

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніка
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Розроблення системи комп'ютерного зору для розпізнавання об'єктів на базі
нейронної мережі для квадрокоптера

(тема)

Виконав:

студент 4 курсу, групи АКТАКІТ-21- 2

Олексій КОРНІЄНКО

(власне ім'я, прізвище)

Спеціальність 151 Автоматизація та
комп'ютерноінтегровані технології

(код і повна назва спеціальності)

Тип програми Освітньо-професійна

Освітня програма Автоматизація та
комп'ютерно-інтегровані технології

(повна назва освітньої програми)

Керівник проф. Владислав ЄВССЄВ

(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. Кафедри КІТАР

(підпис)

Ігор НЕВЛЮДОВ

(власне ім'я, прізвище)

2025 р.

Я, Корнієнко Олексій Віталійович, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

“5” червня 2025 р.

Олексій КОРНІЄНКО

A handwritten signature in black ink, consisting of stylized, overlapping letters that appear to be 'AK' followed by a horizontal line extending to the right.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКТ _____

Кафедра _____ КІТАР _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва)

Тип програми _____ Освітньо-професійна _____

Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. Кафедри КІТАР _____
(підпис)

«30» квітня 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Корнієнко Олексію Віталійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення системи комп'ютерного зору для розпізнавання об'єктів на базі нейронної мережі для квадрокоптера

Затверджена наказом університету від 19.05.2025р. № 390 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 16.06.2025р.

3. Вихідні дані до роботи:

3.1 Розпізнавання об'єктів з використанням нейронної мережі

3.2 Камера до 5 Мп

3.3 Відстань розпізнавання до 50 метрів

3.4 Мінімальне освітлення для розпізнавання 15 Люкс

4. Перелік питань, що потрібно опрацювати в роботі:

4.1 Вступ;

4.2 Аналіз сучасних квадрокоптерів;

4.3 Аналіз сучасних систем комп'ютерного зору;

4.4 Розробка системи комп'ютерного зору для квадрокоптера;

4.5 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Демонстраційний матеріал, представлений у форматі презентації PowerPoint (*.ppt). – 10 с. Формату А4.

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 | Терміни виконання етапів роботи | Примітка |
|---|---|---------------------------------|----------|
| 1 | Аналіз сучасних квадрокоптерів | 30.04.2025 | Виконано |
| 2 | Розробка системи комп'ютерного зору для квадрокоптера | 10.05.2025 | Виконано |
| 3 | Експериментальна частина | 04.06.2025 | Виконано |
| 4 | Оформлення пояснювальної записки | 06.06.2025 | Виконано |
| 6 | Подання роботи на рецензію | 08.06.2025 | Виконано |
| 7 | Подання роботи на підпис зав. кафедри | 16.06.2025 | Виконано |
| | | | |
| | | | |
| | | | |

Дата видачі завдання 30.04.2025 р.

Здобувач

(підпис)

Олексій Корнієнко

(власне ім'я, прізвище)

Керівник роботи _____

(підпис)

професор Владислав Євсєєв

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 65 с., 1 табл., 45 рис., 4 дод., 28 джерела.

КОМП'ЮТЕРНИЙ ЗІР, КВАДРАКОПТЕР, НЕЙРОНА МЕРЕЖА, ШТУЧНИЙ ІНТЕЛЕКТ, АЛГОРИТМ

Мета роботи – покращення ефективності ідифікації об'єктів за рахунок навчання нейронної мережі на базі MobileNetV2 для квадрокоптера.

Об'єкт розробки – процес обробки зображення з системи комп'ютерного зору.

Предмет розробки – методи, апаратне та програмне забезпечення для розпізнавання об'єктів на базі нейронної мережі.

В даній кваліфікаційній роботі проведено аналіз учасник квадрокоптерів, їх конструкція та засоби керування. Проведено аналіз апаратних модулів. На базі проведеного аналізу – обрано відповідні апаратні модулі. Розроблено структурну схему. Розроблена робоча програма для пошуку та ідифікації об'єктів с використанням нейронної мережі. Розроблена програма для отримання та передачі відео для використання ESP32-CAM.

Проведені експерименти с точністю розпізнавання об'єктів з урахуванням відстані та навколишнім освітленням.

ABSTRACT

Explanatory note: 65 p., 1 table, 45 figures, 4 appendices, 33 sources.

COMPUTER VISION, QUADROCOPTER, NEURAL NETWORK, ARTIFICIAL INTELLIGENCE, ALGORITHM

The purpose of the work is to improve the efficiency of object identification by training a neural network based on MobileNetV2 for a quadcopter.

The object of the development is the process of image processing from a computer vision system.

The subject of the development is methods, hardware and software for object recognition based on a neural network.

In this qualification work, an analysis of the quadcopter participants, their design and control means was carried out. An analysis of the hardware modules was carried out. Based on the analysis, the appropriate hardware modules were selected. A structural diagram was developed. A working program was developed for searching and identifying objects using a neural network. A program for receiving and transmitting video has been developed for use with ESP32-CAM.

Experiments have been conducted with the accuracy of object recognition taking into account distance and ambient lighting.

ЗМІСТ

| | |
|--|----|
| Перелік скорочень..... | 9 |
| Вступ..... | 10 |
| 1 Аналіз сучасних квадрокоптерів..... | 12 |
| 1.1 Аналіз технічних характеристик сучасних квадрокоптерів..... | 12 |
| 1.2 Аналіз систем керування квадрокоптерів..... | 16 |
| 1.3 Аналіз систем комп'ютерного зору для розпізнавання об'єктів..... | 18 |
| 2 Розробка системи комп'ютерного зору..... | 23 |
| 2.1 Вибір середовища розробки..... | 23 |
| 2.2 Розробка загального алгоритму..... | 24 |
| 2.3 Програмна реалізація системи комп'ютерного зору..... | 25 |
| 3 Розробка макету системи комп'ютерного зору..... | 31 |
| 3.1 Розробка структурної схеми..... | 31 |
| 3.2 Вибір апаратних модулів..... | 32 |
| 3.3 Розробка прошивки роботи ESP-32CAM..... | 35 |
| 3.4 Збірка макету квадрокоптера..... | 38 |
| 4 Експериментальна частина..... | 41 |
| 4.1 Експеримент с дальністю розпізнавання об'єктів..... | 41 |
| 4.2 Експеримент с освітленням..... | 43 |
| 4.3 Аналіз результатів..... | 46 |
| 4.4 Розрахунки мікроклімату та освітлення приміщення..... | 48 |
| Висновки..... | 51 |
| Перелік джерел посилань..... | 52 |
| Додаток А Апробація..... | 55 |
| Додаток Б Код програми для отримання та обробки зображення..... | 59 |
| Додаток В Код програми для ESP32-CAM..... | 61 |
| Додаток Г Демонстраційний матеріал..... | 65 |

ПЕРЕЛІК СКОРОЧЕНЬ

МК – мікроконтролер;

НДР – науково-дослідна робота;

ПЗ – програмне забезпечення;

FPV – First Person View;

GPS (англ. Global Positioning System) – Глобальна позиція системи;

IDE (англ. Integrated Development Environment) – інтегроване середовище розробки;

URL (англ. Uniform Resource Locator) – веб-адрес.

ВСТУП

На сьогоднішній день квадрокоптери це один з найпопулярніших автономних роботів, які здатні виконувати багато задач. Їхня популярність обґрунтована дешевим та швидким виробництвом легкістю керування, великими можливостями для виконання від простих до складних задач, починаючи від простої іграшки, закінчуючи військовою зброєю, яка дуже сильно полегшує життя нашим захисникам.

Квадрокоптери складаються з контролера польоту, акумулятором, антеною та 3 або більше серводвигунами с пропелерами. Також можна додавати безліч засобів для покращення функціоналу дрона, як приклад камера, різні датчики, зброю тощо. Ці компоненти дозволяють виконувати різні важкі задачі в рятувальних, військових або розвідувальних цілях.

Системи комп'ютерного зору допомагають операторам квадрокоптера орієнтуватися де знаходиться квадрокоптер та приймати рішення. Використання нейронної мережі може допомогти оператору в керуванні та знаходження необхідних об'єктів, беруче на себе задачі по пошуку.

Таким чином метою даною кваліфікаційної роботи є покращення ефективності ідентифікації об'єктів за рахунок навчання нейронної мережі на базі MobileNetV2 для квадрокоптера.

Об'єкт розробки – процес обробки зображення з системи комп'ютерного зору.

Предмет розробки – методи, апаратне та програмне забезпечення для розпізнавання об'єктів на базі нероної мережі.

Для досягнення поставленої мети потрібно вирішити такі завдання:

- аналіз сучасних квадрокоптерів, їх конструкція та засоби керування;
- аналіз систем комп'ютерного зору;
- розробити структурну схему системи.
- розробити загальний алгоритм роботи системи комп'ютерного зору;
- розробити програму роботи системи комп'ютерного зору;

- розробити структурну схему системи;
- вибрати апаратні частини для розробки макету;
- запрограмувати апаратні модулі;
- зібрати робочий прототип макету.

Кваліфікаційна робота виконана згідно ДСТУ 3008:2015 [1], а також з методичними вказівками з підготовки й оформлення кваліфікаційної роботи здобувачами першого (бакалаврського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» [2] та навчальним посібником [3][4] та положення про академічну доброчесність [5].

1 АНАЛІЗ СУЧАСНИХ КВАДРАКОПТЕРІВ

1.1 Аналіз технічних характеристик сучасних квадрокоптерів

Квадрокоптери, як окрема категорія безпілотних літальних апаратів з 4 двигунами, посідають важливе місце в сучасному технологічному розвитку. Популярність і функціональність квадрокоптерів зумовлена комплексом технічних і програмних інновацій, що дозволяють забезпечити стабільний політ, автономність навігації та інтеграцію з системами збору й обробки даних.

З технічного погляду, квадрокоптер є вертикально взлітаючим літальним апаратом із симетрично розміщеними двигунами, які забезпечують підйомну силу, маневреність та стабілізацію в повітрі. Контроль над положенням в просторі здійснюється шляхом варіювання швидкості обертання окремих пропелерів, що дозволяє точно регулювати рух по всіх осях. У поєднанні з сучасними системами інерціального орієнтування, глобального позиціонування (GPS), барометричними датчиками та системами комп'ютерного зору, вони здатні функціонувати у складних умовах із високою точністю.

Квадрокоптери відіграють дедалі більшу роль як інструмент оперативного збору геопросторової інформації. Їх використовують для картографування місцевості, аналізу стану рослинного покриву, моніторингу змін кліматичних умов, а також у археологічних розвідках. Завдяки високій маневреності та здатності до точкового зависання, квадрокоптери ефективні для досліджень об'єктів із важкодоступною або небезпечною для людини локалізацією.

Крім того, квадрокоптери є об'єктом активного інженерного і програмного вдосконалення. Розробники зосереджують увагу на підвищенні енергоефективності, зменшенні ваги елементної бази, інтеграції штучного інтелекту для розпізнавання об'єктів і прийняття рішень у режимі реального часу.

Таким чином, квадрокоптери представляють собою високотехнологічні

платформи, здатні до виконання багатофункціональних завдань із високим ступенем автономії.

Як приклад можна розглядати дешевий квадрокоптер моделі E99 (рис. 1.1), цей дешевий дрон с камерою, чудово підходить для ознайомлення новачків с особливостями керування.



Рисунок 1.1 – Квадрокоптер моделі E99 [6]

Він має такі технічні характеристики:

- складний розмір: 13*8*5,5 см;
- розгорнутий розмір: 20*16,5*5,5 см;
- частота 2,4 г;
- ємність батареї: 1800 ма/ч;
- відстань дистанційного керування: 150 метрів;
- робоча батарея: 15 хвилин (фактичні 10 хв);
- відстань передачі зображень: 15-30 м;
- сигнал передачі зображення: передача wi-fi.

