженерія мікропроцесорних систем

(повна назва)

ЗАТВЕРДЖУЮ:

В.о. зав. кафедри МТС



Олег ЗУБКОВ

(підпис)

«23» 06 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Мачонісу Томасу Сігітасовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та дослідження системи безпеки з функцією
автоматизованого розпізнавання рухомих об'єктів

затверджена наказом університету від 17 04 2025 р. № 292Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 23 06 2025 р.

3. Вихідні дані до роботи Розпізнавання людини на фоні руху інших рухомих об'єктів:
автомобілів, дерев, домашніх тварин

4. Перелік питань, що потрібно опрацювати в роботі

- 1) Провести огляд та аналіз методів та систем розпізнавання зображень різних класів об'єктів
- 2) Провести вибір архітектури нейронної мережі для розпізнавання зображень людини та розробку дата сету для навчання нейронної мережі
- 3) Виконати навчання нейронної мережі та провести тестування ефективності її роботи
- 4) Розробити схему електричну структурну пристроя розпізнавання людей системи безпеки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін виконання етапів роботи	Примітка
1	Огляд та аналіз методів та систем розпізнавання зображень різних класів об'єктів	21.04.2025-27.04.2025	виконано
2	Провести вибір архітектури нейронної мережі для розпізнавання зображень людини та розробку дата сету для навчання нейронної мережі	28.04.2025-7.05.2025	виконано
3	Навчання нейронної мережі	8.05.2025-16.05.2025	виконано
4	Тестування ефективності роботи нейронної мережі	17.05.2025-24.05.2025	виконано
5	Розробити схему електричну структурну пристроя розпізнавання людей системи безпеки	25.05.2025-7.06.2025	виконано
6	Написання пояснювальної записки	8.06.2025-15.06.2025	виконано

Дата видачі завдання 21 _____ 04 _____ 2025р.

Здобувач _____
(підпис)

Керівник роботи _____
(підпис)

_____ проф. Олег ЗУБКОВ
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Кваліфікаційна робота: 91 с., 2 таб., 23 рис., 33 джерел, 4 додаток.

РОЗПІЗНАВАННЯ, ЗОБРАЖЕННЯ, НЕЙРОННА МЕРЕЖА, ЗГОРТКА, АЛГОРИТМ, ДАТАСЕТ, ВБУДУВАННЯ, СИСТЕМА БЕЗПЕКИ.

Об'єкт дослідження – система безпеки з функцією розпізнавання рухомих об'єктів.

Мета дослідження – розробка вбудованої нейронної мережі розпізнавання рухомих об'єктів та дослідження впливу інтенсивності освітлення і дальності до об'єкту на ефективність розпізнавання.

Методи дослідження – аналітичний, статистичний, оптимізаційні методи.

Проаналізовано сучасні алгоритми та структури нейронних мереж для розпізнавання різних класів об'єктів на зображенні, а також можливість використання цих алгоритмів та мереж на сучасних апаратних платформах. Обґрунтовано вибір структур нейронних мереж для практичної реалізації та вбудування у апаратну платформу ESP32-S3-CAM. Розроблено датасет для навчання нейронної мережі, виконано навчання нейронних мереж. Здійснено вбудування навчених нейронних мереж у апаратну платформу. Проведені дослідження впливу інтенсивності освітлення та дальності до об'єкту розпізнавання на імовірності вірного розпізнавання та помилкового розпізнавання. Виконано оптимізацію нейронної мережі. Результатом роботи є створена нейронна мережа та пропозиції до практичного використання мережі у різних умовах експлуатації та встановлення.

Отримані результати можуть бути впроваджені у практичну діяльність фірм та підприємств, що пропонують клієнтам функції безпеки із інтелектуальними функціями.

ABSTRACT

Qualification thesis: 91 pages, 2 tables, 23 figures, 33 sources, 4 appendices.

RECOGNITION, IMAGE, NEURAL NETWORK, CONVOLUTION, ALGORITHM, DATASET, EMBEDDING, SECURITY SYSTEM.

Object of the research – a security system with the function of recognizing moving objects.

Purpose of the research – development of an embedded neural network for recognizing moving objects and studying the influence of lighting intensity and distance to the object on recognition efficiency.

Research methods – analytical, statistical, and optimization methods.

The thesis analyzes modern algorithms and neural network architectures for recognizing different classes of objects in images, as well as the possibility of implementing these algorithms and networks on modern hardware platforms. The choice of neural network structures for practical implementation and embedding into the ESP32-S3-CAM hardware platform is substantiated. A dataset for training the neural network was developed, and the neural networks were trained. The trained neural networks were embedded into the hardware platform. Research was conducted on the influence of lighting intensity and object distance on the probabilities of correct and incorrect recognition. The neural network was optimized.

The result of the work is a developed neural network and recommendations for its practical use under various operating and installation conditions. The obtained results can be implemented in the practical activities of companies and enterprises that offer their clients security solutions with intelligent features.

ЗМІСТ

ВСТУП	7
1 ОГЛЯД І АНАЛІЗ СУЧАСНИХ МЕТОДІВ І ПЛАТФОРМ ДЛЯ РОЗПІЗНАВАННЯ РУХОМИХ ОБ’ЄКТІВ	
1.1 Огляд існуючих відеосистем розпізнавання рухомих об’єктів	10
1.2 Комп’ютерний зір, алгоритми розпізнавання рухомих об’єктів	13
1.3 Огляд алгоритмів та структур нейронних мереж	22
1.4 Аналіз апаратних платформ для реалізації нейронних мереж	30
1.5 Висновки до першого розділу	34
2 ПРОЕКТУВАННЯ СИСТЕМИ БЕЗПЕКИ З АВТОМАТИЧНИМ РОЗПІЗНАВАННЯ РУХОМИХ ОБ’ЄКТІВ	
2.1 Вибір апаратної платформи та її особливості ESP32-S3-CAM	36
2.2 Опис архітектурної та структурної схеми системи безпеки	39
2.3 Вибір алгоритму розпізнавання рухомих об’єктів	45
2.4 Вибір нейронної мережі	47
2.5 Висновки до другого розділу	49
3 РОЗРОБКА ТА НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ	
3.1 Створення та підготовка датасету	51
3.2 Навчання нейронної мережі	54
3.3 Аналіз ефективності навчання нейронної мережі	57
3.4 Оптимізація та впровадження нейронної мережі на ESP32-S3-CAM	59
3.5 Висновки до третього розділу	60
4 ОЦІНКА ЕФЕКТИВНОСТІ РОБОТИ СИСТЕМИ БЕЗПЕКИ	
4.1 Експериментальні результати	62
4.2 Аналіз точності та швидкості розпізнавання	63
4.3 Дослідження впливу зовнішніх факторів на ефективність розпізнавання	65
4.4 Висновки до четвертого розділу	67

ВИСНОВКИ	69
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	71
ДОДАТОК А	75
ДОДАТОК Б	76
ДОДАТОК В	80
ДОДАТОК Г	83

ВСТУП

У сучасному світі системи безпеки відіграють ключову роль у захисті особистого майна, приватних осель, підприємств, заводів, банків та інших. Цифрові технології та штучний інтелект в системах захисту дозволяють замінити велику кількість людей необхідних за наглядом та охороною територій підприємств, заводів або навіть просто осель. Розвиток цифрових технологій та штучного інтелекту дозволяє за необхідністю створювати автономні та ефективні рішення для автоматичного розпізнавання загроз та виявлення несанкціонованого проникнення у реальному часі. Однією з найбільш перспективних технологій у цій сфері є використання нейронних мереж, які дозволяють здійснювати аналіз відеопотоку, ідентифікувати об'єкти та приймати відповідні рішення без участі людини.

Актуальність теми. У сучасному світі постійно зростає попит на автоматизовані системи безпеки, здатні оперативно розпізнавати підозрілі об'єкти. Методи які зараз існують, такі як датчики руху, часто дають помилкові спрацювання через тварин, зміну освітлення або погодні умови. Впровадження нейронної мережі до комп'ютерного зору надасть можливість розпізнавати об'єкти руху дозволяючи усунути раніше перелічені недоліки, тим самим роблячи систему безпеки більш точною, адаптивною та доступною.

Впровадження технологій з використанням таких модулів як Arducam ESP32 UNO PSRAM, ESP32-CAM або ESP32-S3-CAM, відкривають нові можливості для створення бюджетних, компактних та автономних систем безпеки. Завдяки глибокому навчанню нейронних мереж, навіть малопотужні пристрої можуть ефективно розпізнавати рухомі об'єкти та класифікувати їх (людина, тварина, сторонні предмети тощо).

Об'єктом дослідження є система безпеки, що використовує нейронну мережу для розпізнавання рухомих об'єктів. Комп'ютерний зір та алгоритми

нейронних мереж, вбудовуються в мікроконтролер для автоматичної класифікації рухомих об'єктів.

Метою дослідження є розробка вбудованої нейронної мережі в модуль для автоматичного розпізнавання рухомих об'єктів та дослідження впливу інтенсивності освітлення і дальності до об'єкту на ефективність розпізнавання.

Для досягнення мети поставлено такі завдання:

- провести огляд та аналіз сучасних методів розпізнавання рухомих об'єктів;
- визначити оптимальну нейронну мережу для реалізації на платформі ESP32-S3-CAM;
- створити та підготувати датасет для навчання нейронної мережі;
- навчити нейронну мережу та оптимізувати її для роботи на мікроконтролері;
- вбудувати нейронну мережу в ESP32-S3-CAM і протестувати її продуктивність;
- дослідити вплив зовнішніх факторів (освітлення, відстань, типи об'єктів) на точність розпізнавання.

Методи дослідження, що використовуються в роботі, включають аналітичний, статистичний та оптимізаційний метод. Використовуються математичні моделі машинного навчання, експериментальні дослідження продуктивності модулю ESP32-CAM, а також методи проведення оцінки точності та швидкості розпізнавання руху в різних умовах.

Результати дослідження можуть бути впроваджені у практичну діяльність компаній, що займаються розробкою розумних систем безпеки. Автономна система розпізнавання, побудована на модулі ESP32-S3-CAM, може використовуватися для:

- відеоспостереження у приватних будинках;
- розпізнавання руху в розумних будинках;

- контролю безпеки на підприємствах;
- моніторингу вхідних зон у бізнес-центрах;
- моніторингу зон з обмеженим доступом.

Таким чином, така система безпеки з доступною системою розпізнавання рухомих об'єктів, стає менш фінансово затратною та більш енергоефективною без втрат рівня захисту.

1 ОГЛЯД І АНАЛІЗ СУЧАСНИХ МЕТОДІВ І ПЛАТФОРМ ДЛЯ РОЗПІЗНАВАННЯ РУХОМИХ ОБ'ЄКТІВ

1.1 Огляд існуючих відеосистем розпізнавання рухомих об'єктів

Сучасні системи безпеки відіграють ключову роль у захисті житла, підприємств, транспорту та інформаційних ресурсів. Одним із найперспективніших напрямів розвитку систем безпеки є використання автоматизованого відеоспостереження з функцією розпізнавання об'єктів. Зокрема, у роботах [1, 2, 3] аналізуються типові рішення та технології відеоаналітики в безпекових системах.

Нижче наведено основні типи відеосистем з функцією розпізнавання рухомих об'єктів:

- системи на основі детекції руху;
- системи з використанням методів машинного навчання;
- інтелектуальні системи відеоспостереження з використанням комп'ютерного зору;
- системи з трекінгом об'єктів;
- відеосистеми з багатоканальним аналізом.

Системи на основі детекції руху. Ці системи базуються на простих алгоритмах, які відслідковують зміни між кадрами відео та визначають рух об'єктів. Вони часто використовуються в базових системах відеоспостереження для виявлення руху в межах певної області. Однак ці системи можуть бути обмеженими, оскільки не завжди здатні точно розпізнати об'єкти в умовах змінного освітлення або складного фону.

Системи з використанням методів машинного навчання. У таких відеосистемах застосовуються алгоритми машинного навчання, зокрема глибокі нейронні мережі, для розпізнавання, класифікації та відстеження об'єктів. Вони здатні не лише виявляти рух, але й визначати тип

об'єкта (наприклад, людина, транспортний засіб, тварина) та слідкувати за його переміщенням протягом відеопотоку. Для навчання таких моделей використовують великі об'єми даних, що дозволяє підвищити їхню точність у різноманітних умовах.

Інтелектуальні системи відеоспостереження з використанням комп'ютерного зору. Ці системи поєднують алгоритми детекції руху з передовими методами комп'ютерного зору для покращення точності розпізнавання та класифікації об'єктів. Вони здатні працювати в реальному часі, аналізуючи відео з камер спостереження та приймаючи рішення на основі встановлених алгоритмів (наприклад, сповіщення про несанкціоноване вторгнення або активність в обраних зонах). Такі системи можуть адаптуватися до умов навколишнього середовища (освітлення, погода) завдяки використанню гнучких алгоритмів.

Системи з трекінгом об'єктів. Окрім розпізнавання рухомих об'єктів, ці системи забезпечують точне відстеження їх руху протягом тривалого часу. Застосовуються різні методи трекінгу, зокрема на основі кореляції, оптичного потоку та фільтрів Калмана. Ці системи здатні не лише виявити рух, а й відслідковувати траєкторії об'єктів, що має важливе значення в системах безпеки, де необхідно фіксувати поведінку конкретних об'єктів у просторі.

Відеосистеми з багатоканальним аналізом. Використовуються для покращення ефективності розпізнавання в складних умовах, наприклад, в умовах великої кількості об'єктів або шуму в середовищі. Вони обробляють зображення з кількох камер, об'єднуючи інформацію для досягнення більш точного результату. Такі системи можуть використовувати алгоритми для синхронізації даних та комбінування зображень з різних кутів огляду.

Розглянемо кілька прикладів існуючих відеосистем розпізнавання рухомих об'єктів, що активно використовуються у різних сферах:

1. Система відеоспостереження з інтелектуальним аналізом відео. Hikvision [1] є одним з лідерів у виробництві систем відеоспостереження, що використовують технології розпізнавання руху та об'єктів. Вони застосовують алгоритми комп'ютерного зору та штучного інтелекту для виявлення руху, відстеження об'єктів, розпізнавання облич та навіть виявлення небезпечних ситуацій. Такі системи використовуються на підприємствах, вулицях, паркуваннях та в інших місцях для забезпечення безпеки.

2. Система Tesla Autopilot [3] використовує відеокамери та сенсори для виявлення та відстеження рухомих об'єктів, таких як інші автомобілі, пішоходи, дорожні знаки та інші перешкоди. Це дозволяє автомобілю адаптуватися до умов на дорозі та забезпечує високий рівень автономності.

3. Google DeepMind [4] застосовує передові нейронні мережі для аналізу відео та зображень, виявляючи рух та ідентифікуючи об'єкти. Це використовується не тільки для покращення роботи відеоспостереження, а й для розвитку медицини, робототехніки та інших індустрій.

4. Amazon Rekognition [5] дозволяє автоматично розпізнавати об'єкти та рух у відео, що застосовується для моніторингу безпеки, розпізнавання осіб та аналізу відео для маркетингових та бізнесових цілей.

5. Uptake [6] використовує відеосистеми з алгоритмами машинного навчання для моніторингу промислових об'єктів і оцінки їх стану. Вони допомагають виявляти несправності в обладнанні та спостерігати за рухом персоналу на виробничих лініях.

6. Система Ring [7] є популярним відеодомофоном, який використовує технології розпізнавання руху та об'єктів для забезпечення безпеки вдома. Камери автоматично виявляють рух і сповіщають власників через мобільний додаток.

7. Xiaomi пропонує системи відеоспостереження для домашнього використання з підтримкою розпізнавання руху та людей Xiaomi Mi Home Security Camera [8]. Це дозволяє забезпечити безпеку за доступною ціною.

8. Bosch VMS використовує інтелектуальну відеоаналітику для відстеження руху, розпізнавання об'єктів та моніторингу безпеки на великих об'єктах, таких як заводи, торгові центри чи міські площі Bosch Video Management System (VMS) [9].

1.2 Комп'ютерний зір, алгоритми розпізнавання рухомих об'єктів

Сучасні системи безпеки відіграють ключову роль у захисті житла, підприємств, транспорту та інформаційних ресурсів. Одним із найперспективніших напрямів розвитку систем безпеки є використання автоматизованого відеоспостереження з функцією розпізнавання об'єктів [4]. У таких системах застосовується технологія комп'ютерного зору (computer vision), яка дозволяє аналізувати відеопотік та автоматично визначати, відстежувати й класифікувати об'єкти у реальному часі [10]. Це одна з ключових галузей штучного інтелекту, яка ґрунтується на методах розпізнавання образів та машинного навчання.

Комп'ютерний зір реалізується шляхом аналізу набору пікселів, що формують зображення. Він також передбачає автоматичне вилучення інформації із зображень, такої як 3D-моделі, розпізнавання об'єктів, пошук за змістом тощо. Завдяки цьому технологія широко використовується у відеоспостереженні, де алгоритми штучного інтелекту дозволяють автоматично виявляти рух, розпізнавати людей і оцінювати ситуацію у кадрі [3], [9].

Найбільш поширені завдання комп'ютерного зору:

- Розпізнавання – є одним із базових та першорядних завдань у обробці зображень. Воно допомагає класифікувати та ідентифікувати об'єкти, що характеризуються певним набором властивостей та ознак.

- Відновлення зображень – це видалення шуму з використанням різних методів, наприклад, розмиття за допомогою фільтрів на основі машинного навчання (шум датчика, розмитість рухомого об'єкта і т. д.).
- Аналіз руху – це використання комп'ютерного зору для оцінки швидкості руху об'єктів у відеопотоці.
- Відновлення чи реконструкція сцени – допомагає відтворити тривимірну модель зображення або сцени, що вводиться за допомогою зображень чи відео. Найчастіше моделлю служить набір точок тривимірного простору.
- Обробка та аналіз зображень – це процес трансформації одного 2D-зображення в інше, що здійснюється за допомогою піксельних операцій, зокрема зміни контрастності або обертання зображення.
- Високорівнева обробка – це аналіз невеликого обсягу даних із використанням методів вилучення інформації із сигналів. Вона дозволяє визначати ключові точки або області зображення, де ймовірно розташований об'єкт, що становить інтерес.

Сучасні відеосистеми розпізнавання рухомих об'єктів активно використовуються в різних сферах, від охоронних систем до транспорту та автоматизованих складських приміщень. Вони застосовуються для виявлення, відстеження та ідентифікації об'єктів у реальному часі, що дозволяє автоматизувати процеси спостереження та значно підвищити рівень безпеки [10].

1.2.1 Принцип роботи комп'ютерного зору.

Комп'ютери інтерпретують зображення як послідовність пікселів, кожен з яких має власний набір значень кольору. Пікселі є необробленими будівельними блоками зображення. Кожне зображення складається з набору пікселів, а піксель вважається «кольором» або «яскравістю» світла, що

з'являється на зображенні. Якщо ми розглядаємо зображення як сітку, то кожен квадрат містить один піксель [10], [11].



Рисунок 1.1 – Приклад зображення

Зображення на рисунку 1.1 має роздільну здатність 1280×720 пікселів, де 1280 – це ширина, а 720 – висота. Ми можемо уявити зображення у вигляді матриці. У цьому випадку наша матриця має 1280 стовпців (ширина) та 720 рядків (висота) та містить $1280 \times 720 = 921\,600$ пікселів, які представлені двома способами: а) відтінки сірого (один канал); б) колір.

У зображеннях із градаціями сірого кольору кожен піксель є скалярним значенням від 0 до 255, де нуль відповідає «чорному» кольору, а 255 відповідає «білому». Значення між 0 та 255 мають різні відтінки сірого, де значення ближче до 0 темніше, а значення ближче до 255 світліше. Градієнтне зображення у градаціях сірого (рисунок 1.2) демонструє більш темніші пікселі з лівого боку, а світліші пікселі з правого. На малюнку (рисунок 1.2) можна побачити те, як значення в градаціях сірого перетворюються на двовимірний масив цілих чисел.

Ряди чисел праворуч (крайній правий малюнок) – комп’ютерна вистава введеного зображення. У прикладі зображення має 12 стовпців та 16 рядків, що означає 192 вхідних значень цього зображення.

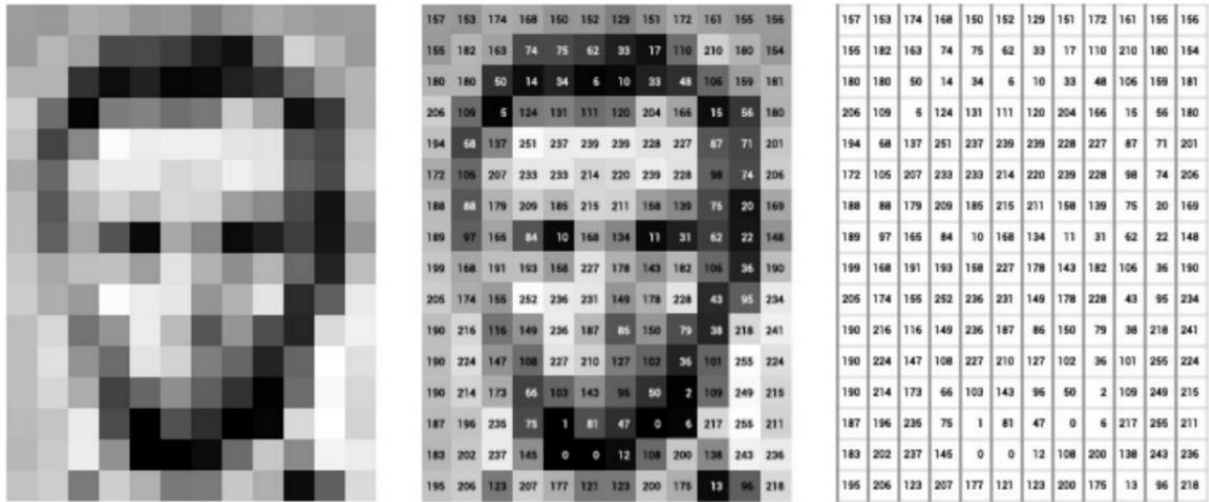


Рисунок 1.2 – Зображення та їх матричні представлення

Ще необхідно описати як комп’ютер перетворює для себе кольорові зображення у вигляді матриць. Пікселі у кольоровому просторі RGB більше не є скалярними значеннями, як було у зображеннях у градаціях сірого, на одному каналі. Натомість пікселі представлені списком з трьох значень: одне значення для компонента червоного (Red), друге для зеленого (Green) та третє для синього (Blue). Щоб визначити колір у кольоровій моделі RGB, потрібно підрахувати кількість червоного, зеленого та синього кольору, що містяться в одному пікселі. Кожен канал Red, Green та Blue може мати певні значення в діапазоні $[0, 255]$, всього 256 відтінків, де 0 означає відсутність уявлення, а 255 – це повне уявлення. Враховуючи що значення пікселя має бути лише в діапазоні $[0, 255]$, то використовують 8 бітові цілі числа без знака для представлення яскравості.

Розглянемо приклад створення різних кольорів зображення на рисунку 1.3.

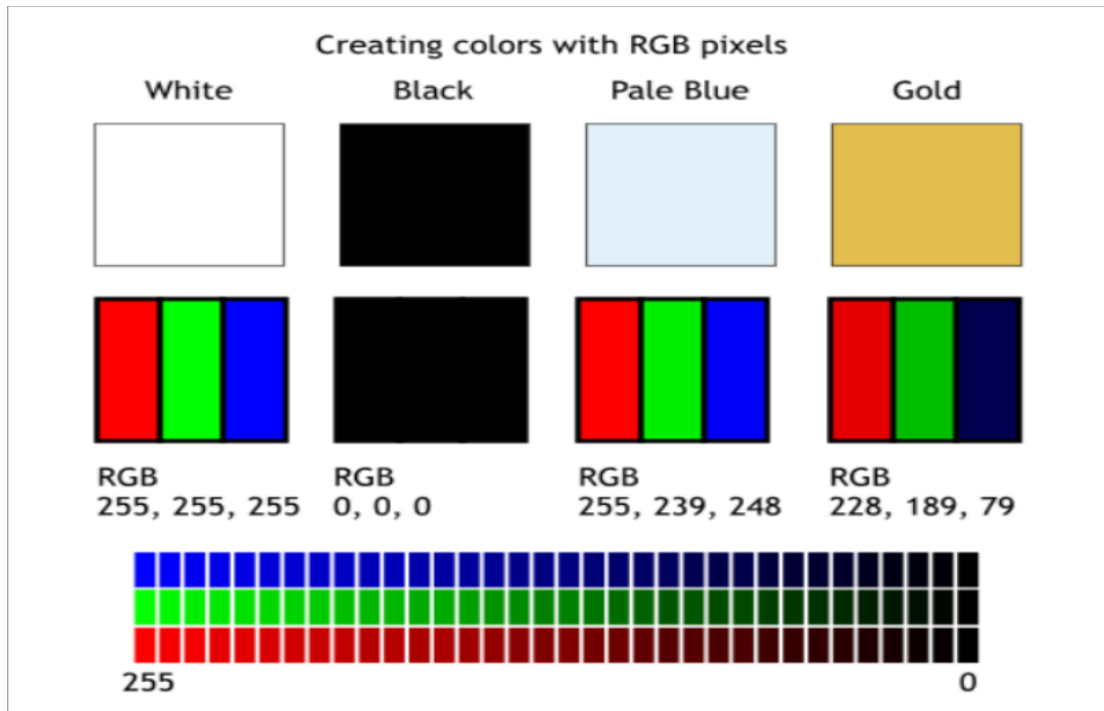


Рисунок 1.3 – Створення кольорів у RGB пікселі

З рисунка 1.3 можна зрозуміти, що кожен піксель складається з трьох різних кольорів, при зміні їх відтінків а також при їхньому комбінуванні можна отримати абсолютно різні кольори. Для вирішення завдань детектування, розпізнавання та класифікації об'єктів на графічному зображенні необхідно попередньо обробляти зображення (наприклад, виконати операцію середнє віднімання чи масштабування) [1], [3], [10]. Оскільки типи даних, які використовуються бібліотеками (наприклад OpenCV), що завантажують зображення з диска, то їх необхідно перетворювати, перш ніж безпосередньо застосовувати алгоритми навчання до зображень. Враховуючи наші три значення Red, Green та Blue, ми можемо поєднати їх у кортеж RGB (червоний, зелений, синій). Цей кортеж представляє цей колір у кольорному просторі RGB. Ми можемо перетворити зображення RGB як три незалежних матриць, шириною W і висотою H , по одній для кожного з компонентів RGB, як показано на рисунку 1.3. Ми можемо об'єднати ці три матриці для отримання багатовимірного масиву з розміром, що визначається формою 1.1:

$$W \times H \times D, \quad (1.1)$$

де W – ширина матриці, кількість стовбців,

H – висота матриці, кількість рядків,

D – глибина чи кількість каналів, для колірного простору RGB глибина $D = 3$.



Рисунок 1.4 – Вихідне зображення та його RGB канали

1.2.2 Перетворення рівня яскравості зображення.

Перетворення яскравості – це базова операція в обробці зображень, яка полягає у зміні значень пікселів для покращення візуальної якості або підготовки до подальшого аналізу. Зазвичай використовуються функції, що змінюють значення яскравості у межах допустимого діапазону (наприклад, 0–255 або 0–1), не виходячи за його межі.

Одним із найефективніших методів є вирівнювання гістограми, яке перерозподіляє яскравість пікселів так, щоб охопити весь діапазон яскравостей з рівномірною частотою. Це покращує контрастність

зображення, особливо в тих випадках, коли воно занадто темне або засвічене. Для цього використовується кумулятивна функція розподілу (cumulative distribution function, CDF), яка перетворює вихідні значення пікселів на нові, з урахуванням їх частоти в зображенні [11].

Алгоритми розпізнавання руху.

Виявлення руху – ключовий етап у відеоаналітиці. Воно передбачає виявлення змін у послідовності кадрів з метою визначення наявності об'єктів, що рухаються. Розглянемо кілька варіантів:

- *Детектування за різницею кадрів* (Frame Differencing). Найпростіший метод, що полягає у порівнянні двох послідовних кадрів. Якщо різниця між ними перевищує поріг – фіксується рух. Метод швидкий, проте має обмежену точність і чутливий до шуму [10].

- *Віднімання фону* (Background Subtraction). Цей підхід дозволяє ізолювати рухомі об'єкти, віднімаючи статичне зображення фону від поточного кадру [11].

- *Оптичний потік* (Optical Flow). Оптичний потік визначає рух пікселів між кадрами, це дозволяє оцінити швидкість і напрямок руху. Цей метод особливо корисний для детального аналізу сцени, але є ресурсномістким [10], [11].

- *Використання ключових точок* (Feature-based). Методи на основі ключових точок, такі як ORB, FAST та BRIEF, дозволяють ефективно виявляти та відстежувати об'єкти навіть за змін умов освітлення чи часткових обертань. Вони працюють із дескрипторами – наборами ознак, що описують певну ділянку зображення [11].

- *Методи глибокого навчання* (наприклад, YOLO, SSD, MobileNet) – сучасні нейронні мережі, що дозволяють точно і швидко детектувати рухомі об'єкти у реальному часі. Але для їхньої роботи потрібні потужні обчислювальні ресурси [11], [12], [13].

1.2.3 Метод віднімання фону для виявлення руху.

Віднімання фону є ключовим методом для виявлення об'єктів на відео, зокрема для виявлення руху людей або транспортних засобів [10], [11]. Головною метою цього методу є виявлення всіх об'єктів на передньому плані як різницю між поточним кадром та статичним зображенням фону.

Існує багато методів віднімання фону, які мають різні сильні та слабкі сторони залежно від вимог до продуктивності та обчислювальних ресурсів. Надійний алгоритм віднімання фону повинен бути стійким до змін освітлення, повторюваних рухів і погодних умов.

Розглянемо які методи існують, у чому їхні переваги та недоліки:

- *Медіанна фільтрація.* Цей метод оцінює модель фону за допомогою обчислення медіанного значення кожного пікселя. Виявлення об'єкта відбувається тільки після того, як піксель пройшов через більшу частину кадру. Переваги: простота конструкції та швидкість. Недоліки: проблеми при відстеженні цілей на анімованих фонах.

- *Різниця кадрів.* Один із найпростіших методів. Попередній кадр використовується як фон, а зміни в пікселях між кадрами визначають рухомі об'єкти. Цей метод ефективний, коли всі пікселі переднього плану рухаються, а фон статичний.

- *Фільтрація MIN-MAX.* Для кожного пікселя використовуються три значення для визначення, чи є піксель частиною фону. Це дозволяє враховувати різні інтенсивності та зміни на фоні.

- *Одинарна гауссова фільтрація.* Визначення фону за допомогою нормального розподілу для кожного фоновго пікселя з використанням стандартного відхилення та середнього значення.

- *Фільтрація Калмана.* Рекурсивний метод, який використовує статистичне оновлення для компенсації змін освітлення і вбудовування об'єктів з тривалими зупинками в модель фону.

- *Моделі Маркова.* Моделі, які можуть ефективно моделювати варіації інтенсивності пікселів у змінних умовах, таких як зміна погоди чи освітлення.

Моделювання фону за допомогою одного зображення вимагає наявності статичного фону без шуму, що складно забезпечити в реальних умовах. Для цього використовуються статистичні методи, наприклад модель змішання Гаусса (GMM), яка здатна ефективно працювати з мультимодальними фонами. Цей підхід є потужним у випадках, коли фон містить анімовані текстури, такі як дерева, що колишуться від вітру (рисунок 1.5).



Рисунок 1.5 – а) Відеозображення без фільтру датчика руху. Б) Відеозображення фільтру датчика руху.

Методи віднімання фону мають різні переваги та недоліки. Методи, що працюють з відео в градаціях сірого, зазвичай дають менш точні результати порівняно з кольоровими відео. Методи, такі як KDE та GMM, демонструють кращі результати в умовах шуму, в той час як метод Eigen має високе споживання пам'яті і не підходить для реального використання через велику потребу в ресурсах.

Методи віднімання фону мають різні рівні точності і складності. Вибір методу залежить від специфічних вимог до системи, таких як швидкість, здатність працювати в умовах зміни освітлення, а також обробка великих обсягів даних.