Зараз у військових дуже популярні Fpv-дрони, або ще відомий як дрон-камікадзе (рис. 1.2), вони дешеві, швидкі і здатні нести вибуховий заряд для ураження противника. Їхні характеристики залежать від конкретної моделі але загальні технічні характеристики такі:

- розгорнутий розмір: 12,7 - 25,4 см;

- відстань дистанційного керування: 3-10 кілометрів;
- швидкість: 100-150 км/г.



Рисунок 1.2 – Fpv-дрон [7]

Один з прикладів Fpv-дронів є “Мамонт” (рис. 1.3), цей дрон здатен нести до 4 кг заряду на відстань до 12 км. Такі характеристики забезпечуються більшими розмірами та потужностями.



Рисунок 1.3 – Fpv-дрон Мамонт [8]

Окрім квадрокоптерів, у сучасному безпілотному літальному середовищі активно використовуються гексакоптери – дрони, що оснащені шістьма

двигунами. Вони вирізняються підвищеними технічними характеристиками, що відкриває перед ними значно ширший спектр можливостей у порівнянні з їхніми чотиригвинтовими аналогами. Завдяки збільшеній кількості двигунів гексакоптери демонструють вищу стабільність у польоті, особливо в умовах сильного вітру або при виконанні складних маневрів. Це робить їх особливо привабливими для застосування в ситуаціях, де потрібна підвищена точність та контроль, зокрема під час аерофотозйомки високої чіткості або при транспортуванні чутливого обладнання.

Гексакоптери характеризуються не лише поліпшеними льотними якостями, але й істотною перевагою у вантажопідйомності. Водночас ця перевага тягне за собою певні компроміси: для підтримки стабільної роботи шести двигунів потрібна значно більша енергія, що ставить перед розробниками нові виклики в контексті розробки більш ємних і надійних джерел живлення.

Потреба в потужніших акумуляторах, своєю чергою, обумовлює збільшення загальної маси апарата, що, в комплексі з розмірами, впливає на конструкційні особливості корпусу. Останній має бути достатньо міцним, щоби витримувати додаткові навантаження, зберігаючи при цьому аеродинамічні властивості та мінімізуючи втрати енергії в польоті. Таким чином, гексакоптери становлять більш складну в технічному плані платформу, яка потребує ретельного інженерного підходу на етапах проектування та експлуатації.

Як приклад візьмемо гексакоптер DJI S900 (рис. 1.4), він має такі технічні характеристики:

- загальний злітна вага: від 4.7 кг до 8.2 кг;
- загальний споряджений вага без корисного навантаження: 3,3 кг;
- акумулятори lipo 6s, ємністю від 10000 маг до 15000 маг, мінімум 15с;
- загальна потужність близько 3000 вт;
- споживання струму в режимі зависання 1000 вт (при вазі 6,8 кг) ;
- час польоту: 18 хвилин, при акумуляторі 12ач і вазі 6,8 кг.



Рисунок 1.4 – Гексакоптер [9]

1.2 Аналіз систем керування квадрокоптера

Система керування квадрокоптерами є однією з найважливіших складових, що забезпечує ефективну експлуатацію безпілотного літального апарата в повітряному просторі. Залежно від конструкційного рівня, технічного оснащення та призначення дрона, можуть застосовуватись різні підходи до організації дистанційного управління, кожен з яких має свої технічні особливості, переваги та обмеження. На сучасному етапі розвитку технологій управління квадрокоптерами відзначається високий ступінь інтеграції як апаратних, так і програмних засобів, що дозволяє значно розширити функціонал та зменшити вимоги до спеціальної підготовки користувача.

Традиційно управління дроном здійснюється за допомогою спеціалізованих пультів дистанційного керування, які забезпечують надійний канал зв'язку між оператором та квадрокоптером. Такі пульти часто обладнані аналоговими або цифровими джойстиками, які дозволяють вручну змінювати висоту, напрямок руху, швидкість та інші параметри польоту в реальному часі (рис. 1.5). Надійність подібного методу зумовлена використанням спеціальних радіочастот, що забезпечують стійкий сигнал на значній відстані, мінімізуючи затримки та вплив перешкод. Завдяки цьому пілот має можливість здійснювати точне й оперативне керування, що особливо важливо у складних метеоумовах або при виконанні завдань, пов'язаних із високою відповідальністю.



Рисунок 1.5 – Пульти керування квадрокоптера [10]

Утім, із розвитком мобільних технологій та інтерфейсів бездротового зв'язку, зокрема Wi-Fi та Bluetooth, усе більшої популярності набуває керування квадрокоптерами за допомогою мобільних пристроїв - смартфонів і планшетів (рис. 1.6). Такий підхід базується на застосуванні спеціалізованих мобільних додатків, які дозволяють користувачеві керувати дроном через сенсорний екран, використовуючи віртуальні органи управління або ж вбудовані функції, як-от акселерометр для зміни положення апарата шляхом нахилу самого пристрою. У поєднанні з можливістю перегляду зображення в режимі реального часу, мобільне управління значно розширює інструментарій користувача, створюючи додаткові можливості для планування маршрутів, використання автоматизованих режимів польоту та обробки отриманих даних.



Рисунок 1.6 – RC Fly, програма для керування дронів завдяки смартфону

Проте слід враховувати, що стабільність та дальність зв'язку в цьому випадку, як правило, є обмеженішими, ніж при використанні традиційних пультів, що може бути критичним чинником у професійному застосуванні.

1.3 Аналіз систем комп'ютерного зору для розпізнавання об'єктів

Для ефективнішого використання комп'ютерного зору потрібно проаналізувати існуючі системи та задачі які вони можуть виконувати. Основні задачі які виконуються:

- розпізнавання об'єктів;
- трекінг одного об'єкта;
- розпізнавання жестів, рис лиця, міміки тощо;
- орієнтація в місцевості;
- пошук та фіксація фото при завданих умовах.

Сучасні технології комп'ютерного зору пропонують широкий спектр рішень - від простих алгоритмів до складних систем зі штучним інтелектом. У більшості випадків вибір алгоритму залежить від конкретної мети. Наприклад, найпростіший алгоритм комп'ютерного зору – *sot.py*. Цей простий алгоритм використовується для захоплення та відстежування об'єкта. Його можна застосовувати на підприємствах для автоматизації процесів контролю, трекінгу квадрокоптерів, системах відеоспостереження, для відстежування руху об'єктів у реальному часі тощо. Простота реалізації таких алгоритмів дозволяє швидко інтегрувати їх у практичні проекти, зменшуючи витрати на розробку та впровадження.

Більш складні алгоритми можуть розпізнавати різні об'єкти або їх частини. Як приклад для розпізнавання обличчя використовуючи геометричні фігури. У цьому підході обличчя умовно розбивається на набір ключових елементів, які формують певні геометричні структури. Алгоритм аналізує взаємне розташування цих елементів – очей, носа, рота, ліній щелепи тощо – і створює унікальну "мапу" обличчя. Потім ця мапа порівнюється з раніше збереженими

зразками в базі даних, що дозволяє точно ідентифікувати особу.

Подібна технологія має широке застосування у сфері безпеки та захисту даних. Вона може бути використана, наприклад, для контролю доступу до пристроїв або приміщень, де важлива ідентифікація особи. Також обличчя може виступати в ролі додаткового або навіть основного ключа для автентифікації, забезпечуючи зручність користування і водночас високий рівень захисту особистої інформації.

Складні системи комп'ютерного зору також використовують нейронну мережу для збільшення ефективності та виконання більш складних задач, таких як розпізнавання різних об'єктів. На відміну від простих алгоритмів, які здатні лише відстежувати рух або визначати базові форми, нейронні мережі надають змогу здійснювати значно глибший аналіз візуальної інформації. Вони здатні навчатися на великій кількості даних, розпізнавати тонкі відмінності між об'єктами, адаптуватися до змін навколишнього середовища та враховувати контекст, у якому перебуває об'єкт. Проте, розробка та впровадження нейронних мереж потребують значних обчислювальних ресурсів і часу на навчання, але результат часто виправдовує витрати.

Ці технології відкривають нові можливості для автоматизації, аналітики та взаємодії систем із навколишнім середовищем, значно розширюючи межі того, що можна досягти за допомогою комп'ютерного зору.

Як приклад модель LSTM (Long Short-Term Memory) використовується для прогнозування траєкторії руху руки людини в робочій зоні колаборативного робота-маніпулятора завдяки своїй здатності враховувати часові залежності в послідовних даних. Основний принцип роботи LSTM полягає в тому, що вона здатна запам'ятовувати важливу інформацію з попередніх кроків і використовувати її для прийняття рішень на наступних етапах. Це досягається завдяки спеціальній архітектурі, яка включає в себе комірки пам'яті (memory cells), що дозволяють зберігати або відкидати інформацію через механізми "вхідних" і "забуваючих" шлюзів (gates). Коли LSTM отримує послідовність координат руху руки, вона аналізує їх, виявляючи закономірності та залежності між попередніми і поточними положеннями. Завдяки цьому модель може

передбачити, куди рухатиметься рука в наступний момент часу. В контексті колаборативного робота це дозволяє роботу динамічно адаптувати свої дії, уникаючи зіткнень і забезпечуючи безпеку оператора (рис. 1.7).

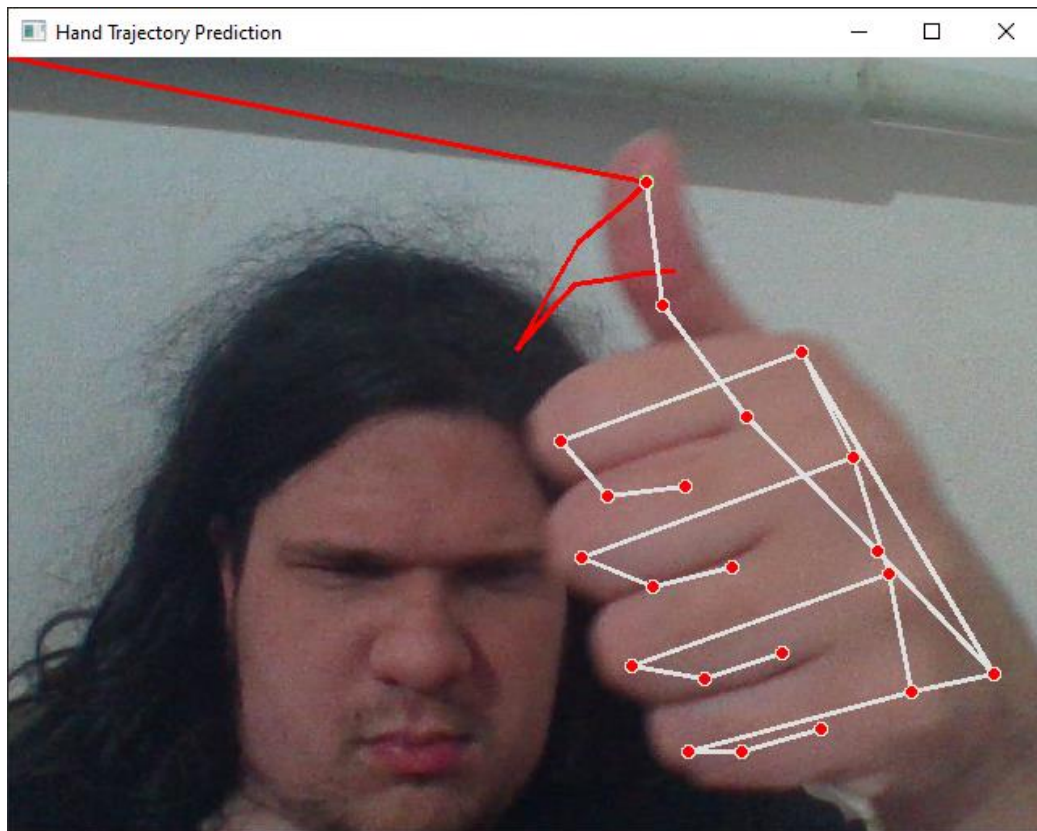


Рисунок 1.7 – Приклад роботи LSTM

Як приклад для ідентифікації різних об'єктів модель MobileNetV2 [12]. Це одна з ефективних і водночас легких моделей глибокого навчання, яка була спеціально розроблена для роботи на пристроях із обмеженими ресурсами. Завдяки оптимізованій структурі, MobileNetV2 демонструє високу швидкість обробки зображень при збереженні достатнього рівня точності, що робить її ідеальним варіантом для застосування в мобільних додатках, вбудованих системах та пристроях Інтернету речей.

Модель здатна в реальному часі розпізнавати кілька об'єктів одночасно, що значно розширює сферу її можливого використання. Вона не лише дозволяє виявляти та класифікувати об'єкти, а й ефективно працює в умовах змінного освітлення, різних кутів огляду та динамічних сцен (рис. 1.8).

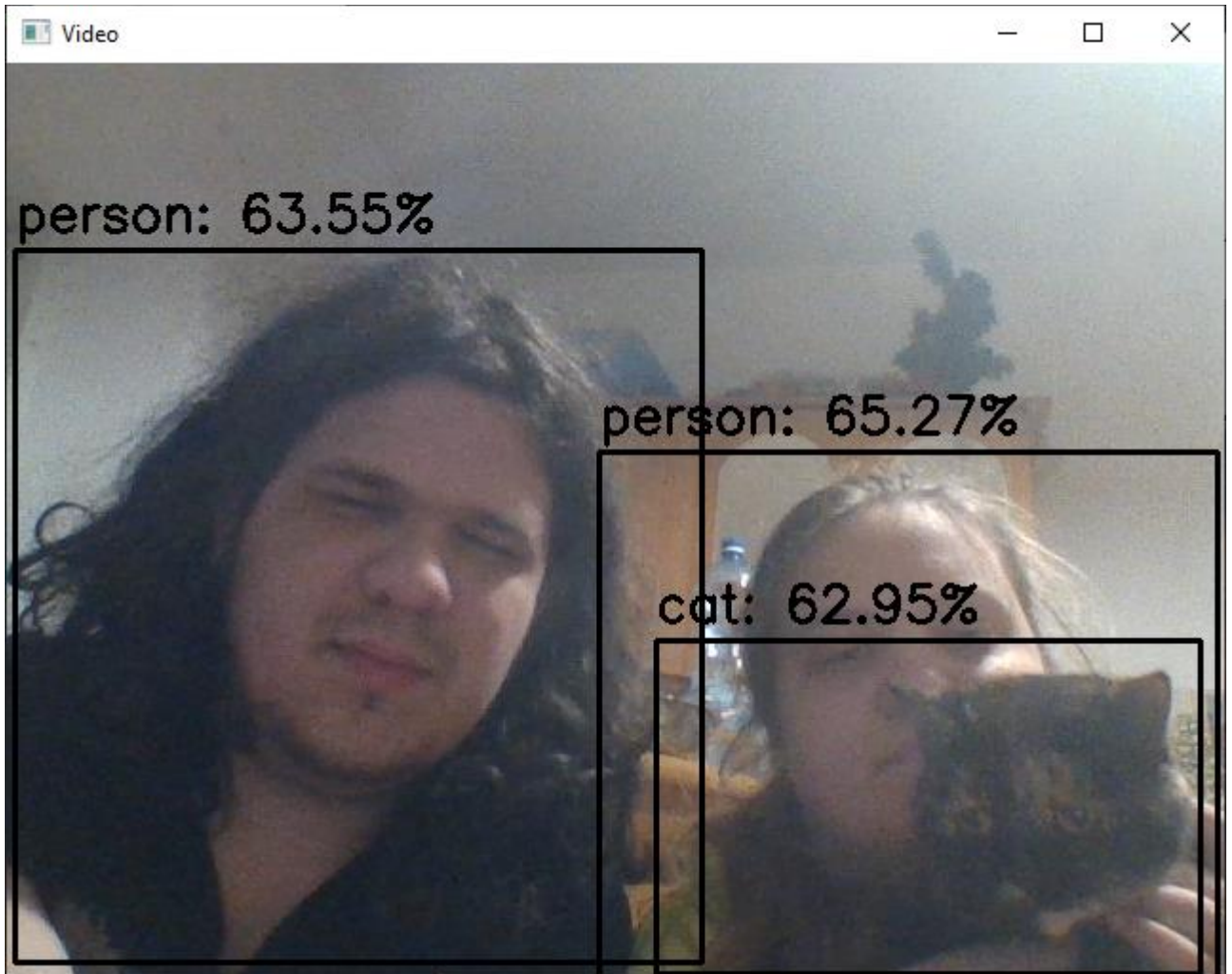


Рисунок 1.8 – Приклад роботи MobileNetV2

MobileNetV3 [13], як наступна еволюційна версія в серії MobileNet, представляє собою вдосконалене поєднання ефективності та точності. Вона була спеціально оптимізована для мобільних і вбудованих пристроїв із урахуванням сучасних потреб у глибокому навчанні. Завдяки поділу на дві окремі конфігурації, Small та Large, модель дозволяє гнучко обирати між максимальною швидкістю обробки або покращеною точністю, що робить її універсальною для широкого спектра застосунків, зокрема на пристроях із різною обчислювальною потужністю. MobileNetV3 демонструє вищу ефективність у порівнянні зі своїми попередниками, особливо в задачах розпізнавання об'єктів, зберігаючи при цьому невелику вагу моделі. Однак ця архітектура є більш складною у використанні, а її налаштування потребує глибших знань та потужніших обчислювальних ресурсів, що може бути перешкодою для застосування в умовах обмеженого середовища.

SqueezeNet [14] натомість представляє собою вкрай компактну згорткову нейронну мережу, яка була розроблена з метою зменшення кількості параметрів без суттєвих втрат у точності. Завдяки унікальній архітектурі, що включає «вогняні блоки» (fire modules), модель досягає дуже малої ваги - менше 5 мегабайт, що робить її однією з найкращих у класі надлегких нейромереж. Це дозволяє легко інтегрувати її у пристрої з мінімальними обчислювальними ресурсами, зокрема мікроконтролери, портативні сенсори або автономні дрони. Попри це, SqueezeNet демонструє обмежену точність у порівнянні з новішими моделями, такими як MobileNetV2, і менш ефективна при роботі зі складними візуальними сценами. Тим не менш, її надзвичайна компактність і простота залишаються ключовими перевагами у багатьох практичних сценаріях, де розмір і швидкість моделі мають вирішальне значення.

2 РОЗРОБКА СИСТЕМИ КОМП'ЮТЕРНОГО ЗОРУ

2.1 Вибір середовища розробки

Python [15] – це мова програмування загального призначення, яка вже багато років посідає провідні позиції в ІТ-індустрії завдяки своїй гнучкості, простоті синтаксису та широкому спектру застосування. З огляду на стрімкий розвиток технологій штучного інтелекту та обробки великих даних, Python утвердився як один із ключових інструментів, що дозволяє ефективно реалізовувати складні алгоритмічні завдання, не втрачаючи при цьому зручності у розробці та читабельності коду.

Однією з найбільш вагомих переваг цієї мови є розгалужена екосистема бібліотек та фреймворків, які охоплюють практично всі сфери сучасного програмування – від обробки природної мови до робототехніки. У контексті комп'ютерного зору Python відкриває широкі можливості завдяки таким потужним інструментам, як OpenCV, TensorFlow, Keras, PyTorch та інші. Вони забезпечують не лише зручний доступ до низькорівневих функцій, необхідних для аналізу зображень, а й готові моделі глибокого навчання, які значно скорочують час розробки та підвищують ефективність реалізованих рішень.

PyCharm є однією з провідних інтегрованих серед розробки (IDE) для мови програмування Python, створеною компанією JetBrains. У порівнянні з іншими середовищами, такими як Visual Studio Code [16], або JupyterLab [17], PyCharm вирізняється передусім глибокою інтеграцією інструментів для повномасштабної розробки програмного забезпечення. На відміну від більшості текстових редакторів з розширеннями, PyCharm реалізує комплексний підхід до організації робочого процесу: від управління середовищами виконання та інтеграції з системами контролю версій до засобів рефакторингу коду, глибокого аналізу помилок та підтримки тестування.

Суттєвим чинником переваги PyCharm є його здатність забезпечувати високий рівень автодоповнення, статичного аналізу та навігації кодом, що

базуються на внутрішніх механізмах побудови абстрактного синтаксичного дерева. Це надає розробнику не лише гнучкість, а й суттєво знижує когнітивне навантаження під час масштабних або багатокомпонентних проєктів.

2.2 Розробка загального алгоритму

Для початку розробки програмної частини був розроблен алгоритм роботи (рис. 2.1).

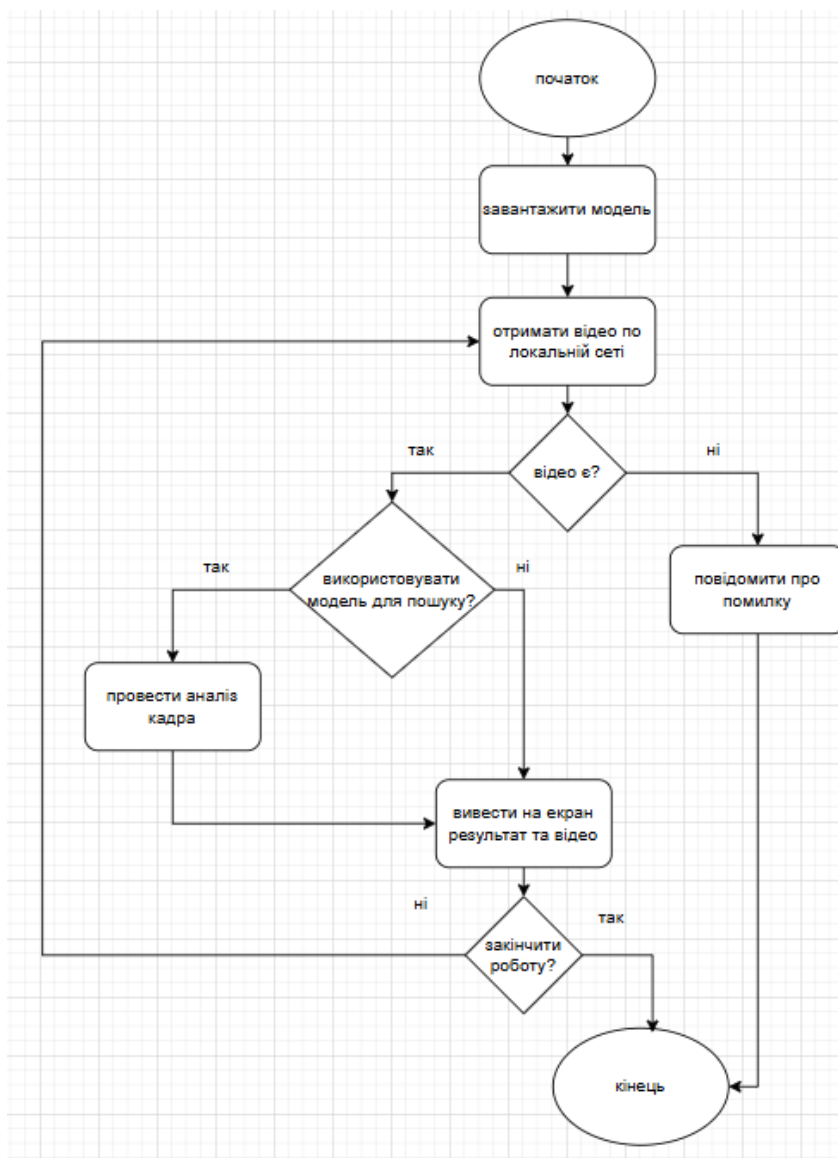


Рисунок 2.1 – Алгоритм роботи програми

Де на початку програма завантажує модель нейронної мережі MobileNetv2. Після завантаження моделі, отримує відео по локальній мережі від

камери, якщо відео є та оператор дал команду аналізувати, кадр передається моделі для аналізу. Після аналізу виводить на екран результат. Якщо відео не надходить, або оператор закінчив роботу, програма закінчує роботу.