1.3 Огляд алгоритмів та структур нейронних мереж

Штучні нейронні мережі (нейромережі або просто мережі) є класом моделей машинного навчання, що ґрунтуються на принципах роботи центральної нервової системи ссавців. Нейронна мережа складається з кількох взаємопов'язаних різних шарів, таких як вхідний шар, щонайменше один прихований шар і вихідний шар (рисунок 1.6). Їх найкраще використовувати при виявленні об'єктів для розпізнавання образів, країв (вертикальні/горизонтальні), форми, кольори та текстури. Приховані шари в згорткових мережах часто є шарами згортки. Основними параметри нейронів є вхідний (синій колір) та вихідний шар (зелений колір) [2], [6].

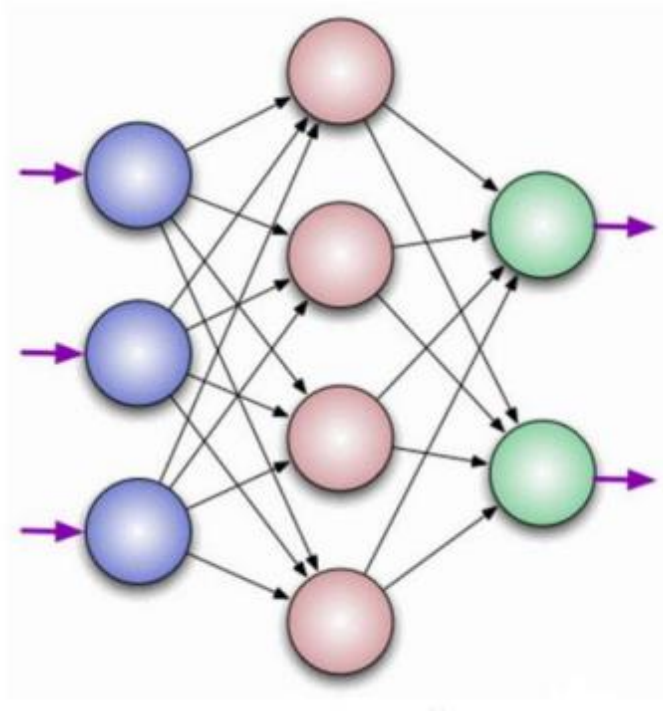


Рисунок 1.6 – Модель нейронної мережі.

З великою кількістю згорткових шарів, щоразу, коли новий вхід відправляється на наступний згортковий шар, він змінюється по-різному (рисунок 1.7). Наприклад, у згортковому шарі фільтр може ідентифікувати форму/колір у певній області, останній згортковий шар може класифікувати об'єкт. У загальному випадку згорткова нейронна мережа складається з

великої кількості шарів. На останніх етапах зазвичай використовується один або кілька повнозв'язкових шарів.

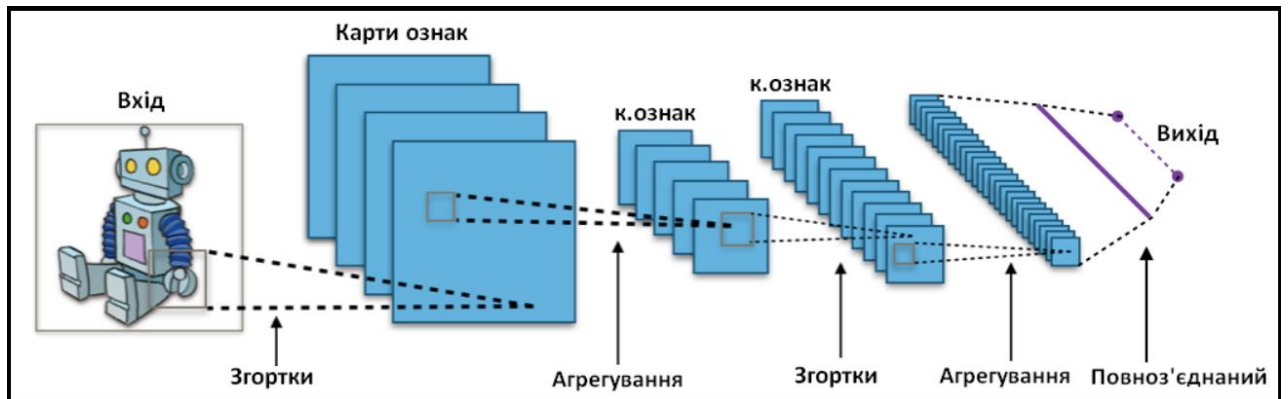


Рисунок 1.7 – Топологія згорткової нейронної мережі.

Згорткові нейронні мережі забезпечують часткову стійкість до змін масштабу, зсувів, поворотів, зміни ракурсу та інших спотворень. Згорткові нейронні мережі поєднуються у три архітектурні ідеї, повороту зсуву та просторових спотворень:

- Локальні рецепторні поля (забезпечують локальну двовимірну зв'язність нейронів).
- Загальні синоптичні коефіцієнти (забезпечують детектування деяких рис у будь-якому місці зображення та зменшують загальну кількість вагових коефіцієнтів).
- Ієрархічна організація із просторовими підвибірками.

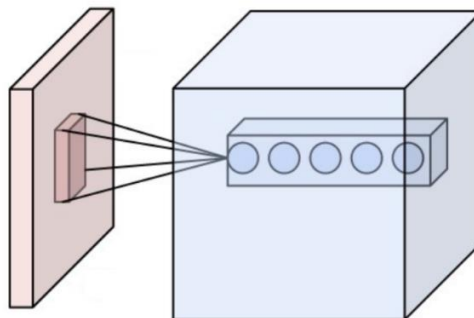


Рисунок 1.7 – Нейрони згорткового шару (синього), з'єднані з їхнім рецептивним полем (червоним).

1.3.1 Розпізнавання об'єктів за допомогою нейронної мережі.

Для того, щоб навчити нейронну мережу виявляти об'єкти на будь-якому зображенні, з приблизно однаковою формою та кольорами, слід застосувати різні фільтри (рисунку 1.9).

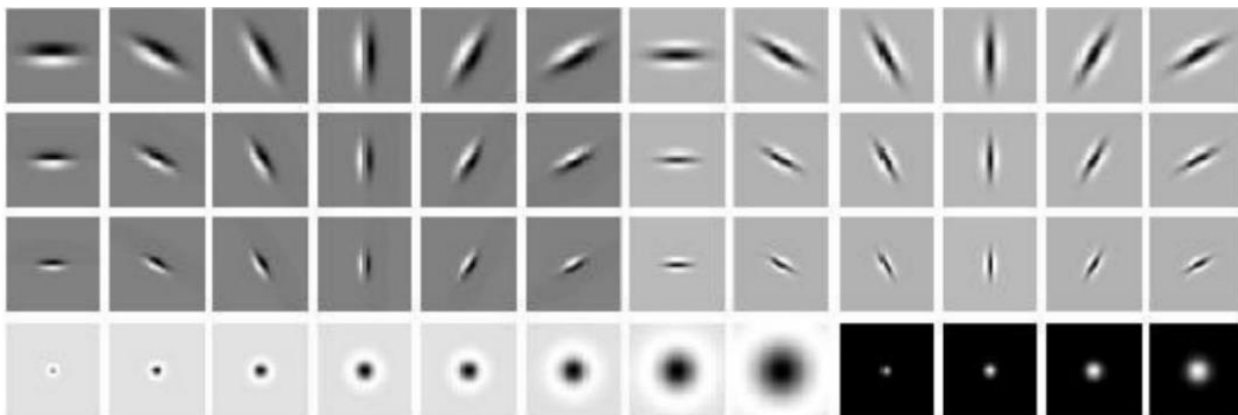


Рисунок 1.9 – Фільтри для різноманітних фрагментів зображення.

Для того, щоб мережам не доводилося окремо розпізнавати об'єкти в різних частинах зображення, ми розділяємо ваги, що відповідають за розпізнавання, між різними фрагментами вихідного зображення. Розглянемо зображення, у якому необхідно не просто виділити об'єкт, а встановити кількісну точність рішення чотирьох фрагментів (рисунку 1.9).

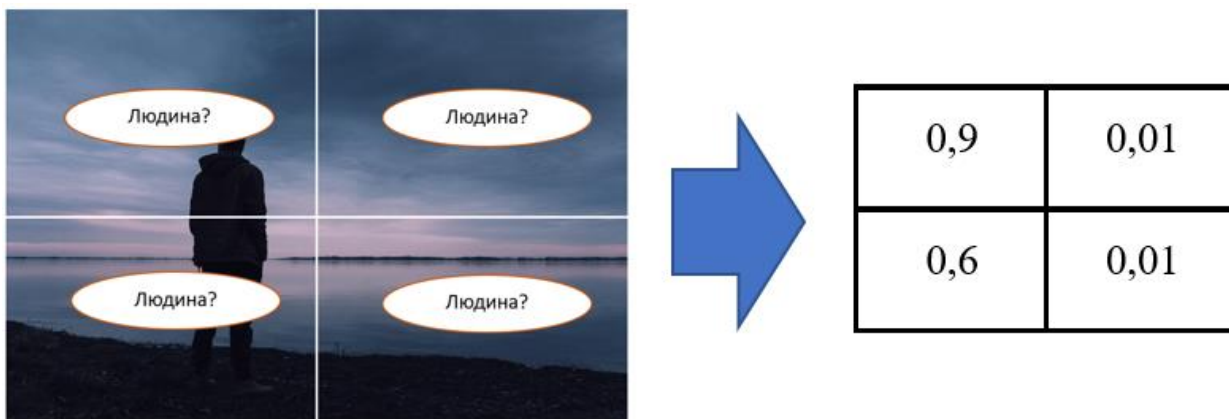


Рисунок 1.10 – Розпізнавання об'єкта та характеристики точності розпізнавання.

У таблиці вказані кількісні характеристики рішення, що показують ступінь точності розпізнавання об'єкта у цьому фрагменті зображення. Знаходимо максимальне значення, яке і буде необхідною характеристикою.

$$\max=\{0.9, 0.3, 0.01, 0.01\}=0.9 \quad (1.2)$$

Основна ідея алгоритму полягає у наступному:

- виконуємо поділ ваг (weight sharing) для створення “фільтруючого вікна”, що пробігає по зображенню;
- застосований до зображення фільтр допомагає виділити фрагменти, важливі для розпізнавання;
- у той час як у традиційному машинному зорі фільтри конструювали вручну, неймережі дозволяють нам сконструювати оптимальні фільтри за допомогою навчання;
- процес фільтрації зображень можна поєднати з обчисленнями в межах нейронної мережі.

1.3.2 Згортковий шар нейронної мережі.

Згортковий шар нейронної мережі – це базовий компонент згорткової нейронної мережі (CNN), що використовується для автоматичного виділення ключових ознак із зображень. У контексті розпізнавання руху та класифікації об'єктів, зокрема людей, згорткові шари дають змогу неймережі самостійно навчитися розпізнавати важливі шаблони: краї, форми, контури фігур тощо [1], [2].

Принцип роботи згорткового шару полягає у «прокатуванні» по зображенню спеціального фільтра (ядра згортки), який обчислює зважену суму пікселів у локальній області. Кожне таке ядро навчається під час тренування мережі та виявляє певний тип ознак. В результаті ми отримуємо

карту ознак, яка підкреслює наявність тих чи інших патернів на зображенні. Кількість карток визначається вимогами до завдання, якщо взяти велику кількість карток, то підвищиться якість розпізнавання, але збільшиться обчислювальна складність.

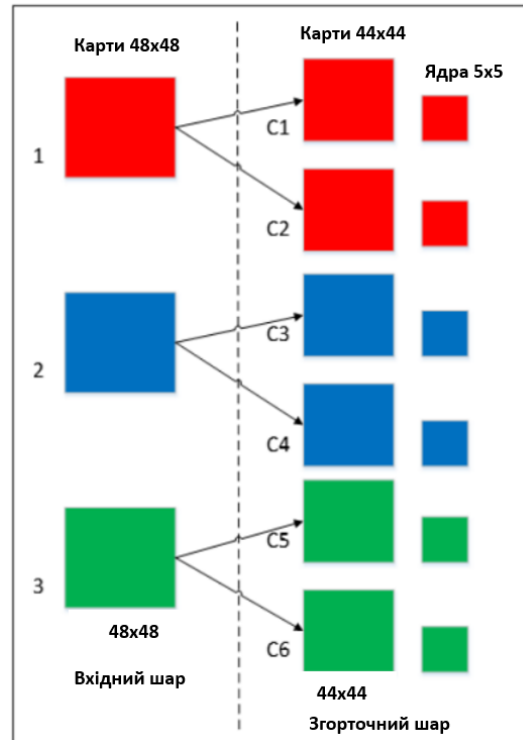


Рисунок 1.10 – Організація зв'язків між картками згорточного та вхідного (попереднього) шару.

Математична операція згортки двовимірного зображення X із ядром K розміром $m \times n$ описується формулою 1.3:

$$Y(i, j) = \sum_{u=0}^{m-1} \sum_{v=0}^{n-1} X(i+u, j+v) \cdot K(u, v) \quad (1.3)$$

де X – вхідне зображення або попередній шар,

K – ядро згортки (набір ваг),

$Y(i, j)$ – результат згортки в точці (i, j) ,

$m \times n$ – розмір фільтра.

У випадку, коли система працює на мікроконтролері з обмеженими обчислювальними ресурсами, згортковий шар є частиною оптимізованої нейромережі, яка повинна бути досить легкою для виконання в реальному часі. Саме тому часто використовуються компактні моделі типу MobileNet, де згорткові операції спрощені або замінені на глибокі згортки.

Після згорткового шару зазвичай йде функція активації (наприклад, ReLU), яка додає нелінійність, та підвибірка (наприклад, max pooling), яка зменшує розмір карти ознак, зберігаючи при цьому найважливішу інформацію.

Таким чином, згортковий шар дозволяє ефективно перетворювати вхідні зображення у набір корисних ознак, які потім можуть бути використані для класифікації – наприклад, визначити, чи є на зображенні людина, чи ні.

1.3.3 Підвибірковий шар.

Підвибірковий шар також, як і згортковий, має карти, але їх кількість збігається з попереднім (згортковим) шаром. Ціль використання даного шару – зменшити розмірності карт попереднього шару. Якщо на попередній операції згортки вже були виявлені деякі ознаки, то для подальшої обробки настільки докладне зображення вже не потрібне і воно ущільнюється до менш докладного. До того ж, фільтрація вже непотрібних деталей допомагає не перевивчатися.

У процесі сканування ядра підвибіркового шару (фільтром) карти попереднього шару, скануюче ядро не перетинається на відміно від згорткового шару. Зазвичай кожна карта має ядро розміром 2×2 , що дозволяє зменшити попередні карти згорткового шару в 2 рази. Вся карта ознак поділяється на комірки 2×2 елементи, у тому числі вибираються максимальні за значенням.

Існує кілька типів підвибірки:

- Max Pooling – обирається максимальне значення в кожному регіоні;
- Average Pooling – обчислюється середнє значення кожного регіону;
- Global Pooling – використовується для скорочення розмірності до одного значення на карту ознак.

Найпоширенішим є Max Pooling, який ефективно виділяє найяскравіші (найінформативніші) ознаки в кожному фрагменті зображення (рисунок 1.11).

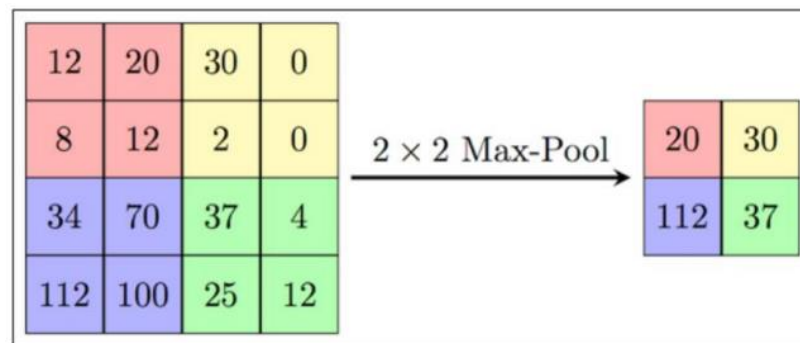


Рисунок 1.11 – Формування нової карти підвибірочного шару на основі попередньої картки згорткового шару. Операція підвибірки (MaxPooling).