2.3 Програмна реалізація системи комп'ютерного зору

Для початку роботи потрібно завантажити бібліотеки (рис. 2.2). Серед яких є:

- cv2 – це одна з найпотужніших бібліотек для комп'ютерного зору та обробки зображень. Вона дозволяє виконувати операції на зображеннях і відео;
- numpy - це основна бібліотека для чисельних обчислень у Python. Вона забезпечує ефективну роботу з багатовимірними масивами та матрицями, а також широкий набір математичних функцій;
- tensorflow - це гнучка та масштабована платформа для машинного навчання та глибокого навчання. Вона дозволяє створювати, навчати, оцінювати та використовувати нейронні мережі;
- urllib.request використовується для роботи з HTTP-запитами, зокрема для завантаження файлів або отримання даних із URL. Це базовий інструмент для інтеграції з інтернет-ресурсами.

```
import cv2
import numpy as np
import tensorflow as tf
import urllib.request
```

Рисунок 2.2 – Завантажені бібліотеки

На рисунку 2.3 зображене завантаження моделі та масив назв об'єктів які можуть бути знайдені.

```
# Завантаження моделі MobileNetV2 для розпізнавання об'єктів
model = tf.saved_model.load(r"C:\Users\User\Desktop\дiплом\мoдeль нейронкi\ssd-mobilenet-v2\1")
# Список класів об'єктів
class_names = [
    '__background__', 'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus',
    'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'N/A', 'stop sign',
    'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow',
    'elephant', 'bear', 'zebra', 'giraffe', 'N/A', 'backpack', 'umbrella', 'N/A', 'N/A',
    'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball',
    'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard', 'tennis racket',
    'bottle', 'N/A', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl',
    'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza',
    'donut', 'cake', 'chair', 'couch', 'potted plant', 'bed', 'N/A', 'dining table',
    'N/A', 'N/A', 'toilet', 'N/A', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone',
    'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'N/A', 'book',
    'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush'
]
```

Рисунок 2.3 – Модель та назви об'єктів

Програмна реалізація функції для розпізнавання об'єктів на зображенні (рис. 2.4).

```
# Функція для розпізнавання об'єктів на зображенні
def detect_objects(image):
    input_tensor = tf.convert_to_tensor(image, dtype=tf.uint8)
    input_tensor = input_tensor[tf.newaxis, ...]
    detections = model.signatures['serving_default'](input_tensor)
    return detections
```

Рисунок 2.4 – Функція для розпізнавання об'єктів на зображенні

Для коректної роботи потрібні наступні параметри (рис. 2.5):

- url – посилання для отримання відео;
- detection_start – параметр який відповідає, чи буде проводитись пошук об'єктів у реальному часі, чи призупинити пошук.

```
url = 'http://192.168.43.117/cam-hi.jpg'
detection_start = 0
```

Рисунок 2.5 – URL звідки програма буде отримувати відео

Початок головного циклу програми та створення назви та розміри вікна показані на рисунку 2.6.

```
while True:  
    cv2.namedWindow("live transmission", cv2.WINDOW_AUTOSIZE)
```

Рисунок 2.6 – Назва та розміри вікна

На рисунку 2.7 показана реалізація отримання кадра від камери.

```
img_resp = urllib.request.urlopen(url)  
imgnp = np.array(bytearray(img_resp.read()), dtype=np.uint8)  
im = cv2.imdecode(imgnp, -1)
```

Рисунок 2.7 – Отримання кадра від камери

При натисканні на пробел змінюється параметр `detection_start` (рис.2.8).

```
if key == 32:  
    if detection_start == 0:  
        detection_start = 1  
    else:  
        detection_start = 0
```

Рисунок 2.8 – Зміна параметра `detection_start`

Якщо цей параметр дорівнює 1, то починається пошук об'єктів на кадрі та виведення їх на екран (рис. 2.9).

```

if detection_start == 1:
    detections = detect_objects(im)
    # Обробка результатів розпізнавання
    num_detections = int(detections['num_detections'][0])
    detection_classes = detections['detection_classes'][0].numpy().astype(np.int64)
    detection_boxes = detections['detection_boxes'][0].numpy()
    detection_scores = detections['detection_scores'][0].numpy()
    # Виведення результатів на кадр
    for i in range(num_detections):
        if detection_scores[i] > 0.5:
            box = detection_boxes[i]
            y1, x1, y2, x2 = box
            y1, x1, y2, x2 = int(y1 * im.shape[0]), int(x1 * im.shape[1]), int(y2 * im.shape[0]), int(x2 * im.shape[1])
            # Отримати назву класу
            class_id = detection_classes[i]
            class_name = class_names[class_id] if class_id < len(class_names) else 'Unknown'
            # Виведення результатів на кадр
            cv2.rectangle(im, (x1, y1), (x2, y2), (0, 0, 0), 2)
            label = f"{class_name}: {detection_scores[i]*100:.2f}%"
            cv2.putText(im, label, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 0, 0), 2)
cv2.imshow('live transmission', im)

```

Рисунок 2.9 – Пошук об’єктів на кадрі

В цій частині коду модель отримує кадр та аналізує її, повертаючи ім’я об’єкта, зона розташування та вірогідність правильності (рис. 2.10). Ця частина кода виконується тільки після отримання команди від оператора, перемикаючи між режимом відео потік і аналіз та виведення результатів на відео потік.

```

# Обробка результатів розпізнавання
num_detections = int(detections['num_detections'][0])
detection_classes = detections['detection_classes'][0].numpy().astype(np.int64)
detection_boxes = detections['detection_boxes'][0].numpy()
detection_scores = detections['detection_scores'][0].numpy()

```

Рисунок 2.10 – Отримання результатів розпізнавання

Виведення на екран всіх результатів аналізу об’єктів на кадрі якщо вірогідність знайденого об’єкта більше 50% наведен на рисунку 2.11.

```

for i in range(num_detections):
    if detection_scores[i] > 0.5:
        box = detection_boxes[i]
        y1, x1, y2, x2 = box
        y1, x1, y2, x2 = int(y1 * im.shape[0]), int(x1 * im.shape[1]), int(y2 * im.shape[0]), int(x2 * im.shape[1])
        # Отримати назву класу
        class_id = detection_classes[i]
        class_name = class_names[class_id] if class_id < len(class_names) else 'Unknown'
        # Виведення результатів на кадр
        cv2.rectangle(im, (x1, y1), (x2, y2), (0, 0, 0), 2)
        label = f"{class_name}: {detection_scores[i]*100:.2f}%"
        cv2.putText(im, label, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 0, 0), 2)

```

Рисунок 2.11 – Виведення результатів на кадр

Закінчення роботи програми виконується або примусовим вимкненням, або натисканням на клавішу esc (рис. 2.12).

```

if cv2.waitKey(1) & 0xFF == ord(27):
    break
cap.release()
cv2.destroyAllWindows()

```

Рисунок 2.12 – Кінець роботи програми

Розроблена програма забезпечує оператору можливість безпосереднього перегляду відеопотоку з камери квадрокоптера в режимі реального часу, що дозволяє ефективно спостерігати за навколишнім середовищем та оперативно приймати рішення у процесі керування польотом. Ключовою особливістю є інтеграція функціоналу комп'ютерного зору та штучного інтелекту для автоматичного виявлення об'єктів на відео потоці. Активація режиму виявлення об'єктів здійснюється інтуїтивно просто – за допомогою натискання клавіші пробілу, що дозволяє оператору без затримок переходити від звичайного спостереження до аналітичного режиму, у якому програма виконує ідентифікацію об'єктів, таких як люди або інші цільові об'єкти.

Після активації режиму аналізу, алгоритм штучного інтелекту обробляє поточний відео потік, здійснюючи пошук об'єктів на кожному кадрі. Всі виявлені об'єкти, незалежно від їх кількості, відображаються на екрані у реальному часі, що забезпечує максимально повне та точне представлення

інформації про поточну ситуацію. Для зручності сприйняття кожен об'єкт візуалізується у вигляді чітких рамок, що дозволяє оператору швидко орієнтуватися у просторі, не перевантажуючи візуальну складову інтерфейсу та зберігаючи високий рівень наочності.

Така архітектура програми поєднує зручність використання із потужними можливостями автоматизованого аналізу даних, що значно підвищує ефективність управління квадрокоптером у складних умовах, забезпечуючи оператору додатковий інструмент для прийняття рішень в режимі реального часу.

3 РОЗРОБКА МАКЕТУ СИСТЕМИ КОМП'ЮТЕРНОГО ЗОРУ

3.1 Розробка структурної схеми

На цьому етапі було розроблена структурна схема управління та обробки відео квадрокоптера з використанням нейронною мережею (рис. 3.1).

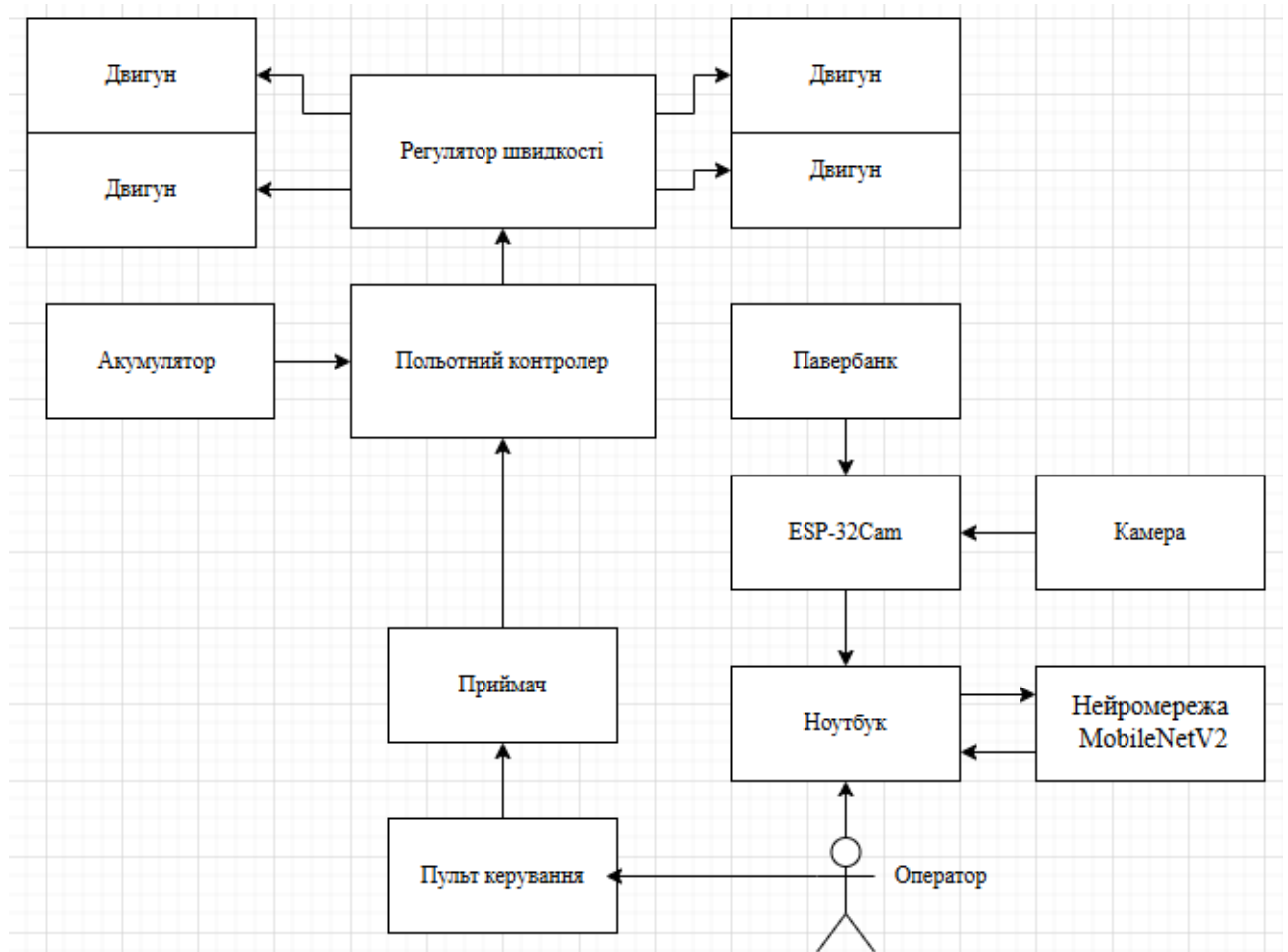


Рисунок 3.1 – Структурна схема роботи макету

Дана структурна схема включає в себе перелік наступних компонентів:

1. Пульт керування – забезпечує дистанційний зв'язок з квадрокоптером та дозволяє оператору в реальному часі керувати всіма його рухами та функціями;
2. Приймач – приймає сигнали з пульта та передає їх до польотного контролера для подальшої обробки;
3. Польотний контролер – центр квадрокоптера, обробляє сигнали від сенсорів і команди оператора, координуючи роботу всіх систем квадрокоптера;

4. Регулятор швидкості – відповідає за точне регулювання обертів двигунів для стабілізації та зміни траєкторії польоту;
5. Двигуни – генерують підйомну силу та забезпечують рух дрона в заданому напрямку;
6. Акумулятор – є основним джерелом живлення для всіх компонентів квадрокоптера під час автономного польоту;
7. Камера – використовується для трансляції зображення з повітря в режимі реального часу;
8. Передатчик – отримує відео від камери та передає її оператору на ноутбук;
9. Ноутбук – пристрій для отримання відео з камери квадрокоптера та використовується для подальшого аналізу;
10. Неймережа MobileNetV2 – здійснює обробку відео, виявляє та розпізнає об'єкти в кадрі в реальному часі.

3.2 Вибір апаратних модулів

Для роботи можна використати різні камери які будуть підключатись до контролера, або бути самостійними від нього, для порівняння (таблиця 3.1) були обрані наступні пристрої:

- FPV камера Caddx Ant Lite 4:3 (рис. 3.2) [18];
- камера FPV нано RunCam Nano 2 2.1мм (рис. 3.3) [19];
- плата ESP32-CAM з камерою OV2640 (рис. 3.4) [20].



Рисунок 3.2 – FPV камера Caddx Ant Lite 4:3



Рисунок 3.3 – Камера FPV нано RunCam Nano 2 2.1мм



Рисунок 3.4 – Плата ESP32-CAM з камерою OV2640 [21]

Таблиця 3.1 – Порівняльні характеристики камер

| Пристрій | FPV камера Caddx Ant Lite 4:3 | Камера FPV нано RunCam Nano 2 2.1мм | Плата ESP32-CAM з камерою OV2640 |
|---------------------------------------|----------------------------------|---|---|
| Роздільна здатність | 1200TVL (1280 x 960) | 800TVL (720 P) | UXGA 1600x1200 (15 fps) / SVGA 800x600 (30 fps) |
| Потребує додаткового обладнання | + | + | - |
| Розміри | 14×14 мм | 14x14x16 мм | 40,5 x 27 x 4,5 мм |
| Ціна | 655 грн | 840 грн | 269 грн |
| Вага | 1,7 г | 3,2 г | 10 г |
| Напруга живлення | 5 v | 3-5 v | 5 v |

Був обраний ESP32-CAM, він самостійний, дешевий, передає відео по Wi-Fi, а також він має відносно низьку вагу та розміри, які дозволять використовувати його для квадрокоптера.

Для подальшої розробки прототипу буде використан старий та дешевий квадрокоптер A6 [16] без камери (рис. 3.5). Технічні характеристики:

- польотний час – 20 хв;
- відстань дистанційного керування – 300 метрів;
- наявність барометра;
- частота роботи – 2.4 GHz.



Рисунок 3.5 – Квадрокоптер А6 [22]

До цього квадрокоптера буде додана ESP32-CAM та павербанк для його живлення, підключення до програми буде завдяки wi-fi.

3.3 Розробка прошивки роботи ESP32-CAM

Для подальшої роботи потрібно прошити ES332-CAM, він прошивається завдяки Arduino IDE [23]. Для правильної роботи потрібно завантажити бібліотеки (рис. 3.6).



Рисунок 3.6 – Бібліотека для прошивки ESP32-cam

Налаштування прошивки наведені на рисунку 3.7, вони забезпечать правильність прошивки та її швидкість.

| | |
|--|---|
| Core Debug Level: "None" | > |
| Erase All Flash Before Sketch Upload: "Disabled" | > |
| Flash Frequency: "80MHz" | > |
| Flash Mode: "QIO" | > |
| Partition Scheme: "Huge APP (3MB No OTA/1MB SPIFFS)" | > |
| Upload Speed: "921600" | > |

Рисунок 3.7 – Налаштування прошивки

Для початку роботи с кодом потрібно навести бібліотеки для правильної роботи (рис. 3.8).

```
#include <WebServer.h>
#include <WiFi.h>
#include <esp32cam.h>
```

Рисунок 3.8 – Бібліотеки для ESP32-CAM

На рисунку 3.8 прийняті наступні значення:

- WebServer.h – бібліотека для роботи веб сервера, на яку будуть відправлятися відео та звідки можна буде їх отримати для подальшої роботи іншими пристроями;
- Wifi.h – бібліотека для підключення до мережі вай-фай;
- esp32cam.h – бібліотека для роботи с ESP32-CAM.

Потрібно створити константи логіна та пароля від вай-фай мережі, щоб міг підключитись до неї (рис. 3.9).

```
const char* WIFI_SSID = "www";
const char* WIFI_PASS = "www777www";
```

Рисунок 3.9 – Логін та пароль від wi-fi

Створино 3 режима передачі відео (рис. 3.10).

```
static auto loRes = esp32cam::Resolution::find(320, 240);  
static auto midRes = esp32cam::Resolution::find(350, 530);  
static auto hiRes = esp32cam::Resolution::find(800, 600);
```

Рисунок 3.10 – 3 Режима роботи

Вони відповідають за роздільну здатність відео яке буде передаватись, надалі буде використано midRes з роздільною здатністю 350/530. Функція setup має в собі параметри пінів для ініціалізації камери (рис. 3.11).

```
void setup(){  
  Serial.begin(115200);  
  Serial.println();  
  {  
    using namespace esp32cam;  
    Config cfg;  
    cfg.setPins(pins::AiThinker);  
    cfg.setResolution(hiRes);  
    cfg.setBufferCount(2);  
    cfg.setJpeg(80);  
  }  
}
```

Рисунок 3.11 – Конфігурація пінів

При вдалому підключенні до Wi-fi мережі надсилає у консоль ArduinoIDE посилання на сайти звідки можна буде взяти фото або відеопотік для подальшої роботи (рис. 3.12).

```
WiFi.persistent(false);
WiFi.mode(WIFI_STA);
WiFi.begin(WIFI_SSID, WIFI_PASS);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
}
Serial.print("http://");
Serial.println(WiFi.localIP());
Serial.println(" /cam-lo.jpg");
Serial.println(" /cam-hi.jpg");
Serial.println(" /cam-mid.jpg");

server.on("/cam-lo.jpg", handleJpgLo);
server.on("/cam-hi.jpg", handleJpgHi);
server.on("/cam-mid.jpg", handleJpgMid);

server.begin();
}
```

Рисунок 3.12 – Посилання для подальшої роботи

3.4 Збірка макету квадрокоптера

Результат роботи та працюючий прототип наведені на рисунку 3.13.



Рисунок 3.13 – Зібраний прототип

Успішне тестування системи комп'ютерного зору для розпізнавання об'єктів на базі нейронної мережі для квадрокоптера наведено на рисунку 3.14.

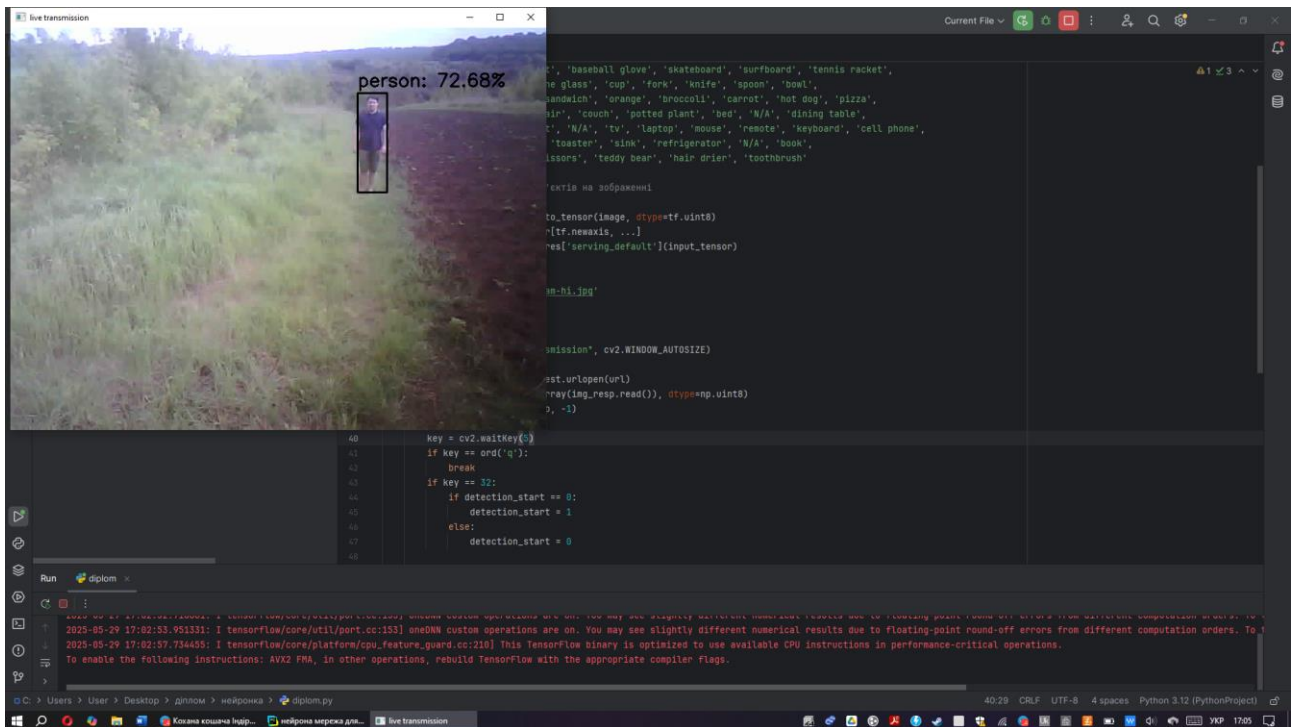


Рисунок 3.14 – Тестування системи комп'ютерного зору

Розроблений прототип відкриває широкі можливості для проведення експериментальних досліджень, спрямованих на оцінку працездатності, ефективності та стійкості системи в умовах, наближених до реального середовища експлуатації.