У випадку, коли система працює на мікроконтролері з обмеженими обчислювальними ресурсами, підвибіркові шари дозволяють значно зменшити обсяг даних для наступних шарів мережі. Це критично важливо для ефективної роботи компактних моделей, таких як MobileNet, які розгортаються на пристроях з обмеженою пам'яттю та продуктивністю [12], [14].

1.3.4 Повнозв'язковий шар.

Повнозв'язковий шар – це шар нейронної мережі, в якому кожен нейрон з'єднаний з усіма нейронами попереднього шару [1], [2]. Він

відповідає за прийняття високорівневих ознак, що були витягнуті попередніми згортковими та підвибірковими шарами, і перетворює їх у конкретний вихід – наприклад, клас об'єкта.

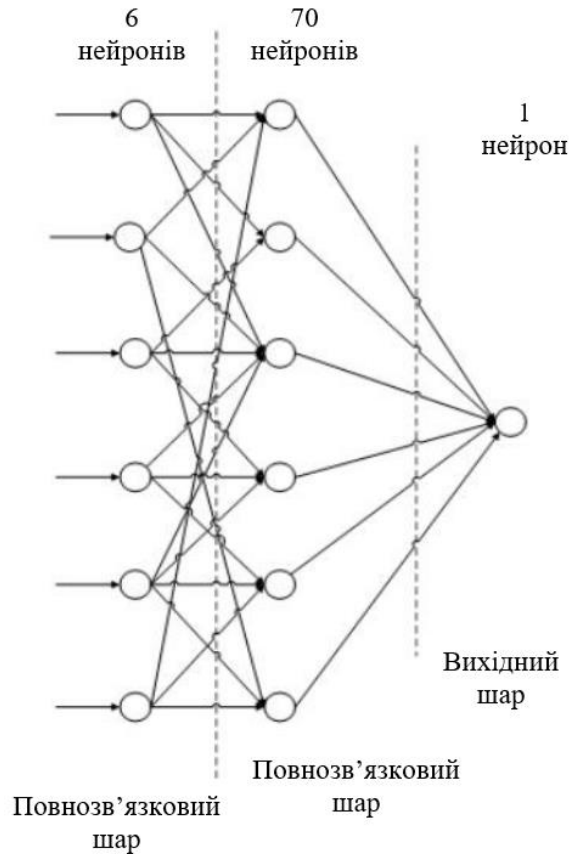


Рисунок 1.12 – Приклад повнозв'язкових шарів.

У повнозв'язковому шарі відбувається матричне множення вхідного вектора ознак на вагову матрицю, після чого додається зсув і застосовується функція активації, як-от ReLU або Softmax (на останньому шарі для класифікації). Формально, вихід обчислюється за формулою 1.4:

$$y = f(Wx + b) \quad (1.4)$$

де x – вхідний вектор ознак,

W – матриця ваг,

b – вектор зміщень.

У системах на базі мікроконтролерів або інших обмежених пристроїв повнозв'язкові шари часто використовуються тільки наприкінці моделі та мають обмежену кількість нейронів. Це дозволяє зменшити розмір моделі та забезпечити її швидке виконання. У таких випадках повнозв'язковий шар виконує заключну роль – класифікує, наприклад, чи є на зображенні людина, чи ні.

1.4 Аналіз апаратних платформ для реалізації нейронних мереж

Реалізація нейронних мереж на апаратному рівні є важливим етапом, оскільки правильний вибір платформи забезпечує високу ефективність і швидкодію, особливо при роботі з великими обсягами даних або на пристроях з обмеженими обчислювальними ресурсами [1], [2], [14]. Існують різні варіанти апаратних платформ для виконання нейронних мереж, від загальнодоступних процесорів до спеціалізованих акселераторів [6], [12].

Розглянемо такі основні платформи що використовуються для реалізації згорткових нейронних мереж (CNN) та інших типів нейронних мереж:

- центральні процесори (Central Processing Unit (CPU)),
- графічні процесори (Graphics Processing Unit (GPU)),
- спеціалізовані чіпи для машинного навчання (Tensor Processing Unit (TPU), Neural Processing Unit (NPU)),
- мікроконтролери та вбудовані платформи.

Нижче більш детально розглянемо ці платформ виявимо які їх основні переваги та в чому їх недоліки.

1. Центральні процесори (CPU). Центральні процесори (CPU) є традиційними обчислювальними платформами для виконання нейронних мереж. Вони широко використовуються завдяки своїй універсальності та можливості виконувати різноманітні завдання. CPU зазвичай добре підходять

для невеликих або середніх моделей нейронних мереж, а також для прототипування.

Серед основних переваг центральних процесорів можна виділити їхню універсальність і доступність, що дозволяє застосовувати їх у багатьох галузях і завданнях. CPU добре підходять для роботи з невеликими моделями нейронних мереж, а також для навчання на обмежених обсягах даних. Крім того, вони мають широку підтримку з боку популярних бібліотек і фреймворків машинного навчання, таких як TensorFlow та PyTorch, що спрощує розробку і тестування моделей.

Втім, центральні процесори мають і низку недоліків. Вони характеризуються обмеженою пропускнуою здатністю при виконанні паралельних обчислень, що знижує ефективність при роботі з великими моделями. Також CPU не є оптимальними для обчислень у реальному часі, де важлива швидкодія. Ще одним мінусом є відносно високе енергоспоживання під час виконання складних обчислювальних задач, що може бути критичним для систем з обмеженими ресурсами.

Таким чином, хоча центральні процесори є хорошим вибором на етапі розробки, тестування або для виконання простих завдань, їх використання на стадії розгортання у складних проектах або вбудованих системах із жорсткими обмеженнями ресурсів може бути неефективним.

2. Графічні процесори (GPU). Графічні процесори (GPU) стали одними з найпоширеніших платформ для навчання та виконання нейронних мереж завдяки своїй здатності до високоефективного виконання паралельних обчислень. GPU оптимізовані для виконання однотипних операцій на великих наборах даних, що є основною вимогою для нейронних мереж.

Серед головних переваг GPU варто зазначити їхню високу пропускну здатність та здатність до паралельної обробки великої кількості обчислень, що дає змогу значно прискорити процес навчання моделей порівняно з використанням CPU. Крім того, графічні процесори підтримуються широким

спектром популярних фреймворків, таких як TensorFlow, PyTorch, Caffe, що дозволяє розробникам ефективно реалізовувати навіть дуже складні архітектури нейронних мереж.

Однак, незважаючи на велику потужність, GPU мають і певні недоліки. По-перше, вони зазвичай є доволі дорогими, що може стати серйозним обмеженням у випадку проєктів із невеликим бюджетом. По-друге, графічні процесори споживають велику кількість енергії, що є критичним фактором у системах, де важлива енергоефективність. Також для їх роботи потрібно багато оперативної пам'яті, оскільки великі моделі нейронних мереж і обсяги вхідних даних займають значні ресурси.

Таким чином, GPU є ідеальним вибором для навчання великих моделей на потужних серверних платформах або в хмарних середовищах. Водночас їх застосування у вбудованих або мобільних системах із обмеженими ресурсами часто є недоцільним через енергоспоживання, вартість і вимоги до пам'яті.

3. Спеціалізовані чіпи для машинного навчання (TPU, NPU). Для більш ефективного виконання нейронних мереж були розроблені спеціалізовані процесори, зокрема тензорні процесори (TPU) від Google та нейронні процесори (NPU) від різних виробників. Ці апаратні рішення спеціально оптимізовані для виконання обчислень, характерних для глибокого навчання, особливо матричних операцій, які є основою роботи більшості нейронних мереж.

Тензорні процесори (TPU) – це спеціалізовані апаратні прискорювачі, створені компанією Google, головним чином для обробки великих обсягів даних у процесі роботи з глибокими нейронними мережами. TPU реалізують спеціальні блоки для виконання матричних операцій, що значно підвищує швидкість обчислень у порівнянні з традиційними процесорами. Завдяки цьому TPU демонструють високу ефективність у роботі з великими

моделями та глибокими нейронними мережами, особливо в умовах хмарної інфраструктури.

Попри значні переваги, такі як висока швидкість і продуктивність, TPU мають обмеження. Основним недоліком є те, що вони не призначені для локального використання на кінцевих пристроях і найчастіше застосовуються на серверах або в хмарних обчислювальних середовищах, де забезпечується повна реалізація їхніх можливостей.

Нейронні процесори (NPU), у свою чергу, були розроблені для ефективного виконання нейронних мереж безпосередньо на пристроях з обмеженими обчислювальними ресурсами, таких як смартфони, мікроконтролери або вбудовані системи. Вони спеціально адаптовані для роботи з невеликими обсягами даних, що дає змогу реалізовувати різноманітні додатки штучного інтелекту на кінцевих пристроях без необхідності передавати дані на сервер.

Перевагами NPU є висока енергоефективність у порівнянні з GPU та можливість оптимального використання в умовах обмежених ресурсів. Проте існують і недоліки, зокрема обмежена кількість підтримуваних фреймворків і алгоритмів, а також менша гнучкість порівняно з більш універсальними обчислювальними платформами. Це означає, що використання NPU зазвичай обмежується вузьким колом завдань та моделей.

4. Мікроконтролери та вбудовані платформи. Для реалізації нейронних мереж на вбудованих пристроях, таких як мікроконтролери (наприклад, ESP32 або Raspberry Pi), використовуються спеціалізовані платформи та оптимізовані моделі. Такі системи зазвичай мають обмежені апаратні ресурси, зокрема невелику оперативну пам'ять і невисоку обчислювальну потужність, тому ефективне впровадження нейронних мереж на них вимагає попередньої оптимізації моделей, що забезпечить необхідну швидкодію в реальному часі.

Однією з найпоширеніших платформ є ESP32-CAM – компактний модуль з камерою на базі мікроконтролера ESP32. Цей модуль оптимізований для виконання базових завдань комп'ютерного зору, зокрема для виявлення людей з використанням згорткових нейронних мереж. Іншою популярною платформою є Raspberry Pi, яка підтримує більш широкий набір функцій і дає змогу використовувати TensorFlow Lite для запуску оптимізованих моделей нейронних мереж.

Серед головних переваг цих платформ варто відзначити їхню компактність, низьку вартість та здатність працювати на пристроях з обмеженими ресурсами. Завдяки цьому вони дозволяють реалізувати прості моделі в умовах реального часу навіть без потужного обладнання.

Разом з тим, такі платформи мають і певні обмеження. Зокрема, вони не здатні ефективно обробляти складні моделі або великі обсяги даних через обмежену продуктивність. У цьому контексті вибір апаратної платформи залежить від конкретних вимог до задачі. Якщо пріоритетом є швидкість обробки або навчання складних моделей, доцільно обирати потужніші рішення, як-от GPU або TPU. Натомість для вбудованих систем, де важливі енергоефективність і компактність, краще підходять мікроконтролери та спеціалізовані платформи, здатні забезпечити базову функціональність нейронних мереж у реальному часі.

1.5 Висновок до першого розділу

У першому розділі було розглянуто основні аспекти нейронних мереж і їх застосування в задачах комп'ютерного зору. Спочатку було проведено огляд принципів роботи нейронних мереж, акцентуючи увагу на їх здатності до автоматичного навчання та виявлення закономірностей у великих обсягах даних. Далі, було визначено роль нейронних мереж у контексті задач комп'ютерного зору, зокрема, для розпізнавання рухомих об'єктів, таких як люди, на зображеннях, що є важливим для систем безпеки.

Аналіз існуючих методів розпізнавання рухомих об'єктів показав, що найбільш ефективними є згорткові нейронні мережі (CNN), які здатні обробляти зображення та виявляти важливі ознаки об'єктів. Для розпізнавання людей на зображеннях важливим аспектом є вибір правильної архітектури нейронної мережі, що дозволяє досягти високої точності та ефективності в реальному часі. Зокрема, для задачі виявлення людей на обмежених апаратних ресурсах було обрано ефективні моделі CNN, оптимізовані для роботи в реальному часі з низькими вимогами до ресурсів, такі як MobileNet або TinyYolo.

Особливо важливим є вибір платформи для реалізації цієї системи. Для роботи з нейронними мережами на обмежених пристроях було вибрано модуль ESP32-CAM. Цей модуль має вбудовану камеру, що дозволяє здійснювати захоплення зображень без необхідності додаткових периферійних пристроїв. Оскільки ESP32-CAM підтримує мікроконтролер ESP32 з достатньою обчислювальною потужністю та можливістю підключення до Wi-Fi, цей пристрій є ідеальним для створення компактних систем безпеки. ESP32-CAM також є низько вартісним і споживає менше енергії порівняно з іншими варіантами, що робить його чудовим вибором для реальних вбудованих систем з обмеженими ресурсами.

Загалом, нейронні мережі є потужним інструментом для вирішення задач комп'ютерного зору, а правильний вибір апаратної платформи є важливим для забезпечення ефективності та швидкодії системи, особливо при роботі в реальному часі на обмежених ресурсах.

2 ПРОЕКТУВАННЯ СИСТЕМИ БЕЗПЕКИ З АВТОМАТИЧНИМ РОЗПІЗНАВАННЯ РУХОМИХ ОБ'ЄКТІВ

2.1 Вибір апаратної платформи та її особливості ESP32-S3-CAM

Для реалізації системи безпеки з автоматичним розпізнаванням рухомих об'єктів було обрано апаратну платформу ESP32-S3-CAM, яка базується на модулі ESP32-S3-WROOM-1. Цей модуль є покращеною версією популярного ESP32-CAM і має розширені обчислювальні можливості, що робить його ідеальним рішенням для застосування в задачах комп'ютерного зору, обробки зображень та реалізації згорткових нейронних мереж [15], [16], [17], [18], [19].

Технічні характеристики ESP32-S3-CAM:

- контролер: двоядерний 32-бітний процесор Xtensa® LX7 з частотою до 240 МГц;
- пам'ять:
 - 384 КБ ПЗП (ROM);
 - 512 КБ ОЗП (SRAM);
 - 16 МБ псевдостатичної пам'яті (PSRAM);
- робоча напруга: від 3 В до 3,6 В;
- до 45 GPIO (універсальних цифрових входів/виходів);
- діапазон робочих температур: від -40 до +65 °С;
- радіоінтерфейс:
 - Wi-Fi: до 150 Мбіт/с, частота 2.4–2.5 ГГц;
 - bluetooth 5.0 LE, підтримка BLE Mesh, швидкість до 2 Мбіт/с;
- камера: 8/16-бітний DVP, підтримка форматів RGB/YUV;
- USB: Full-Speed USB 2.0 OTG (вбудований PHY).

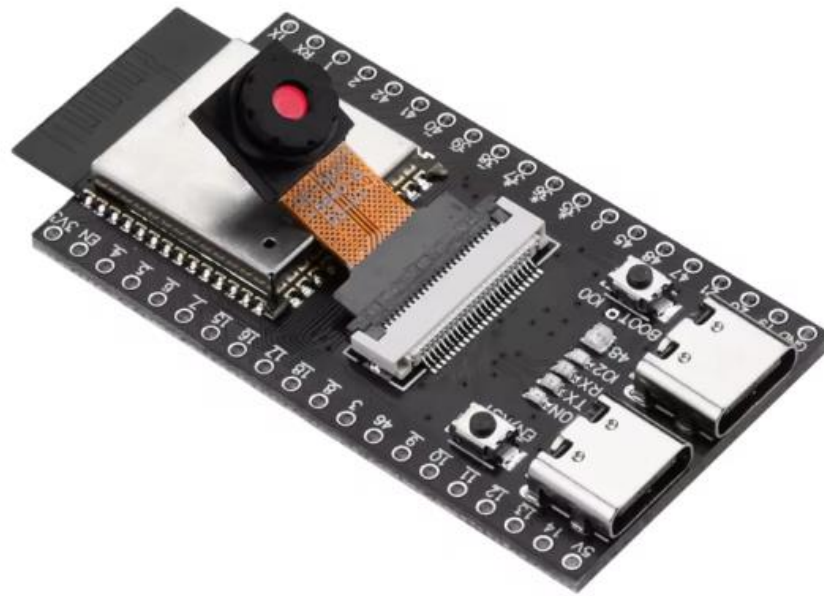


Рисунок 2.1 – Зображення плати ESP32-S3-CAM

Однією з головних причин вибору ESP32-S3-CAM є його висока обчислювальна потужність. Мікроконтролер оснащений двома 32-бітними ядрами Xtensa LX7, що працюють на частоті до 240 МГц, а також підтримує SIMD-інструкції (Single Instruction, Multiple Data), які значно пришвидшують виконання алгоритмів обробки зображень та нейронних мереж.