У рамках запланованих експериментів передбачається перевірка точності розпізнавання об'єктів на різних відстанях, що дозволить оцінити роботу системи у ситуаціях, коли об'єкт знаходиться на межі видимості або віддаляється від камери спостереження.

Крім того, окрему увагу буде приділено дослідженню впливу освітлення на ефективність розпізнавання. Перевірка точності роботи системи при недостатньому або нерівномірному освітленні дозволить оцінити її адаптивність до змін умов навколишнього середовища та здатність підтримувати стабільну якість розпізнавання об'єктів у складних ситуаціях.

Отримані під час експериментів результати дозволять комплексно оцінити

працездатність прототипу та зробити висновки щодо його придатності для виконання завдань у реальних умовах експлуатації.

Загалом проведення таких експериментальних досліджень є важливим етапом у процесі створення надійної та ефективної системи комп'ютерного зору, яка здатна працювати в широкому спектрі умов, забезпечуючи високу точність і стабільність роботи, необхідні для вирішення практичних задач у різних сферах застосування.

4 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

4.1 Експеримент с дальністю розпізнавання об'єктів

Перший експеримент визначить з якою дальністю система комп'ютерного зору зможе побачити об'єкт. Під час експерименту було перевірено ефективність розпізнавання на відстані 5 метрів, 10 метрів, 20 метрів і 50 метрів. Результат роботи програми наведен на рисунку 4.1, де нейрона мережа успішно знайшла об'єкт на відстані 5 метрів.

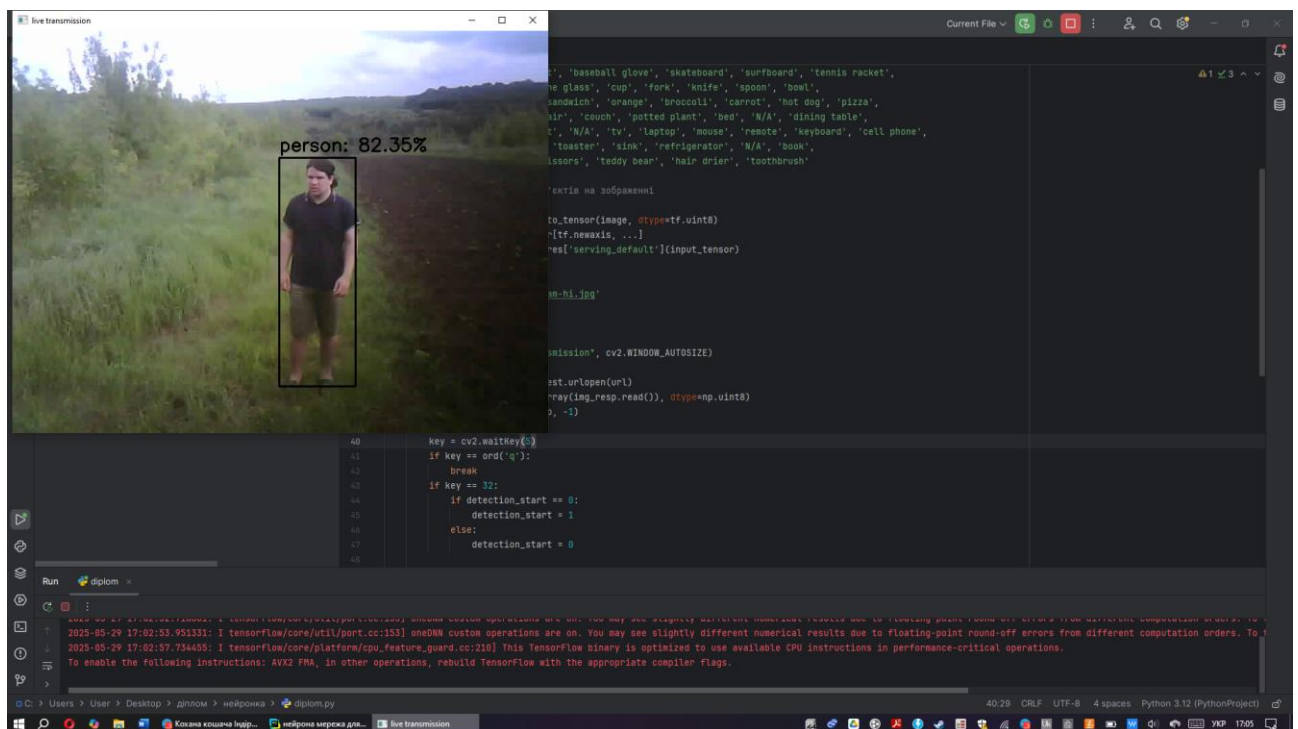


Рисунок 4.1 – Відстань 5 метрів

Наступним етапом буде збільшення відстані до 10 метрів (рис. 4.2).

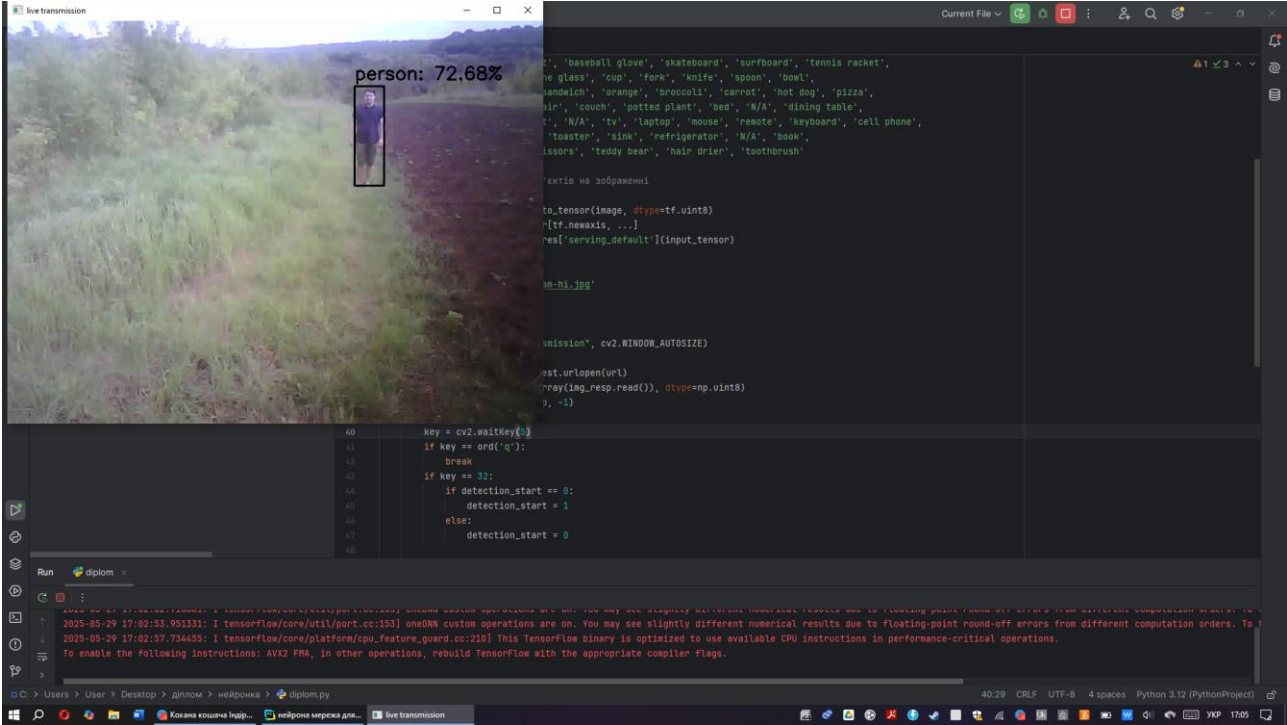


Рисунок 4.2 – Відстань 10 метрів

На рисунку 4.3 відстань була збільшена до 20 метрів.

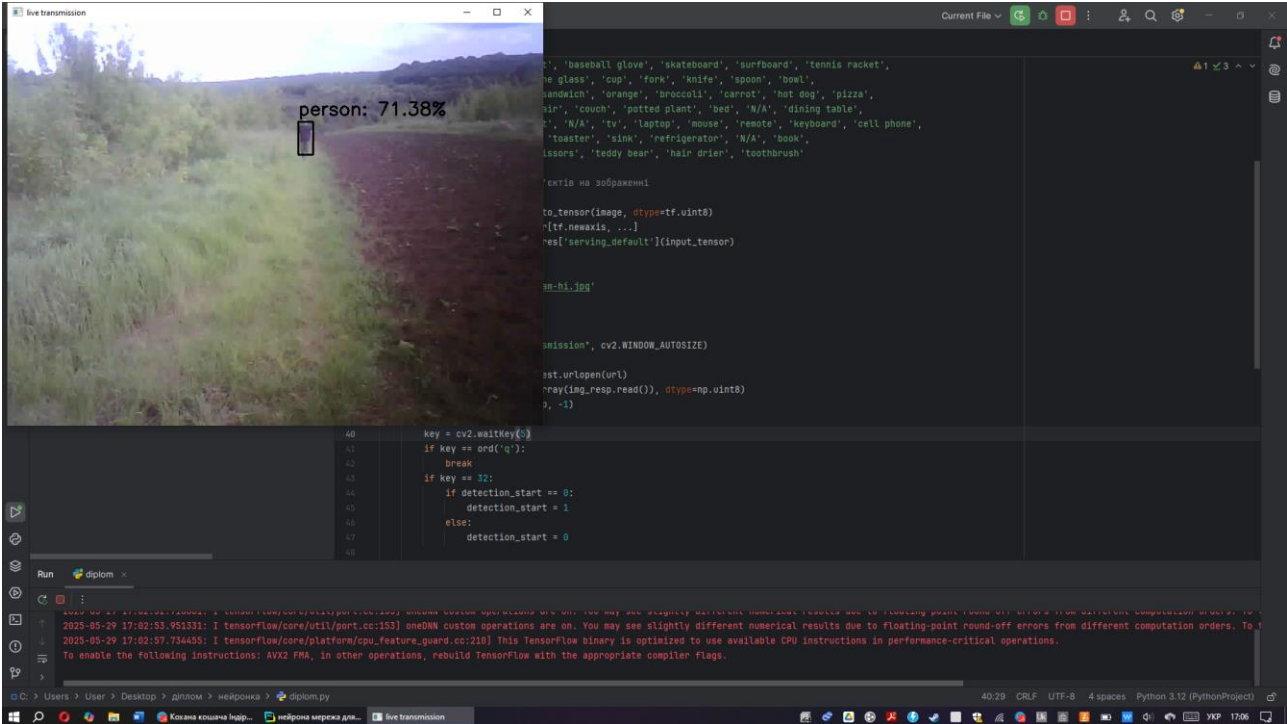


Рисунок 4.3 – Відстань 20 метрів

На відстані 50 метрів (рис. 4.4) нейрона мережа вже не дуже впевнено знаходить, але все також результат позитивний.