Також модуль має інтегроване прискорення для нейронних мереж. ESP32-S3 сумісний з TensorFlow Lite for Microcontrollers – спеціалізованою версією бібліотеки TensorFlow для пристроїв з обмеженими ресурсами. Це дозволяє запускати компактні згорткові нейронні мережі прямо на мікроконтролері, без необхідності у зовнішньому обчислювальному блоці.

Модуль має широкі можливості підключення периферії завдяки великій кількості доступних інтерфейсів. До 45 GPIO-портів, підтримка UART, SPI, I2C, I2S, PWM, ADC, DAC, SDIO та USB-OTG дозволяють під'єднувати різноманітні додаткові пристрої – від світлодіодних індикаторів і реле до датчиків руху, температури, GSM або LoRa-модулів. Це відкриває великі можливості для розширення функціоналу системи безпеки.

Ще одним важливим фактором є збільшений обсяг оперативної пам'яті, який включає до 512 КБ внутрішньої SRAM та до 16 МБ зовнішньої PSRAM. Такий обсяг пам'яті дозволяє зберігати кілька кадрів для буферизації, проводити їхню попередню обробку та передавати вхідні дані до нейронної мережі без втрати продуктивності.

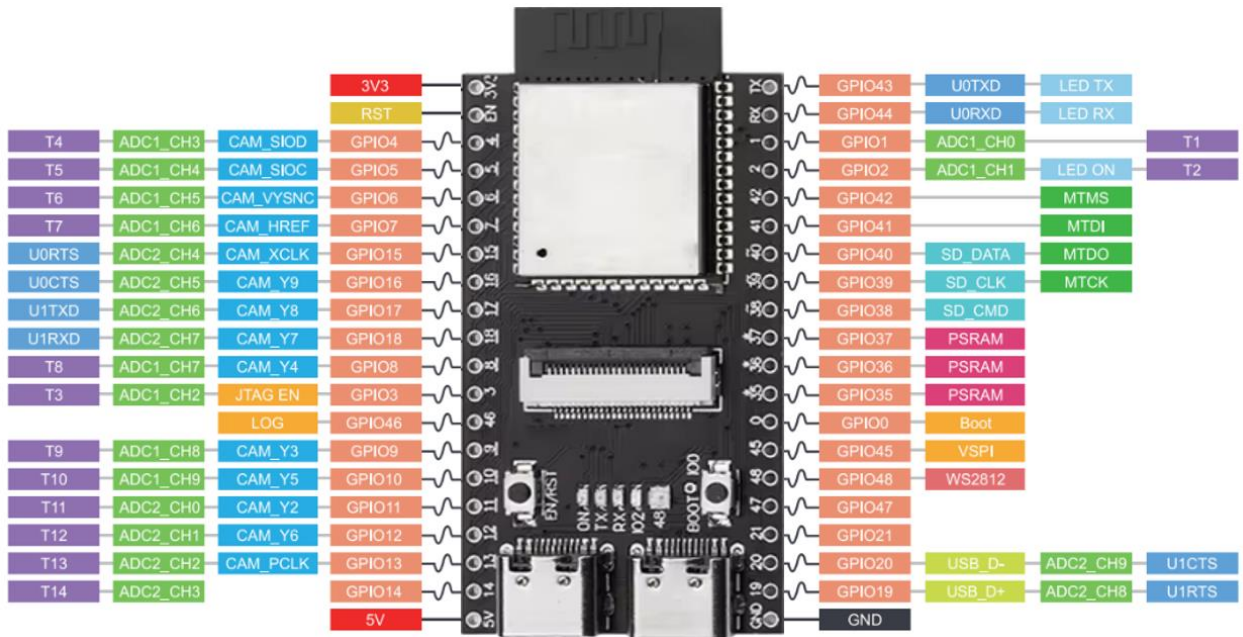
Для реалізації функції візуального моніторингу ESP32-S3-CAM підтримує підключення цифрових камер через інтерфейс DVP, зокрема модулі OV2640 або OV5640, з частотою до 40 МГц. Це дозволяє отримувати зображення з високою швидкістю для подальшої обробки.

Крім того, ESP32-S3-CAM може працювати в умовах низької освітленості при наявності інфрачервоного підсвічування, яке легко підключається через GPIO, або завдяки використанню камер з високою чутливістю до інфрачервоного спектру.

Наявність вбудованого Wi-Fi (802.11 b/g/n) та Bluetooth 5.0 (LE) дозволяє організувати бездротовий обмін даними між пристроєм та користувачем або сервером. Це дає змогу реалізувати сповіщення про події, оновлення прошивки через інтернет або інтеграцію з мобільними додатками.

ESP32-S3 підтримує безпечне OTA-оновлення (Over-The-Air), протоколи захисту WPA/WPA2, а також мережеві протоколи TLS, MQTT, TCP/IP тощо. Це забезпечує як безпеку комунікації, так і масштабованість системи, дозволяючи реалізовувати розподілені або хмарні рішення.

Таким чином, ESP32-S3-CAM поєднує високу функціональність, енергоефективність та підтримку нейромереж, що робить його ідеальною платформою для компактних систем безпеки з елементами штучного інтелекту.



- PWD Power
- GND Ground
- Serial Serial for Debugging/Programming
- ADCX_CH Analog-to-Digital Converter
- RESET Reset the chip
- GPIOX GPIO Input and Output
- TOUCHX Touch Sensor Input Channel
- STRAP Strapping Pin Functions
- SD On-board SD Card Pin
- LED On-board LED Pin
- WS2812 On-board WS2812 Pin
- PSRAM Built-in expansion memory chip pin
- USB USB Function Pin
- CAMERA Camera Pin
- JTAG Jtag for Debugging
- PWM Capable Pin

Рисунок 2.2 – Схема модуля (роspіновка)

2.2 Опис архітектурної та структурної схеми системи безпеки

Розроблена система безпеки базується на вбудованій платформі ESP32-S3-CAM, яка виконує функцію збору відеоінформації з навколишнього середовища. Основним елементом архітектури є модуль ESP32-S3-CAM, що

поєднує в собі мікроконтролер і камеру, завдяки чому забезпечується компактність та енергоефективність. Камера захоплює відео у режимі реального часу, після чого зображення обробляються безпосередньо на пристрої за допомогою попередньо натренованої згорткової нейронної мережі, оптимізованої для виявлення людей [15], [20].

У разі виявлення рухомого об'єкта, система активує тривожний механізм, який може включати світлову або звукову сигналізацію, а також надсилання повідомлення на мобільний пристрій користувача через Wi-Fi. Передача даних здійснюється за допомогою бездротового з'єднання, що дозволяє віддалено контролювати події в зоні спостереження.

Для забезпечення автономної роботи навіть у разі зникнення основного живлення, у систему інтегровано блок живлення з резервним акумулятором. Акумулятор дозволяє підтримувати працездатність пристрою протягом тривалого часу після вимкнення електроенергії. У разі зникнення живлення з основного джерела, система автоматично переходить на живлення від акумулятора та надсилає сигнал тривоги користувачеві про втрату електроживлення.

Архітектурно системи складається з таких основних функціональних блоків, кожен з яких виконує окрему роль у процесі автоматичного виявлення рухомих об'єктів (рисунки 2.3):

- блок збору зображень (камера);
- блок попередньої обробки (фрейм-граббер і буфер);
- обчислювальний блок (нейронна мережа на ESP32);
- блок прийняття рішень;
- блок зворотного зв'язку (модуль повідомлення або сигналізації);
- блок живлення з можливістю резервного енергозабезпечення.



Рисунок 2.3 – Архітектури системи захисту на базі модуля ESP32-S3-CAM.

Блок збору зображень (камера) – це модуль ESP32-CAM, оснащений вбудованою камерою OV2640, яка забезпечує захоплення відеопотоку з навколишнього середовища. Камера працює в режимі реального часу та підтримує передачу кадрів у різних форматах (наприклад, JPEG або RGB), з можливістю зміни роздільної здатності. У темряві система може використовувати інфрачервоне підсвічування (IR LED), що дозволяє здійснювати нічне спостереження завдяки високій світлочутливості сенсора OV2640 до ІЧ-випромінювання.

Блок попередньої обробки (фрейм-граббер і буфер) відповідає за початкову підготовку відеоданих до аналізу. Фрейм-граббер програмно витягує окремі кадри з відеопотоку, знижуючи частоту кадрів для зменшення навантаження на систему. Буфер, у свою чергу, тимчасово зберігає захоплені кадри, забезпечуючи їх синхронізовану передачу до обчислювального блоку.

У цьому блоці також можуть виконуватись попередні операції, такі як зміна розміру зображення, зменшення кольорової глибини або стиск, що дозволяє заощадити ресурси пам'яті та процесора.

Обчислювальний блок (нейронна мережа на ESP32) виконує основне завдання – аналіз зображень з метою виявлення рухомих об'єктів, у даному випадку людей. Тут розміщена згортова нейронна мережа (CNN), оптимізована для роботи на мікроконтролері. Обчислення виконуються безпосередньо на ESP32, що дозволяє уникнути потреби у зовнішньому сервері.

Блок прийняття рішень аналізує вихід нейронної мережі та визначає, чи слід активувати певну дію. Якщо в кадрі виявлено людину або інший об'єкт, що викликає тривогу, цей блок генерує сигнал для зворотного зв'язку. При відсутності зворотного зв'язку приймає рішення для запису відео на SD носій, а при відновленні зворотного зв'язку переслати відеофайл. Він також може реалізовувати просту логіку для зниження кількості хибних спрацьовувань, наприклад, ігнорування незначних рухів або періодичне підтвердження наявності об'єкта.

Блок радіо зв'язку (модуль повідомлення або сигналізації) відповідає за інформування користувача про події. Він може включати модуль Wi-Fi для надсилання push-повідомлень, електронної пошти, або активацію світлової/звукової сигналізації (наприклад, сирени або світлодіодів). При підключенні до мережі Інтернет повідомлення можуть пересилатися на смартфон або іншій пристрій користувача.

Блок живлення з функцією резервного енергозабезпечення дасть можливість забезпечити стабільну роботу системи відеоспостереження в умовах можливих перебоїв у зовнішньому електропостачанні. Для блока живлення була розроблена схема яка дозволяє автоматично перемикатися між зовнішнім джерелом живлення та акумулятором, без зупинки роботи пристрою.

Схема системи живлення приведена нижче (рисунок 2.4).

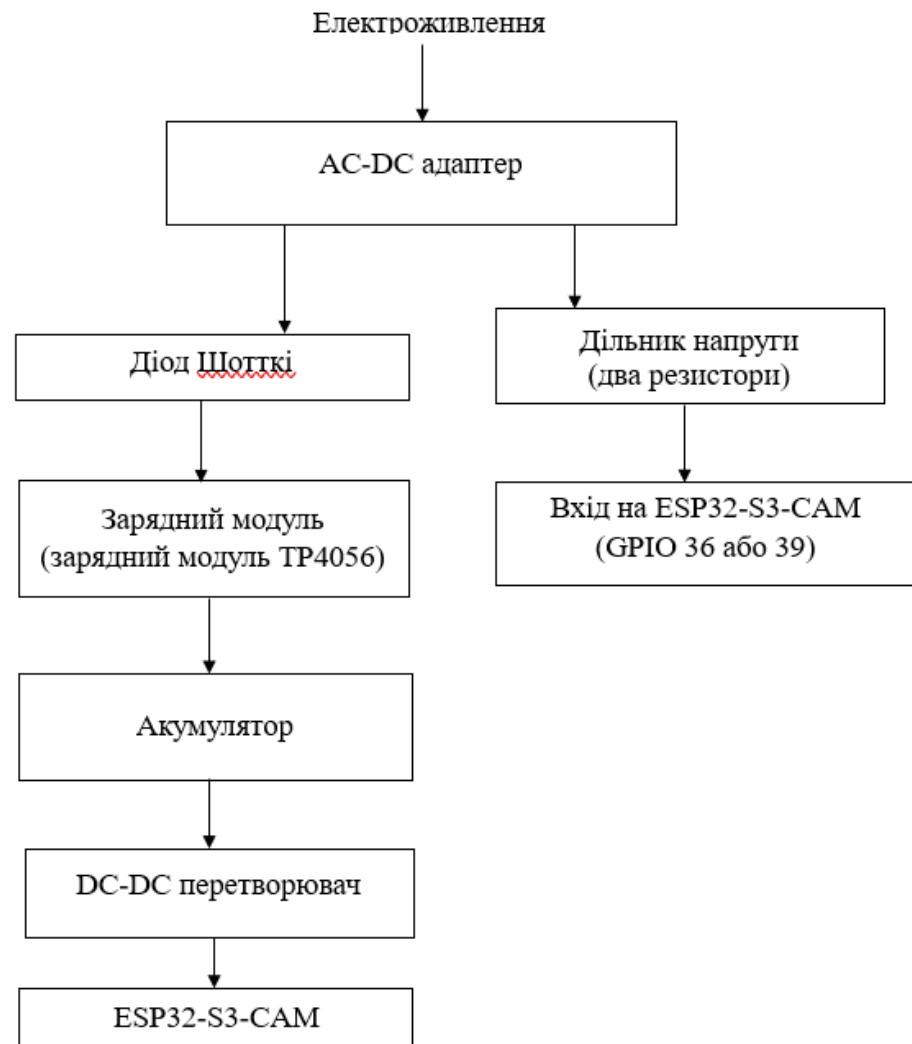


Рисунок 2.4 – Схема системи живлення

У якості джерела живлення використовується мережа змінного струму 220 В, яка через AC-DC адаптер перетворюється на 5 В постійного струму. Після стабілізації напруга подається одночасно на зарядний модуль TP4056 та на дільник напруги, підключений до одного з аналогових входів (ADC) ESP32-S2-CAM.

Зарядний модуль TP4056 відповідає за живлення літій-іонного акумулятора (3,7 В) та контролює процес зарядки, забезпечуючи захист від перезарядження, глибокого розряду та короткого замикання. Після

акумулятора в схемі використовується DC-DC перетворювач, що стабілізує напругу до необхідного рівня 3,3 В (або 5 В) для живлення ESP32.

Для запобігання зворотному току у випадку зникнення зовнішнього живлення до схеми включено діод Шотткі (SS14 або 1N5819), встановлений між виходом AC-DC адаптера та зарядним модулем. Він забезпечує автоматичне перемикання на акумуляторне живлення без потреби у релейних або електронних ключах.

Щоб ESP32 міг визначати факт зникнення зовнішнього живлення, після стабілізатора додано резисторний дільник напруги, вихід якого підключено до ADC-порту (GPIO 36). У разі присутності живлення на виході стабілізатора (5 В), дільник видає на вхід ESP32 безпечне значення напруги. Для цього використовуються резистори: $R1 = 10$ кОм (верхній) та $R2 = 6,8$ кОм (нижній), що формують напругу близько 2,02 В, яка знаходиться в межах допустимого діапазону для ADC ESP32 ($\leq 3,3$ В). Якщо зовнішнє живлення зникає, напруга на виході дільника стає близькою до нуля, що дозволяє ESP32 зафіксувати подію і відповідним чином реагувати (надсилання сповіщення, перехід у режим енергозбереження тощо).

У системі застосовується акумулятор типу Li-Ion 3,7 В ємністю приблизно 10000 мА·год, що забезпечує тривалу автономну роботу. За умови середнього енергоспоживання системи в межах 160 – 180 мА у звичайному режимі роботи (камерне опитування, обчислення моделі, передача по Wi-Fi, DC-DC перетворювач), теоретичний час автономного функціонування становить близько від 50 до 60 годин, це буде більше 2 діб безперервної роботи.