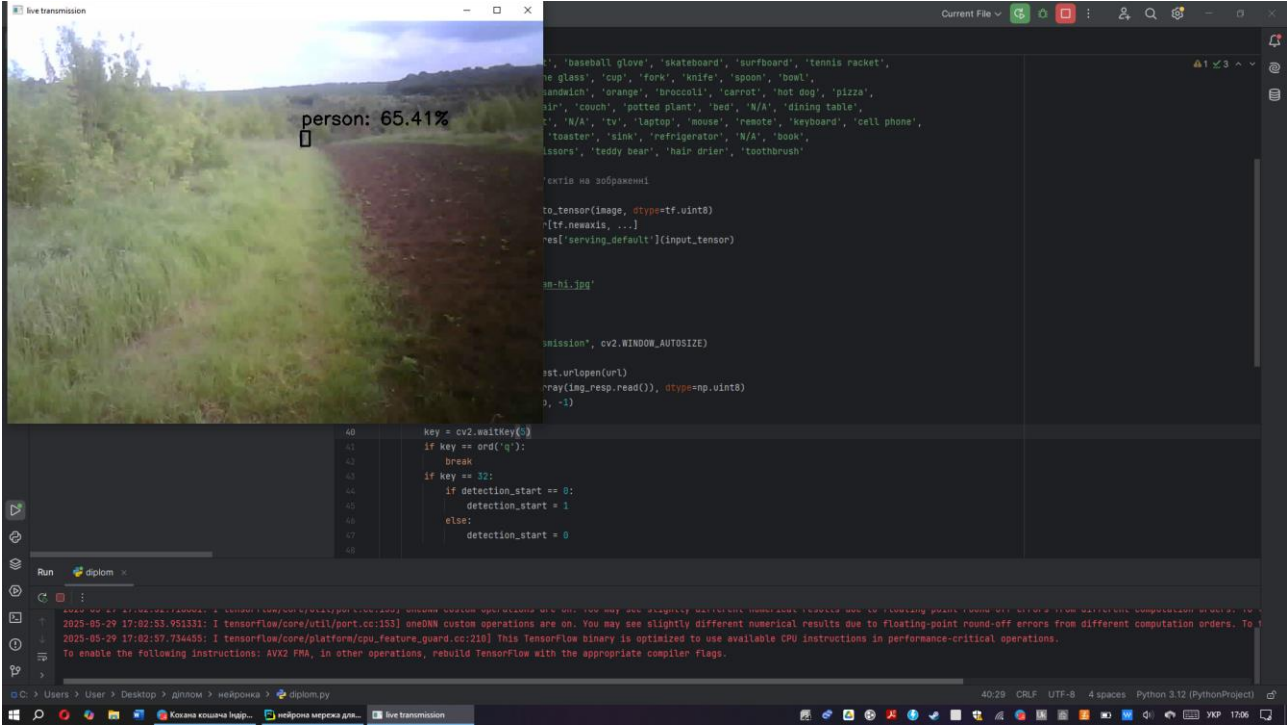


Рисунок 4.4 – Відстань 50 метрів

За результатами експерименту програма успішно знайшла та ідентифікувала об’єкт на відстані 50 метрів с непоганою точністю.

4.2 Експеримент с освітленням

Під час наступного експерименту була перевірена ефективність розпізнавання при надмірному освітленні та при недостатньому освітленні. Під час першого тесту, люксометр показав 257 люкс (рис. 4.5).



Рисунок 4.5 – Заміри люксометра при надмірному освітленні

Результат роботи програми при надмірному освітленні (рис. 4.6). Результат очікувано позитивний.

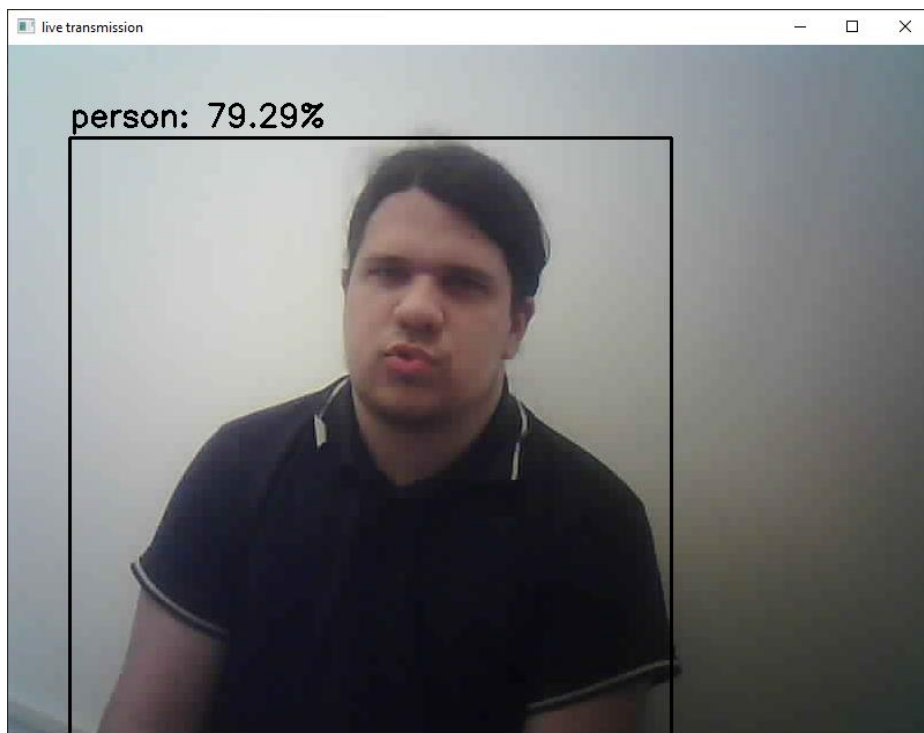


Рисунок 4.6 – Результат програми при 257 люкс

Під час другого тесту з недостатнім освітленням люксометр показав 15 Люкс (рис. 4.7).



Рисунок 4.7 – Заміри люксометра при недостатньому освітленні

Результат роботи при недостатньому освітленні наведено на рисунку 4.8. Важливо примітити, як освітлення впливає на точність розпізнавання, при одній і тій самій відстані, програма успішно розпізнала мене, але с меншою точністю.



Рисунок 4.8 – Результат роботи при 15 люкс

4.3 Аналіз результатів

Аналізуючи результати проведених експериментальних досліджень, стає можливим виявлення характерних закономірностей, що описують вплив відстані до об'єкта на ефективність функціонування системи комп'ютерного зору, побудованої на основі нейронних мереж. Для наочного представлення отриманих даних доцільно скористатися графічною інтерпретацією, де залежність точності розпізнавання об'єктів від відстані відображена у вигляді графіка (рис. 4.9).

Аналіз отриманого графіка дозволяє зробити висновок про те, що навіть за умов значного видалення об'єкта від системи розпізнавання, зокрема на відстані до 50 метрів, рівень точності залишається на досить високому рівні. Це особливо важливо, враховуючи динамічний характер експериментальних умов, де об'єкти

перебувають у русі, що зазвичай ускладнює задачу розпізнавання та призводить до додаткових навантажень на алгоритм обробки даних.

Таким чином, система демонструє стійкість до зміни просторових параметрів сцени та здатність до ефективного функціонування навіть за умов, що наближені до реальних сценаріїв застосування.

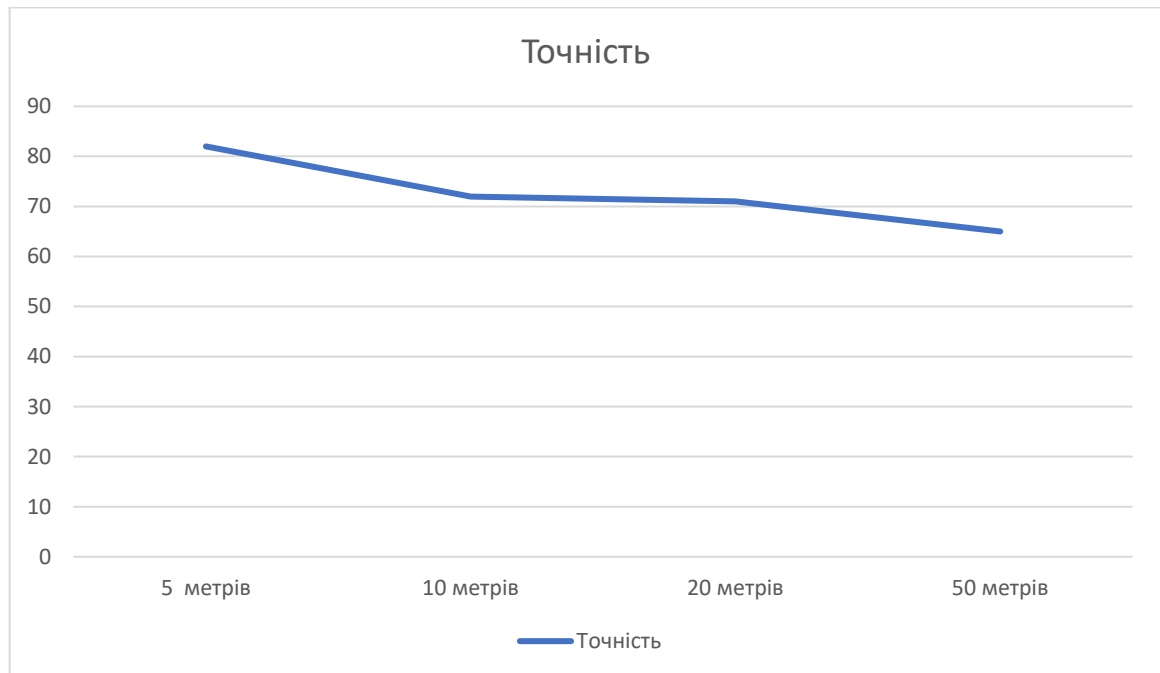


Рисунок 4.9 – Графік точності розпізнавання в залежності від відстані

Окрему увагу в ході експериментів було приділено впливу параметрів освітлення на ефективність розпізнавання (рис. 4.10). Варто зазначити, що зміни освітленості призводили до помітного зниження точності роботи нейронної мережі. Зокрема, при збереженні відстані між камерою та об'єктом на постійному рівні, зменшення рівня освітлення в середовищі спричинило зниження точності розпізнавання приблизно на 5%. Цей результат свідчить про важливу роль освітленості як одного з ключових факторів, що впливають на стабільність та надійність роботи системи комп'ютерного зору.

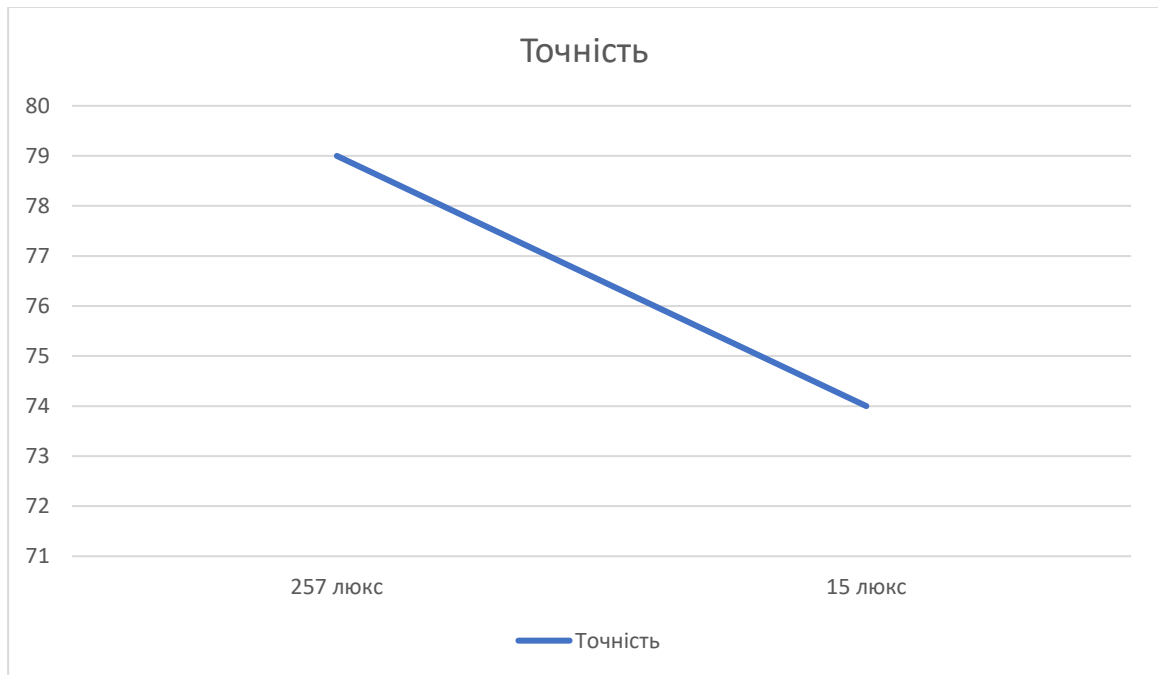


Рисунок 4.10 – Графік залежності точності від освітлення

Проведені експерименти показують високу ефективність системи комп'ютерного зору при різних умовах та його стійкість, що зараз є однією з найважливіших характеристик при використанні в різних умовах експлуатації. Ця система дозволить полегшити роботу операторів квадрокоптерів по виявленню та ідентифікації об'єктів з непоганою точністю на відстані 50 метрів та навіть при поганому освітленні.