Завдяки такій структурі система є автономною, економічною та легко масштабованою, що дозволяє використовувати її для різних сценаріїв охоронного моніторингу, включаючи контроль доступу, виявлення вторгнень або автоматичне сповіщення про несанкціонований рух та перебої в електропостачанні.

2.3 Вибір алгоритму розпізнавання рухомих об'єктів

Для реалізації функції автоматичного виявлення руху в системі безпеки було обрано метод віднімання фону, оскільки він демонструє просту логіку, високу швидкість обробки кадрів і чудово підходить для реалізації на пристроях із обмеженими апаратними ресурсами, таких як ESP32-S3-CAM.

Першим кроком у обробці зображення є перетворення формату RGB у зображення у відтінках сірого. Це перетворення значно знижує навантаження на мікроконтролер, адже замість трьох кольорових каналів обробляється лише один. Процес виконується за формулою 2.1:

$$K = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B \quad (2.1)$$

де K – значення пікселя в градаціях сірого,

R, G, B – значення кольорових каналів.

Такий підхід дозволяє сконцентрувати увагу системи на змінах яскравості, які найбільш інформативні для виявлення руху, та підвищує загальну стійкість до помилкових спрацювань, спричинених змінами кольору чи освітлення.

При використанні камери модуля ESP32-S3-CAM із сенсором OV2640 або OV5640, можливе отримання зображень з максимальною роздільною здатністю до 1600×1200 пікселів (UXGA). У такому форматі кадр має високу деталізацію, що дозволяє виявляти навіть дрібні об'єкти на сцені або визначати контури фігур з більшою точністю. Проте, така роздільна здатність вимагає суттєвого обсягу пам'яті для зберігання кадру.

Розрахуємо об'єм пам'яті необхідний на один кадр як у в форматі RGB (3 байти на піксель) за формулою 2.2 так і у відтінках сірого за формулою 2.3:

$$M = X \times Y \times 3 = 1600 \times 1200 \times 3 = 5\,760\,000 \text{ байт} \quad (2.2)$$

$$M = X \times Y \times 1 = 1600 \times 1200 \times 1 = 1\,920\,000 \text{ байт} \quad (2.3)$$

де M – Об'єм кадру (байт),

X – ширина кадру в пікселях,

Y – висота кадру в пікселях.

Тобто один кадр у форматі RGB займає приблизно $\approx 5,6$ МБ, а у відтінках сірого $\approx 1,8$ МБ. Через це в системах із обмеженими апаратними ресурсами, таких як ESP32-S3, максимальна роздільна здатність використовується переважно для фотозйомки або локального збереження зображень, тоді як для потокової обробки (наприклад, виявлення руху чи розпізнавання об'єктів) оптимально використовувати зменшені формати, це дозволяє знизити навантаження на оперативну пам'ять і прискорити обробку зображень у реальному часі.

З урахуванням апаратних обмежень ESP32-S3-CAM та необхідності буферизації кадрів, було вибрано оптимальний розмір кадру – 320×240 пікселів (QVGA). Один кадр у градаціях сірого займає приблизно 76,8 КБ пам'яті, що дозволяє зберігати його у внутрішній або псевдостатичній пам'яті мікроконтролера. Такий розмір також забезпечує достатню деталізацію сцени для подальшої нейронної обробки.

Алгоритм віднімання фону ґрунтується на порівнянні поточного кадру з фоновим зображенням для виявлення змін у сцені. Фонове зображення може бути сформоване двома способами: заздалегідь (статичне) або динамічно, на основі попередніх кадрів.

Фонове зображення, сформоване заздалегідь, є фіксованою опорною сценою, зазвичай знятою у момент, коли в кадрі не було рухомих об'єктів. Його основна перевага – стабільність: воно не змінюється, що дозволяє легко виявляти навіть незначні переміщення об'єктів. Такий підхід ефективний у приміщеннях зі сталою сценою (наприклад, склад, коридор, кімната), де фон

не змінюється з часом. Проте його недоліком є чутливість до змін умов освітлення, появи нових статичних об'єктів або зміни положення камери – все це призводить до хибних спрацьовувань.

У свою чергу, фонове зображення, яке формується динамічно на основі попередніх кадрів, адаптується до змін у сцені. Це дозволяє враховувати поступові зміни освітлення, легке коливання об'єктів (наприклад, дерев або штор), або поява нових елементів, які з часом вважаються частиною фону. Такий метод підвищує гнучкість системи, зменшує кількість помилкових детекцій і краще працює у змінному середовищі. Водночас його мінусом є те, що повільні об'єкти або ті, що надовго залишаються в кадрі, можуть бути “поглинуті” фоном, що ускладнює їх подальше виявлення.

Коли абсолютна різниця значень пікселів між поточним кадром і фоновим перевищує заданий поріг, система вважає, що в цій частині сцени відбулося переміщення. На основі цієї інформації створюється бінарна маска, де пікселі зі змінами мають біле значення, а решта – чорне. Маска дає змогу виділити області руху та надалі використовувати їх, наприклад, для побудови контурів об'єктів або передачі вхідних даних нейронній мережі для класифікації.

2.4 Вибір нейронної мережі

У процесі проектування системи безпеки з функцією автоматичного розпізнавання рухомих об'єктів особливу увагу було приділено вибору нейронної мережі, яка б відповідала технічним обмеженням апаратної платформи ESP32-S3-CAM і водночас забезпечувала достатню точність класифікації. Основна задача нейромережі полягає у виявленні рухомого об'єкта в кадрі та визначенні, чи є ним людина. У разі позитивного результату система реагує відповідним чином – спрацьовує сигналізація або надсилається сповіщення. Якщо ж виявлений об'єкт виявляється твариною,

птахом, тінню або іншим незначущим елементом – система ігнорує його, що знижує кількість хибних спрацювань.

З огляду на обмежений обсяг оперативної пам'яті ESP32-S3-CAM (до 512 КБ SRAM + PSRAM) та помірні обчислювальні можливості, для реалізації задач глибинного навчання було обрано фреймворк TensorFlow Lite for Microcontrollers (TFLM). Це спеціальна версія популярного TensorFlow, розроблена Google для запуску оптимізованих моделей на пристроях із обмеженими ресурсами. TFLM дозволяє створювати та квантувати моделі, перетворюючи їх у формат, придатний для роботи в мікроконтролерному середовищі без необхідності зовнішнього процесора або сервера.

У рамках реалізації було обрано легку згорткову нейронну мережу (CNN), побудовану на основі спрощеного варіанта MobileNetV1. Ця модель адаптована під TFLM та складається з 2–3 згорткових шарів, шарів підвибірки (MaxPooling) і одного-двох повнозв'язкових шарів. Використання глибинних згорток (depthwise separable convolutions) дозволяє значно зменшити кількість обчислень, зберігаючи при цьому ефективність виявлення.

Основні переваги цієї нейронної архітектури полягають у низькому споживанні пам'яті – після квантування її розмір не перевищує 250 КБ, що дозволяє зберігати її у внутрішній пам'яті ESP32-S3-CAM. Час inference (обробки одного кадру) становить лише 200–300 мс, що дозволяє виконувати класифікацію в реальному часі. Повна сумісність з ESP32-S3-CAM дозволяє запускати модель автономно без використання додаткових модулів.

Навчання нейронної мережі виконується на зовнішньому комп'ютері з використанням TensorFlow, після чого модель конвертується у формат .tflite, проходить квантування і прошивається в мікроконтролер. В системі безпеки ESP32-S3-CAM використовує метод віднімання фону для виявлення зони руху, а вже після цього передає цю зону нейронній мережі для уточнення –

чи є об'єкт людиною. Такий підхід дозволяє поєднати ефективність класичних алгоритмів з точністю нейромереж.

Таким чином, обрана нейронна мережа на базі TensorFlow Lite Micro забезпечує оптимальний баланс між продуктивністю, точністю та ефективністю, дозволяючи реалізувати потужну систему розпізнавання людей навіть на мікроконтролерній платформі ESP32-S3-CAM.

2.5 Висновок до другого розділу

У другому розділі було здійснено детальне проектування системи безпеки з автоматичним виявленням і розпізнаванням рухомих об'єктів на базі мікроконтролерної платформи ESP32-S3-CAM. Було обґрунтовано вибір апаратної платформи з урахуванням її високої обчислювальної потужності, підтримки нейронних обчислень, енергоефективності та широких можливостей підключення периферійних пристроїв. Здійснено архітектурний аналіз системи та описано її структурну реалізацію, включаючи всі основні блоки – від захоплення зображення до живлення з резервним акумулятором і повідомлення користувача.

Для виявлення руху в системі було обрано метод віднімання фону, який є простим у реалізації та ефективним з погляду продуктивності на пристроях із обмеженими ресурсами. Проведено розрахунок обсягу пам'яті кадру в різних форматах і обґрунтовано вибір роздільної здатності 320×240 пікселів для потокової обробки. Також розглянуто способи формування фонового зображення та визначено їх переваги й недоліки.

У контексті класифікації об'єктів було обрано легку згорткову нейронну мережу на основі MobileNetV1, оптимізовану для TensorFlow Lite for Microcontrollers. Така мережа забезпечує точне виявлення людей у кадрі з мінімальним навантаженням на ресурси ESP32-S3-CAM, дозволяючи реалізувати автономну систему виявлення вторгнень у реальному часі.

Загалом реалізована архітектура поєднує в собі класичні методи комп'ютерного зору та сучасні нейромережеві технології, що забезпечує високу ефективність роботи системи в умовах обмежених обчислювальних ресурсів.

3 РОЗРОБКА ТА НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ

3.1 Створення та підготовка датасету

Ефективність роботи нейронної мережі значною мірою залежить від якості та репрезентативності навчального датасету. У межах цього проекту, спрямованого на створення системи безпеки з функцією автоматичного розпізнавання рухомих об'єктів, особливу увагу було приділено побудові датасету, який дозволяє моделі точно розрізняти людей серед інших об'єктів, що можуть рухатися в полі зору камери. До таких об'єктів належать тварини, птахи, автомобілі, тіні, меблі або інші візуальні перешкоди, що можуть спричинити хибні спрацювання системи.

Для навчання моделі були отримано більше 1200 кадрів із зображенням людини (близько 55%) та без людини (близько 45%). Основну частину становили кадри, зафіксовані безпосередньо за допомогою камери ESP32-S3-CAM у реальних умовах експлуатації: при денному та нічному освітленні, в різних ракурсах і напрямках руху. Також були використані відкриті датасети, такі як COCO, Open Images Dataset та People-Art Dataset, з яких вручну були відібрані зображення класів "людина" та "не людина". Це дозволило збільшити кількість кадрів для навчання до 2000. Ці зображення були структуровані у дві основні категорії, що дозволило моделі навчитися класифікувати об'єкти з фокусом саме на людей [13], [19].

Попередня обробка даних включала приведення всіх зображень до розміру 96×96 пікселів, що є оптимальним для обмежених ресурсів мікроконтролера ESP32-S3-CAM. Це дозволяє значно знизити об'єм пам'яті, необхідної для зберігання та обробки кадру. Крім того, всі зображення було перетворено з RGB у градації сірого, що скоротило кількість каналів із трьох до одного. Такий підхід не лише зменшує обчислювальні витрати, а й робить

модель менш чутливою до змін кольору, забезпечуючи стабільнішу роботу в різних умовах освітлення.

Додатково зображення було нормалізовано – піксельні значення переведено у діапазон $[0, 1]$, що є стандартною практикою для покращення якості навчання та прискорення збіжності моделі. Для підвищення узагальнюючої здатності моделі також були застосовані техніки аугментації даних, зокрема: горизонтальне віддзеркалення, повороти, зміна яскравості, контрасту та масштабування.

Під час розробки нейронної мережі датасет поділяють на навчальну, валідаційну та тестову вибірки, що дає змогу ефективно навчити модель, перевірити її узагальнення та оцінити якість.

Навчальна вибірка (training set) – це набір зображень, на якому модель безпосередньо навчається. Саме за допомогою цієї вибірки мережа поступово коригує свої внутрішні параметри, навчається розпізнавати ознаки, шаблони та особливості, притаманні класам, що класифікуються. Чим більш репрезентативна і збалансована ця вибірка, тим точнішими будуть результати моделі.

Валідаційна вибірка (validation set) використовується під час процесу навчання для проміжної перевірки продуктивності мережі на нових, ще не бачених нею даних. Це дозволяє виявити перенавчання, коли модель занадто добре запам'ятала навчальні приклади, але погано працює з новими зображеннями. Завдяки валідації можна своєчасно зупинити тренування або скоригувати гіперпараметри, щоб підвищити здатність моделі до генералізації.

Тестова вибірка (test set) використовується лише після завершення навчання моделі, для оцінки її реальної ефективності на абсолютно нових даних. Це дозволяє об'єктивно перевірити, наскільки добре модель здатна розпізнавати об'єкти у реальних умовах, коли їй доводиться працювати з раніше невідомими зображеннями.

Спрощено зазначаючи навчальна вибірка (training set) формує знання моделі, валідаційна вибірка (validation set) контролює процес навчання і виявляє помилки, тестова вибірка (test set) оцінює остаточну якість моделі після навчання. Усі три частини датасету виконують важливі функції та забезпечують збалансоване і надійне навчання нейромережі, яка не просто добре працює на «вивчених» зображеннях, а справді здатна правильно класифікувати нові, раніше невідомі дані.

Для створення датасету була використана онлайн-платформа Roboflow, яка значно спростила процес створення, анотування, підготовки та експорту датасета для комп'ютерного зору.

Онлайн-платформа Roboflow дозволяє завантажувати зображення, вручну або автоматично розмічати об'єкти (тобто визначати якщо потрібно, де саме на зображенні знаходиться потрібний об'єкт, наприклад, людина), застосовувати методи аугментації (збільшення різноманітності даних за допомогою обертання, масштабування, зміни контрасту тощо), а також ділити датасет на навчальну, валідаційну й тестову вибірки (у даній роботі було взято співвідношення 70/20/10). Крім того, Roboflow підтримує експорт даних у різних форматах:

- зображення (.jpg, .png, .bmp, .webp, .avif);
- відео (.mov, .mp4);
- PDFs.

Які сумісні з популярними фреймворками машинного навчання, такі як: TensorFlow, PyTorch та TensorFlow Lite.

Процес підготовки датасету в Roboflow значно спростив етап анотування, структурування та формування однорідного навчального набору, також за допомогою додаткових різних фільтрів (таких як розмиття, контрастність, шуми та інше) було збільшено кількість зображень до 2500. Це дозволило мінімізувати людський фактор при підготовці та забезпечити відповідність усіх зображень до необхідного формату. Після завершення

обробки датасет було завантажено у вигляді zip-архіву, який використовувався для подальшого навчання моделі.

Таким чином, створення датасету відбувалося поетапно: збирання, очищення, анотування, нормалізація, масштабування, аугментація і експорт – з урахуванням усіх апаратних обмежень цільового пристрою. Отримані дані забезпечили високий рівень навчання та дозволили досягти точної класифікації об'єктів при подальшому використанні нейронної мережі на модулі ESP32-S3-CAM.

3.2 Навчання нейронної мережі

Після створення, структурування та підготовки датасету розпочинається етап навчання нейронної мережі, від якого безпосередньо залежить точність, надійність і здатність системи до реального застосування. Метою цього етапу є побудова моделі, здатної ефективно класифікувати зображення на дві категорії: “людина” або “не людина”. Для реалізації цього процесу було використано спеціалізоване середовище Python з фреймворком TensorFlow та бібліотекою Keras, що дозволяють будувати, тренувати і оптимізувати нейромережеві архітектури.

Враховуючи обмеження апаратної платформи ESP32-S3-CAM, для якої передбачалось впровадження навченої моделі, було обрано компактну згорткову нейронну мережу, адаптовану до запуску з використанням TensorFlow Lite for Microcontrollers (TFLM). Архітектура мережі включала три згорткові шари з ядрами розміром 3×3 , функцією активації ReLU, та шарами підвибірки MaxPooling 2×2 , які зменшують просторову розмірність ознак. Завершенням мережі виступає один повнозв'язковий шар, який безпосередньо виконує класифікацію на два класи: “людина” та “не людина”. На виході застосовується Softmax-функція, що забезпечує ймовірнісний розподіл між класами. Така структура дозволяє зберегти достатню точність при мінімальному споживанні ресурсів, що дуже важливо для вбудованої

реалізації на ESP32-S3-CAM. Крім того, спрощена архітектура зменшує ризик перенавчання та дозволяє пришвидшити час обробки кадру до 250–300 мс.