4.4 Розрахунки освітлення приміщення

Комплексна система комп'ютерного зору для квадрокоптера була розроблена та реалізована в лабораторних умовах із використанням персонального комп'ютера. Враховуючи особливості експлуатації обладнання в обмеженому просторі та необхідність забезпечення безпечних і комфортних умов для праці, особливу увагу було приділено дотриманню нормативно-технічних вимог щодо мікроклімату та штучного освітлення приміщення. Параметри мікроклімату визначаються на основі сучасних стандартів, зокрема ДСТУ EN ISO 7730:2019 «Ергономіка теплового середовища» [24] та ДСН 3.3.6.042-99 «Державні санітарні норми мікроклімату виробничих приміщень»

[25]. Згідно з цими нормативними документами, для створення сприятливих умов праці необхідно підтримувати оптимальні температурні режими залежно від сезону (у холодний період – 22–24 °С, у теплий – 23–24 °С), відносну вологість повітря в межах 40–60% та швидкість його руху на рівні 0,1 м/с. Дотримання зазначених параметрів дозволяє знизити ризики розвитку професійних захворювань та підвищити ефективність роботи персоналу.

Окремо варто відзначити важливість раціонального проектування системи штучного освітлення, що визначає комфортність та безпеку виконання робіт. Розрахунок питомої потужності освітлення було здійснено відповідно до положень ДБН В.2.5-28:2018 «Природне і штучне освітлення» [26] та ДСТУ EN 12464-1:2016 «Світло і освітлення. Освітлення робочих місць. Частина 1 [27]. Робочі місця в приміщеннях». В процесі розрахунків використовувалась методика визначення питомої потужності освітлення за співвідношенням сумарної потужності встановлених світильників до площі освітлюваної поверхні. Так, при використанні двох led-світильників потужністю 3500 Лк. Для приміщення площею 20 м² питома потужність освітлення визначається як сумарна потужність ламп помножена на коефіцієнт використання світлового потоку на метр квадратний (4.1).

$$E = \frac{\Phi_{\text{заг}} * N * UF}{S} \quad (4.1)$$

Де E – середнє освітлення;

$\Phi_{\text{заг}}$ – світловий потік лампи; (переробити)

N – кількість ламп;

UF – коефіцієнт використання світлового потоку, дорівнює 0,5;

S – площа кімнати.

Підставляючи значення маємо наступне:

$$E = \frac{3500 * 2 * 0.5}{20} = 175 \text{ Лк/м}^2.$$

Ітогова освітленість в приміщенні становить 175 Лк/м². Згідно з вимогами ДБН В.2.5-28:2018, такі параметри забезпечують достатній рівень освітленості на робочих місцях.

Отримані результати підтверджують відповідність умов мікроклімату та освітлення встановленим нормативним вимогам, що забезпечує безпечне та ефективне функціонування комп'ютерного обладнання та комфортну роботу операторів системи комп'ютерного зору.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було проведено аналіз сучасних квадрокоптерів, їх способи керування та завдання які вони виконують. Також було проведено аналіз сучасних систем комп'ютерного зору, методи їх використання та програмування.

Була розроблена загальна структурна схема проекту, на якій зображені основні компоненти проекту. Був проведений аналіз компонентів, на базі якого був обраний необхідний елемент для проекту. Також був розроблений загальний алгоритм роботи програми для камери та ноутбука.

Розроблено програмне забезпечення для отримання відео від квадрокоптера та його подальший аналіз з використанням нейронної мережі для виявлення об'єктів.

Розроблено програмне забезпечення для ESP32-CAM для квадрокоптера, для передачі відео по локальній мережі.

Заключним етапом було експериментальна частина, де були проведені експерименти ефективності виявлення об'єктів при різних умовах, таких як освітлення, відстань об'єкта.

Проведені експерименти показують високу ефективність системи комп'ютерного зору при різних умовах та його стійкість, що зараз є однією з найважливіших характеристик при використанні в різних умовах експлуатації. Ця система дозволить полегшити роботу операторів квадрокоптерів по виявленню та ідентифікації об'єктів с непоганою точністю на відстані 50 метрів. Також результат експериментів показав високу точність розпізнавання при поганому освітленні яка дорівнює 15 люкс. При неідеальних умовах точність розпізнавання може сягати більше 80%.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. Структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.
2. Методичні вказівки з підготовки кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти денної і заочної форми навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І.Ш. Невлюдов, О.І. Филипенко, О.В. Токарева, С.П. Новоселов, О.В. Сичова. Харків: ХНУРЕ, 2023. 64 с.
3. Навчальний посібник з підготовки кваліфікаційної роботи бакалавра для здобувачів вищої освіти денної і заочної форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» : Навчальний посібник / І. Ш. Невлюдов, В.А. Андрусевич, О. В. Токарева, С. П. Новоселов, О. В. Сичова. – Харків : Видавництво Іванченка І. С., 2022. 151 с.
4. Кафедра комп'ютерно-інтегрованих технологій, автоматизації та робототехніки. Про нас. Офіційний сайт кафедри КІТАР ХНУРЕ . Режим доступу: <https://tapr.nure.ua/golovna/pro-nas>. (Дата доступу: 19.04.2025)
5. Положення про академічну доброчесність [Електронний ресурс]: Наказ ХНУРЕ від 02 лютого 2021 р. No 50. – Режим доступу: https://nure.ua/wpcontent/uploads/Main_Docs_NURE/polozhennja-pro-akademichnu-dobrochesnist.pdf.
6. Фотографія та технічні характеристики дрона E99. Режим доступу: <https://rozetka.com.ua/ua/415954218/p415954218/> (Дата доступу: 19.04.2025.)
7. Загальна інформація про Fpv-дрони. Режим доступу: [https://uk.wikipedia.org/wiki/FPV-%D0%B4%D1%80%D0%BE%D0%BD_\(%D0%B7%D0%B1%D1%80%D0%BE%D1%8F\)](https://uk.wikipedia.org/wiki/FPV-%D0%B4%D1%80%D0%BE%D0%BD_(%D0%B7%D0%B1%D1%80%D0%BE%D1%8F)) (Дата доступу: 19.04.2025)
8. Фото та технічні характеристики важкого Fpv-дрона Мамонт. Режим доступу: [https://texty.org.ua/fragments/111185/zsu-pochaly-zastosovuvaty-na-](https://texty.org.ua/fragments/111185/zsu-pochaly-zastosovuvaty-na)

peredovij-vazhki-fpv-drony-sho-rujnuyut-budivli-foto-video/ (Дата доступу: 19.04.2025.)

9. Фотографія та технічні характеристики гексакоптера . Режим доступу: https://vseplus.com/product/geksakopter-f550-388259?utm_source=google&utm_medium=cpc&utm_campaign=21393854607&utm_term=&gad_source=1&gclid=Cj0KCQjwqv2_BhC0ARIsAFb5Ac_FWvyvo5E-A9zhLH03O4-nliS5ueV0iiWxRo_xKygoCw-7yroFvekaAgiWEALw_wcB (Дата доступу: 19.04.2025)

10. Фото пульта керування. Режим доступу: <https://fx-drones.com/radioaparatura-radiomaster-tx12-mkii-elrs-pult-dystantsiinohokeruvannia/> (Дата доступу: 19.04.2025)

11. Інформація та фото про програму RC FLY. Режим доступу: https://play.google.com/store/apps/details/RC_FPV?id=com.cooingdv.rcfpv&hl=uk (Дата доступу: 19.04.2025)

12. Інформація про нейрону мережу MobileNetV2. Режим доступу: <https://www.kaggle.com/models/tensorflow/ssd-mobilenet-v2/tensorFlow2/ssd-mobilenet-v2/1?tfhub-redirect=true> (Дата доступу 19.04.2025)

13. Інформація про нейрону мережу MobileNetV3. Режим доступу: <https://github.com/diasirish/mobilenetv3> (Дата доступу 19.04.2025)

14. Інформація про нейрону мережу SqueezeNet. Режим доступу: <https://en.wikipedia.org/wiki/SqueezeNet> (Дата доступу: 19.04.2025)

15. Програмне забезпечення та інформація про нього – Режим доступу: <https://www.jetbrains.com/pycharm/> (Дата доступу: 10.05.2025)

16. Програмне забезпечення та інформація про нього – Режим доступу: <https://code.visualstudio.com/> (Дата доступу: 10.05.2025)

17. Програмне забезпечення та інформація про нього – Режим доступу: <https://jupyter.org/> (Дата доступу: 10.05.2025)

18. Інформація та фото про камеру FPV камера Caddx Ant Lite 4:3 – Режим доступу: <https://www.flydron.com.ua/fpv-drony-i-komplektuyushchiye/kamery-dlja-fpv/fpv-kamera-caddx-ant-lite-43> (Дата доступу 10.05.2025)

19. Інформація та фото про камеру Камера FPV нано RunCam Nano 2 2.1мм – Режим доступу: <https://arduino.ua/prod6100-fpv-videokamera-radiolink-800tv1-z->

peredavachem-5-8-ggc-48-kanaliv-200-

mvt?srsltid=AfmBOoo7yjEPDenxE3kIZR_vL6YGB3tngi_Jsvb4K3L8qmoBU_Selhe
I (Дата доступу: 10.05.2025)

20. Інформація та фото про плату ESP32-CAM – Режим доступу:
<https://ardushop.in.ua/arduino/esp32-cam-with-ov2640-camera-wifi-bluetooth> (Дата
доступу 10.05.2025)

21. Інформація про камеру до плати ESP32-CAM – Режим доступу:
https://arduino.ua/prod4501-modyl-kameri-2mp-ov2640-dlya-esp32-cam-fov160?srsltid=AfmBOooD43_0aAxTc0Ucb6Hx1ck5qRTYaOGmG4Lu_mwzLk8E4GEO56q0 (Дата доступу: 10.05.2025)

22. Інформація та фото квадрокоптера А6 – Режим доступу: https://dron-shop.com.ua/quadcopters/other_drones/ctw_ab_gps (Дата доступу 10.05.2025)

23. Програмне забезпечення ArduinoIDE – Режим доступу:
<https://www.arduino.cc/en/software/> (Дата доступу: 10.05.2025)

24. ДСТУ EN ISO 7730:2019 «Ергономіка теплового середовища» – Режим
доступу: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=28002
(Дата доступу: 28.05.2025)

25. ДСН 3.3.6.042-99 «Державні санітарні норми мікроклімату виробничих
приміщень» – Режим доступу: [https://online.budstandart.com/ua/catalog/doc-
page.html?id_doc=14283](https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=14283) (Дата доступу: 28.05.2025)

26. ДБН В.2.5-28:2018 «Природне і штучне освітлення» – Режим доступу:
https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=79885 (Дата
доступу: 28.05.2025)

27. ДСТУ EN 12464-1:2016 «Світло і освітлення. Освітлення робочих
місць. Частина 1. Робочі місця в приміщеннях» – Режим доступу:
https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=71838 (Дата
доступу: 28.05.2025)

28. Development of an Algorithm for ESP32-Cam Operation in HTTP Server
Mode for Streaming Video. Vladyslav Yevsieiev, Svitlana Maksymova, Nikolaj
Starodubcev, 2022. 178 с.