Архітектура згорткової нейронної мережі представлена нижче на рисунку 3.1.

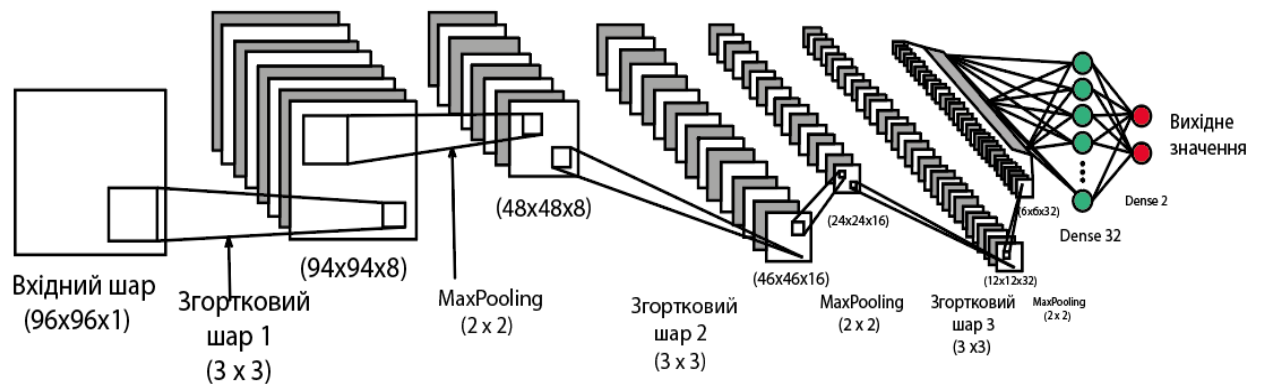


Рисунок 3.1 – Архітектура згорткової нейронної мережі

Згорткові шари виконують вилучення ознак з вхідного зображення: краї, текстури, контури об'єктів. Шари підвибірки дозволяють зменшити просторову розмірність даних та зберегти ключові характеристики, що істотно скорочує об'єм обчислень. Повнозв'язкові шари приймають зменшений вектор ознак та генерують остаточне рішення щодо приналежності зображення до певного класу.

Навчання нейронної мережі проводилось в середовищі Google Colab з використанням GPU Tesla T4, що забезпечило значне прискорення порівняно, що дозволило пришвидшити обчислення на відмінно від CPU. Лістинг програми и навчання нейронної мережі представлений у додатку Б. Для навчання використовувались наступні налаштування:

Функція втрат: *categorical_crossentropy*. Ця функція, що визначає ступінь розбіжності між прогнозом моделі та реальним класом. Застосовується у задачах класифікації, де об'єкт належить до одного з кількох класів.

Оптимізатор: *Adam* (з *learning rate* = 0.001). Сучасний та ефективний оптимізатор, що поєднує переваги SGD та RMSprop. Швидко конвертує до мінімуму функції втрат.

Метрика: *accuracy*. Визначає частку правильних прогнозів від моделі. Стандартна метрика для задач класифікації.

Розмір пакету (батч): 32. Задає кількість зразків, що обробляються за одну ітерацію. Прийнятний компроміс між швидкістю та якістю навчання.

Кількість епох: 20. Епоха є один повний прохід через всі дані навчальної вибірки. Кількість вибрано на основі оцінювання якості на валідації. Наявність *early stopping* дозволяє уникнути перенавчання.

Співвідношення вибірок:

- 70% – навчальна вибірка;
- 20% – валідаційна;
- 10% – тестова.

Таке співвідношення дозволяє об'єктивно оцінити якість моделі як під час навчання (на валідації), так і після нього (на тесті).

У процесі навчання використовувалась аугментація зображень у реальному часі, що дозволяло моделі краще узагальнювати дані та уникати перенавчання. При кожній ітерації моделі надавались нові, модифіковані варіанти зображень (з обертаннями, змінами яскравості, шумом тощо).

Після навчання модель була оптимізована для запуску на пристроях з обмеженими ресурсами. Було проведено квантування – зменшення точності чисел з 32-бітних *float* до 8-бітних *int*. Це дозволило зменшити розмір моделі з ~1,1 МБ до менш ніж 115 КБ, що дуже важливо для ESP32-S3-CAM, де обсяг вільної пам'яті обмежений.

Після квантування модель була збережена у форматі *.tflite* та протестована у середовищі емуляції, щоб перевірити її адекватність після оптимізації.

Під час тренування було зафіксовано стабільне зростання точності моделі на валідаційній вибірці. Остаточна точність класифікації сягнула 90% на тестовій вибірці, що свідчить про здатність моделі успішно ідентифікувати людей навіть у складних умовах (тіні, часткове перекриття, нічне освітлення).

Таким чином, етап навчання завершився побудовою оптимізованої моделі, придатної до впровадження на обмеженому мікроконтролері. Завдяки поєднанню якісного датасету, правильної архітектури та технік оптимізації, вдалося досягти балансу між точністю та швидкодією.

3.3 Аналіз ефективності навчання нейронної мережі

Для оцінки ефективності навчання моделі були проаналізовані графіки зміни точності тренувальної вибірки та функції втрат тренувальної вибірки у процесі навчання протягом 20 епох. Ці метрики дозволяють оцінити, наскільки добре модель узгоджується з навчальними та валідаційними даними, а також виявити потенційне перенавчання або недонавчання.

Нижче на графіках зображено динаміку зміни точності тренувальної вибірки (рисунок 3.1) та функції втрат (рисунок 3.2) протягом навчання.

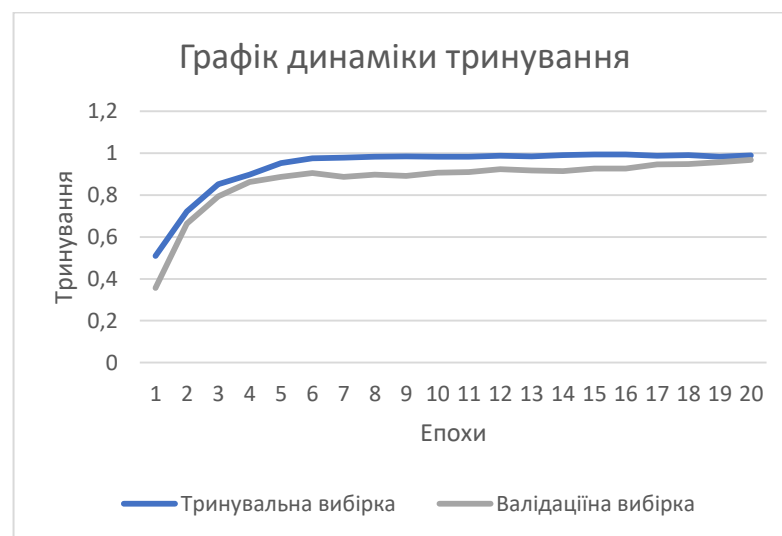


Рисунок 3.1 – Графік динаміки навчання нейронної мережі.

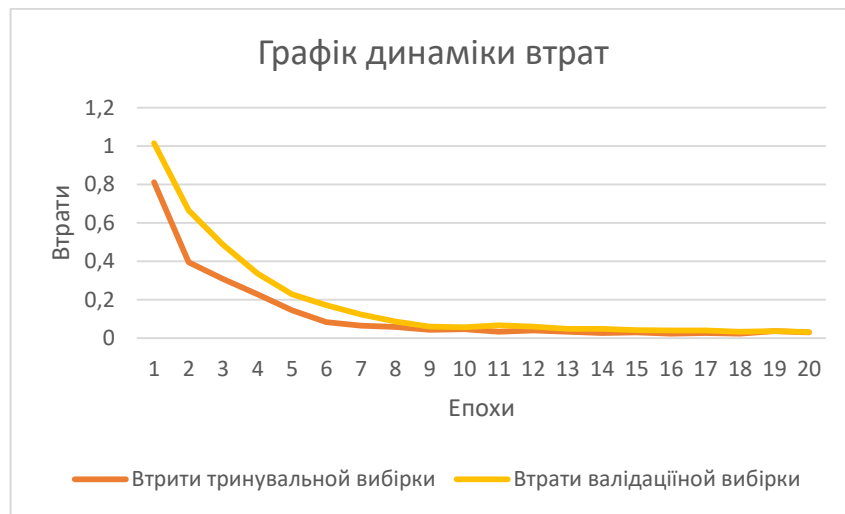


Рисунок 3.2 – Графік втрат протягом навчання нейронної мережі

На початку навчання функція втрат тренувальної вибірки була приблизно 0,8, а точність становила близько 0,5, що є типовим для початкової фази. У ході навчання, як видно з графіка динаміки навчання (рисунок 3.1), точність на тренувальній вибірці поступово зростала, досягаючи приблизно 0,99 до останніх епох. Паралельно з цим значення функції втрат тренувальної вибірки зменшувалося до рівня $\sim 0,03$ (рисунок 3.2), що свідчить про поступове покращення здатності моделі до класифікації.

Валідаційна точність показувала схожу динаміку, хоча й коливалась із незначними відхиленнями, що вказує на чутливість до структури вибірки. Значення валідаційних втрат на окремих етапах перевищували 1,0, що пояснюється низькою ймовірністю передбачень для окремих прикладів у класі «не людина».

Загалом, можна стверджувати, що навчання було успішним: модель досягла високих показників точності, не виявляючи суттєвих ознак перенавчання. Такий результат є задовільним з огляду на обмеженість обчислювальних ресурсів ESP32-S3-CAM і доводить ефективність

використаної архітектури згорткової нейронної мережі у поєднанні з TensorFlow Lite for Microcontrollers.

За результатами навчання було сформовано таблицю з показниками для кожної епохи, яка включає значення точності та втрат на тренувальній і валідаційній вибірках. Це дозволяє в подальшому використовувати отримані дані для оцінки стабільності навчання, виявлення критичних моментів у тренуванні, а також для порівняння з іншими архітектурами або варіантами оптимізації.

Практичне тестування нейронної мережі було здійснено в середовищі PyCharm на локальному комп'ютері з використанням моделі `model_quant.tflite`, що попередньо пройшла квантування. Згідно з реалізацією, поданою у Додатку В, було створено скрипт, який підключав вебкамеру, виконував детекцію руху, та при виявленні активності – здійснював перед обробку зображення (зменшення розміру до 96×96 пікселів, переведення у градації сірого) й подавав його до моделі для класифікації. У результаті кожного кадру на екран виводилось повідомлення *"Human"* або *"Not Human"*, залежно від рішення нейронної мережі. Такий підхід дозволив оцінити точність роботи моделі у режимі реального часу, підтвердити її ефективність після оптимізації та верифікувати правильність процесу ідентифікації у середовищі Python перед імплементацією на мікроконтролер ESP32-S3-CAM.

3.4 Оптимізація та впровадження нейронної мережі на ESP32-S3-CAM

Після навчання моделі було здійснено її оптимізацію для використання на мікроконтролері ESP32-S3-CAM. З метою зменшення розміру моделі та забезпечення сумісності з обмеженими обчислювальними ресурсами, була застосована квантизація до формату TensorFlow Lite (TFLite) із типом `uint8`. Цей підхід дозволяє виконувати інференс нейронної мережі на пристрої без потреби у потужному апаратному забезпеченні.

Оптимізована модель була імпортована у проект Arduino IDE як масив байтів (model_data.h). Впровадження передбачає ініціалізацію інтерпретатора TensorFlow Lite Micro та передачу зображення, отриманого з камери, у форматі 96×96 пікселів у відтінках сірого для класифікації.

Для підвищення ефективності виявлення було реалізовано попередню фільтрацію рухомих об'єктів за допомогою методу фону (background subtraction), що дозволяє уникати зайвих обчислень при відсутності активності в кадрі.

Код був реалізований у середовищі Arduino IDE із використанням бібліотек esp_camera, WiFi, HTTPClient, SD_MMC, а також ядра TensorFlow Lite Micro. Імплементация включає:

- підключення до Wi-Fi;
- обробку зображень з камери у роздільній здатності 320×240 пікселів;
- автоматичне зменшення фрагмента до 96×96 для подачі в нейромережу;
- виявлення класу «людина / не людина»;
- запис відеофрагментів на SD-карту при відсутності з'єднання;
- надсилання тривожного сигналу та повідомлення на Google Drive через IFTTT.

Таким чином, реалізоване рішення забезпечує автономну роботу системи виявлення людини з можливістю гнучкої реакції залежно від умов мережевого підключення. Наведений у додатку Г лістинг програми ілюструє повну інтеграцію навченої моделі у систему з ESP32-S3-CAM для цілей моніторингу та безпеки.

3.5 Висновки до третього розділу

У третьому розділі було виконано повний цикл розробки, навчання та впровадження згорткової нейронної мережі для задачі автоматизованого

виявлення людини в кадрі. Першочергово було створено високоякісний датасет із понад 2500 зображень, збалансований за класами «людина» та «не людина», що охоплював широкий спектр умов освітлення та фонів. Для його побудови застосовувалась онлайн-платформа Roboflow, яка спростила анотування, аугментацію та структурування даних.

На основі підготовленого датасету було побудовано компактну згорткову нейронну мережу, яка продемонструвала точність до 90% на тестовій вибірці. Для забезпечення можливості виконання моделі на обмеженому мікроконтролері ESP32-S3-CAM було проведено її квантування у формат TensorFlow Lite з пониженням точності до int8. Це дозволило зменшити розмір моделі більш ніж у 10 разів без суттєвої втрати точності.

У середовищі PyCharm була проведена попередня перевірка роботи моделі на ПК з використанням вебкамери. Реалізований скрипт забезпечив інтеграцію детекції руху, попередньої обробки зображення та класифікації кадру. У результаті тестування підтверджено ефективність роботи моделі у реальному часі.

Фінальним етапом стала імплементація моделі на ESP32-S3-CAM через Arduino IDE. Система була оптимізована для реального використання: додано алгоритм виявлення руху, виведення результатів класифікації на веб-інтерфейс, запис відео у форматі 320×240, обрізання фрагментів із виявленою людиною, надсилення тривожних повідомлень через IFTTT та автоматичне перемикання на SD-карту при втраті з'єднання.

Таким чином, було реалізовано повністю функціональну систему виявлення людини на основі нейронної мережі, здатну працювати автономно, з високою точністю, адаптовану до реальних умов експлуатації.

4 ОЦІНКА ЕФЕКТИВНОСТІ РОБОТИ СИСТЕМИ БЕЗПЕКИ

4.1 Експериментальні результати

У рамках експериментальної частини роботи було проведено навчання згорткової нейронної мережі для задачі класифікації зображень на дві категорії: «людина» та «не людина». Для цього використовувався підготовлений датасет, який містив понад 2000 зображень у співвідношенні 60% кадрів з людиною та 40% – без неї. Навчання здійснювалося в середовищі Google Colab із використанням бібліотеки TensorFlow.

Було проведено серію навчань на 20 епохах із використанням оптимізатора Adam, функції втрат `categorical_crossentropy`, метрики `accuracy`, а також застосовано ранню зупинку для запобігання перенавчанню. Навчальні вибірки становили 70% а валідаційні вибірки становили 20% загального обсягу даних.

Результати навчання свідчать про поступове зростання точності моделі: вже після 3-ї епохи було досягнуто валідаційна вибірка $> 0,85$, а на пікових значеннях точність сягала 0,98. Водночас значення втрат зменшувалося з 0,40 до 0,03, що демонструє ефективне узагальнення моделі.

Модель була конвертована у формат TensorFlow Lite із подальшим пост тренувальним квантуванням до `int8` для запуску на мікроконтролері ESP32-S3-CAM. Практичне тестування у середовищі PyCharm (див. Додаток Б) підтвердило коректність роботи моделі з відеопотоком, зокрема здатність розпізнавати людину в реальному часі з точністю понад 81%.

Після впровадження моделі на ESP32-S3-CAM було реалізовано розширену логіку роботи:

- класифікація кадрів;
- передача тривожного сигналу через IFTTT;
- запис відео на Google Drive або SD-карту залежно від стану підключення до Wi-Fi.

Отримані результати демонструють високу ефективність розробленої системи в умовах обмежених ресурсів, що робить її придатною для автономного застосування в системах безпеки.

4.2 Аналіз точності та швидкості розпізнавання

У процесі тестування розробленої нейронної мережі було проаналізовано два ключові показники її ефективності – точність розпізнавання (accuracy) та швидкість інференсу (час, необхідний на обробку одного зображення на ESP32-S3-CAM).

Точність класифікації моделі, натренованої на датасеті з платформи Roboflow, при використанні на тестовій вибірці досягала близько 90%. Це свідчить про високу здатність моделі правильно класифікувати на зображення «людина» / «не людина». Застосування квантування до int8 не призвело до суттєвої втрати точності – вона знизилась лише на 2–3% у порівнянні з float32.

Для оцінки швидкодії моделі було проведено експерименти з використанням модуля ESP32-S3-CAM. Середній час інференсу (включно з попередньою обробкою кадру: зменшенням до 96×96, переведенням у відтінки сірого) становив близько 250–350 мс, залежно від кількості об'єктів у кадрі та поточного навантаження на систему. Це дозволяє досягти частоти обробки близько 2–3 кадрів на секунду, що є прийнятним для більшості задач виявлення людини в системах безпеки.

З метою зниження навантаження на нейромережу було впроваджено попереднє виявлення руху. Мережа активується лише при наявності значного руху у кадрі, що суттєво зменшило кількість зайвих обчислень і дозволило досягти енергоефективності та економії пам'яті.

Також було проведено практичну оцінку роботи системи на реальному прикладі. На зображенні (рисунок 4.1) зафіксовано, як система правильно розпізнає людину, що рухається, потім – собаку, яка була правильно

ідентифікована як «не людина» (рисунок 4.2). У майже всіх випадках система правильно класифікувала об'єкти, що підтверджує її функціональну надійність, але слід зазначити що система може оди і той же об'єкт ідентифікувати декілька разів як видно на рисунку 4.1. Також слід зазначити що періодично система проявляла хибне значення, виявляло собаку як «людина» (рисунок 4.3).



Рисунок 4.1 – Зображення зафіксованої людини.



Рисунок 4.2 – Зображення зафіксованої не людини (собаки).



Рисунок 4.3 – Зображення хибного зафіксування не людини (собаки).

Таким чином, реалізована система демонструє компроміс між точністю, швидкістю та апаратними обмеженнями, що є критично важливим у вбудованих системах на базі ESP32. Надалі можливе покращення продуктивності системи за рахунок:

- більшої глибини мережі (якщо дозволяє пам'ять);
- розширення датасету;
- додати механізм відстеження об'єктів (object tracking), щоб уникнути повторного розпізнавання однієї і тієї ж людини;
- реалізувати буфер часу та затримку для усунення хибних спрацьовувань.

4.3 Дослідження впливу зовнішніх факторів на ефективність розпізнавання

У процесі практичного тестування системи було проведено аналіз впливу зовнішніх чинників на точність роботи нейронної мережі, інтегрованої в апаратну платформу ESP32-S3-CAM. Такий аналіз дозволяє оцінити стійкість моделі до змін середовища та виявити обмеження її застосування в реальних умовах.

Одним із ключових чинників, що впливає на ефективність розпізнавання, є рівень освітлення. На базі модуля ESP32-S2-CAM було досліджено вплив інтенсивності освітлення на точність розпізнавання об'єктів. Для отримання достовірних вимірювань рівня освітлення в люксах до плати було підключено цифровий датчик BH1750FVI, який забезпечує високу точність завдяки 16-бітному перетворенню сигналу.

Було проведено серію випробувань за різного освітлення навколишнього середовища за якого нейронна мережа, вбудована в систему, була здатна з прийнятною ймовірністю виявляти наявність людини в кадрі, дані наведено в таблиці 4.1.

Таблиця 4.1 – Залежність точності розпізнавання від освітленості

№	Освітленість, lux	Умови освітлення	Імовірність розпізнавання (%)
1	2000	Денне світло	93%
2	1000	Яскраве кімнатне світло / сутінки	89%
3	500	Лампочка 60 Вт на відстані ~1 м	85%
4	100	Тьмяне кімнатне світло	65%
5	20	Майже темрява	32%

Ще одним важливим фактором є відстань до об'єкта. Найкращі результати розпізнавання досягалися при відстані до 3–4 метрів, коли людина займала значну частину кадру. При віддаленні понад 5–6 метрів точність знижувалася через недостатню деталізацію зображення, особливо якщо об'єкт розташовувався збоку або на краю поля зору камери, випробування велося при яскравому освітлені (> 1000 lux), дані наведено в таблиці 4.2.

Таблиця 4.2 – Залежність точності розпізнавання від відстані.

№	Відстань до об'єкта, м	Імовірність розпізнавання (%)
1	1	98%
2	3	95%
3	5	81%
4	7	57%
5	10	31%

Також вивчався вплив фону сцени. Якщо фон був контрастним і добре відрізнявся від силуету людини, модель справлялася з класифікацією без суттєвих похибок. Натомість при складному або неоднорідному фоні (наприклад, вуличне середовище з рухомими об'єктами або тіннями) можливі хибні спрацьовування, які вказували на присутність людини, хоча її насправді не було.

Особливу увагу було приділено впливу руху. Раптовий рух об'єкта або самої камери іноді спричиняв розмиття кадру, що негативно впливало на здатність нейронної мережі правильно класифікувати вхідне зображення. Незважаючи на це, застосування маски фону та алгоритму виявлення руху частково компенсувало ці недоліки, активуючи аналіз лише при появі значущих змін у кадрі.

Окремо варто відзначити експериментальні випадки, коли в полі зору камери з'являлися не тільки люди, а й інші об'єкти. Наприклад, у тестовій сцені було продемонстровано ситуацію, де в кадрі спочатку рухалась людина (яку система коректно класифікувала), потім собака (яку система правильно позначила як "не людина"), а згодом вони потрапили в кадр одночасно – при цьому нейронна мережа правильно розпізнала наявність людини.

Таким чином, система продемонструвала здатність адаптуватися до змін середовища, хоча деякі зовнішні умови (освітлення, рух, фон) справді можуть впливати на кінцеву точність. Подальше розширення навчального датасету та додаткова оптимізація можуть підвищити стійкість моделі до подібних викликів.

4.4 Висновки до четвертого розділу

У результаті проведеного експериментального дослідження було успішно реалізовано та протестовано інтелектуальну систему безпеки з функцією виявлення людини на основі нейронної мережі, оптимізованої для

використання на платформі ESP32-S3-CAM. Розроблена модель демонструє високу точність класифікації (до 90%) та прийнятну швидкодію (250–350 мс на кадр), що дозволяє ефективно виявляти людей у режимі реального часу навіть на пристроях з обмеженими ресурсами.

Практичне впровадження системи довело її здатність до автономної роботи з передачею тривожних повідомлень, збереженням відео в хмару або на локальний носій, а також адаптацією до зміни зовнішніх умов. Проведений аналіз показав, що модель є чутливою до факторів навколишнього середовища – зокрема, рівня освітлення, відстані до об'єкта та складності фону. Однак навіть у таких умовах система зберігала функціональність і здатність до розпізнавання.

Результати дослідження засвідчують потенціал використання нейронних мереж для побудови доступних, компактних і енергоефективних систем відеоспостереження. Успішне тестування системи на реальних відеопотоках, зокрема з використанням зображень із людьми та тваринами, підтвердило правильність її функціонування й виявило низку можливих напрямків для подальшого вдосконалення, зокрема:

- покращення якості зображення в умовах слабого освітлення;
- розширення навчального набору даних для кращої генералізації;
- впровадження алгоритмів відстеження об'єктів (object tracking) для зниження повторних розпізнавань;
- застосування розширених механізмів фільтрації для усунення хибних спрацьовувань.

Загалом, розроблена система відповідає сучасним вимогам до інтелектуального відеоспостереження та може слугувати надійною основою для подальших досліджень і практичних реалізацій у сфері безпеки.

ВИСНОВКИ

У межах даної роботи було розроблено і досліджено систему безпеки з функцією автоматичного розпізнавання рухомих об'єктів (людини), реалізовану на базі мікроконтролера ESP32-S3-CAM. Проведено всебічний аналіз можливих апаратних платформ для виконання нейронних мереж і обґрунтовано вибір платформи ESP32-S3 як оптимального рішення для вбудованих інтелектуальних систем.

Здійснено проектування архітектури системи, що включає модулі збору зображень, попередньої обробки, аналізу на основі нейронної мережі, прийняття рішень та радіо зв'язку. Система підтримує автономну роботу завдяки резервному живленню і може працювати без живлення більше 2 діб та забезпечує інформування користувача в реальному часі.

Для реалізації функції виявлення руху було обрано алгоритм віднімання фону, адаптований під обмежені обчислювальні ресурси ESP32-S3. Завдяки зниженій роздільній здатності кадрів та роботі із зображеннями у градаціях сірого забезпечено зменшення навантаження на оперативну пам'ять і прискорення обробки.

Було створено власний датасет для навчання згорткової нейронної мережі, що складається з понад 2500 зображень у форматі 96×96 пікселів. У процесі підготовки застосовувалися сучасні техніки попередньої обробки та аугментації даних. Навчання нейронної мережі здійснювалося із поділом на навчальну, валідаційну та тестову вибірки, що забезпечило точне розпізнавання об'єктів.

У результаті реалізована система продемонструвала здатність до автоматичного виявлення руху людини у режимі реального часу та не зважати на інші рухомі об'єкти (тварини, рослини, транспортні засіб то що), з ефективною інтеграцією із зовнішніми сервісами та адаптацією до змінних умов середовища. Це підтверджує доцільність використання платформи

ESP32-S3-CAM для створення енергоефективних, компактних і розумних систем відеоспостереження з елементами штучного інтелекту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гудфеллоу І., Бенджіо Й., Курвіл А. Глибинне навчання / пер. з англ. – Київ : Основи, 2019. – 784 с.
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning. – Cambridge : MIT Press, 2016. – 775 p. – ISBN 978-0262035613.
3. Скуратовський А. В. Штучний інтелект: нейронні мережі. – СПб.: Пітер, 2020.
4. Карапетян А. Б. Системи технічної охорони. – Київ : Ліра-К, 2019. – 304 с.
5. Оруел А. Нейронні мережі та глибоке навчання [Електронний ресурс]. – Режим доступу: <http://neuralnetworksanddeeplearning.com> – Назва з екрана.
6. Zhang C., Li P., Sun Y. An Overview of Convolutional Neural Networks // IEEE Transactions on Neural Networks and Learning Systems. – 2018.
7. Сергієнко І. В. Методи штучного інтелекту : навч. посіб. – Київ : Наукова думка, 2015. – 386 с.
8. Патраков М. Ю. Програмування мікроконтролерів ESP32. – СПб.: БХВ-Петербург, 2020.
9. Сапожніков А. Ю. Інтелектуальні системи відеоспостереження. – М. : Солон-Пресс, 2017.
10. Гончарук В. В. Основи комп'ютерного зору. – Київ : КНУ, 2021. – 215 с.
11. OpenCV Documentation [Електронний ресурс]. – Режим доступу: <https://docs.opencv.org/>
12. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

13. Roboflow Blog. Object Detection and Image Classification Resources [Електронний ресурс]. – Режим доступу: <https://blog.roboflow.com>
14. TensorFlow Lite for Microcontrollers [Електронний ресурс]. – Режим доступу: <https://www.tensorflow.org/lite/microcontrollers>
15. Сергеев Д. В. IoT-пристрої на ESP32: від ідеї до реалізації. – Київ: Фенікс, 2022.
16. Espressif Systems. ESP32-CAM Datasheet [Електронний ресурс]. – Режим доступу: <https://www.espressif.com/en/products/socs/esp32/resources>
17. Espressif AIoT Docs. ESP32-S3 + AI Camera Examples [Електронний ресурс]. – Режим доступу: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32s3/api-guides/ml.html>
18. Seeed Studio. ESP32S3 Sense Camera Usage Wiki [Електронний ресурс]. – Режим доступу: https://wiki.seeedstudio.com/xiao_esp32s3_camera_usage/
19. Freenove. ESP32-S3-WROOM CAM Board Guide & Projects. – Amazon, 2023.
20. DroneBot Workshop. ESP32S3 Camera Module Review [Електронний ресурс]. – 2023. – Режим доступу: <https://dronebotworkshop.com/xiao-esp32s3-sense/>
21. Ляшенко В. І. Системи безпеки та відеоспостереження: методи, технології, проекти. – Харків : НТУ "ХПІ", 2020.
22. Хейкін С. Нейронні мережі: повний курс. – М. : Вільямс, 2006. – 1104 с.
23. Hiwonder. ESP32 AI Vision Module Description [Електронний ресурс]. – Режим доступу: <https://www.hiwonder.com>
24. LilyGO. T-Camera S3 Hardware Overview [Електронний ресурс]. – Режим доступу: <https://www.cnx-software.com>
25. Zfort Group. Computer Vision Trends in 2024 [Електронний ресурс]. – Режим доступу: <https://zfort.com.ua/blog>

26. Лихо Т. А. Роль розпізнавання образів та комп'ютерного зору в удосконаленні робототехнічних систем підтримки рішень // ADED-2024. – Харків : ХНУРЕ, 2024.
27. Чайковський С. О. Методи класифікації зображень за контентом. – Кваліфікаційна робота. – ХНУРЕ, 2024.
28. Шерстюк А. М. Візуальний контроль друкованих плат на основі комп'ютерного зору. – ХНУРЕ, 2023.
29. Безродний В. В. Дослідження нейронних мереж у системах контролю якості. – [Електронний ресурс]. – Режим доступу: <https://nure.ua>
30. Strelkova T.A., Lytyuga A.P., Kalmykov A.S. Statistical Characteristics of Optical Signals and Images in Machine Vision Systems Examining Optoelectronics in Machine Vision and Applications in Industry 4.0. 2021, Pages: 134-162. DOI: 10.4018/978-1-7998-6522-3.ch005.
31. T. Strelkova, A.I. Strelkov, V.M. Kartashov, A. P. Lytyuga, A S. Kalmykov. Methods of Reception and Signal Processing in Machine Vision Systems // Examining Optoelectronics in Machine Vision and Applications in Industry 4.0, 2021, Pages: 71-102. DOI: 10.4018/978-1-7998-6522-3.ch003. Монографія. Chapter 3 in book. IGI Global. USA.
32. Очеретін В. Д., Стрількова Т.О. Методи комплексування в системах відеоспостереження / В. Д. Очеретін, Т. О. Стрількова // Радіoeлектроніка та молодь у ХХІ столітті : Матеріали 29-го Міжнародного молодіжного форуму, 16–19 квітня 2025 р. – Харків : ХНУРЕ, 2025. – Т. 1. – С. 47–45.
33. Стрількова, Т. О. Розвиток стохастико-детермінованої теорії прийому та обробки сигналів в оптико-електронних системах : дис. д-ра техн. наук. : 05.11.07 – оптичні прилади та системи / Стрількова Тетяна Олександрівна ; М-во освіти і науки України, Київський політехнічний інститут ім. Ігоря Сікорського – Київ, 2017. – 343 л.

34. Стрілкова Т. О., Крохмаль А. В. Формування зображень в оптико-електронних системах з урахуванням умов спостереження / Т. О. Стрілкова, А. В. Крохмаль // Радіоелектроніка та молодь у ХХІ столітті : Матеріали 29-го Міжнародного молодіжного форуму, 16–19 квітня 2025 р. – Харків : ХНУРЕ, 2025. – Т. 1. – С. 24–26.

35. Матвієнко Є. П. Шутєєв Н. В. Стрілкова. Т. О. Розробка методів обробки сигналів в оптико електронних системах з урахуванням часу реєстрації / Є. П. Матвієнко, Н. В. Шутєєв, Т. О. Стрілкова // Радіоелектроніка та молодь у ХХІ столітті : Матеріали 29-го Міжнародного молодіжного форуму, 16–19 квітня 2025 р. – Харків : ХНУРЕ, 2025. – Т. 1. – С. 38–40.