Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматизації, комп'ютерно-інтегрованих технологій та систем
(повна назва)
Кафедра Комп'ютерно-інтегрованих технологій, автоматизації, робототехніки та безпекової інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)
Розробка програмного забезпечення для СТЗ мобільної платформи
(тема)

Виконав:
здобувач 4 року навчання,
групи АКТАКІТ-22-2
Данило ТКАКІЩАК
(власне ім'я, прізвище)

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(повна назва освітньої програми)

Керівник доцент Наталія ДЕМСЬКА
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри КІТАРБІ _____ Ігор НЕВЛЮДОВ
(підпис) (власне ім'я, прізвище)

2026 р.

Я, Ткакіщак Данило 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.

"16" липня 2026 р. Данило ТКАКІЩАК

Харківський національний університет радіоелектроніки

Факультет АКІТС
Кафедра КІТАРБІ
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 _____
(підпис)
« 27 » квітня 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувачеві Ткакіщаку Данилу Віталійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка програмного забезпечення для СТЗ мобільної платформи
затверджена наказом університету від 04.06.2026 р. № 472 Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 16.07.2026 р.
3. Вихідні дані до роботи
3.1 Модель розпізнавання об'єктів YOLOv26
3.2 Підготовлений набір зображень з розміткою розташування об'єкту та класу для навчання моделі
3.3 Програмування мовою Python та розгортання моделі у середовищі Anaconda
3.4 Вбудована HD вебкамера ноутбука з роздільною здатністю 720p

4. Перелік питань, що потрібно опрацювати в роботі
4.1 Вступ
4.2 Визначення мети, об'єкта та предмета розробки
4.3 Аналіз предметної області: сучасні методи обробки зображень та алгоритми розпізнавання об'єктів
4.4 Вибір нейромережевої моделі для розробки програмного забезпечення
4.5 Розробка програмного забезпечення СТЗ мобільної платформи
4.6 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Демонстраційний матеріал у форматі (*.ppt), А4 – 11 сторінок

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 / терміни виконання етапів роботи	Примітка
1	Аналіз предметної області	28.04.26 – 05.05.26	виконано
2	Дослідження алгоритмів комп'ютерного зору, вибір програмних й апаратних засобів та проєктування системи	06.05.26 – 15.05.26	виконано
3	Розробка програмного забезпечення системи технічного зору мобільної платформи та навчання моделі розпізнавання	15.05.26 – 12.06.26	виконано
4	Тестування програмного забезпечення та аналіз отриманих результатів	12.06.26 – 19.06.26	виконано
5	Оформлення пояснювальної записки	20.06.26 – 01.07.26	виконано
6	Подання роботи на перевірку Інтернетсервісом StrikePlagiarisma	02.07.26	виконано
7	Подання роботи на рецензію	06.07.26	виконано
8	Подання роботи на підпис зав. кафедри	09.07.26	виконано
9	Подання кваліфікаційної роботи в ЕК	16.07.26	виконано

Дата видачі завдання 27.04.2026 р.

Здобувач _____ Данило ТКАКІЩАК
(підпис) (власне ім'я, прізвище)

Керівник роботи _____ доцент Наталія ДЕМСЬКА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 55 с., 1 табл., 18 рис., 2 дод., 12 джерел.

СИСТЕМА ТЕХНІЧНОГО ЗОРУ, МОБІЛЬНА ПЛАТФОРМА, НЕЙРОННА МЕРЕЖА, YOLO, OPENCV, ДЕТЕКЦІЯ ОБ'ЄКТІВ, ПОМІДОРИ.

Об'єкт розробки – процес автоматизованого розпізнавання об'єктів у системах технічного зору мобільних платформ.

Предмет розробки – методи, алгоритми та програмні засоби детекції й класифікації виробів за ступенем готовності на основі аналізу зображень.

Мета роботи – підвищення ефективності процесу автоматичного розпізнавання об’єктів у системі технічного зору мобільної платформи шляхом використання сучасної моделі YOLO та її донавчання.

У першому розділі проведено аналіз сучасних систем технічного зору, розглянуто основні етапи розвитку комп'ютерного зору та сфери застосування систем технічного зору в різних галузях.

У другому розділі досліджено сучасні алгоритми детекції об'єктів, виконано порівняння класичних та сучасних архітектур. Обґрунтовано вибір моделі YOLOv26 для реалізації системи технічного зору.

У третьому розділі розроблено програмне забезпечення системи технічного зору, реалізовано модулі отримання та обробки відеопотоку.

У результаті розроблено програмне забезпечення, яке забезпечує автоматичне виявлення та класифікацію виробів за ступенем готовності. Проведене тестування підтвердило працездатність розробленої системи та можливість її використання для автоматизованого моніторингу сільськогосподарських культур і як складової мобільних роботизованих платформ.

ABSTRACT

Explanatory note: 55 pages, 1 table, 18 figures, 2 appendices, 12 references.

MACHINE VISION SYSTEM, MOBILE PLATFORM, NEURAL NETWORK, YOLO, OPENCV, OBJECT DETECTION, TOMATOES.

The object of development is the process of automated object recognition in vision systems of mobile platforms.

The subject of development is methods, algorithms and software for detecting and classifying products by degree of readiness based on image analysis.

The purpose of the work is to increase the efficiency of the automatic object recognition process in the vision system of a mobile platform by using the modern YOLO model and its further training.

The first section analyzes modern machine vision systems, considers the main stages of development of computer vision and the scope of application of machine vision systems in various industries.

The second section investigates modern object detection algorithms, compares classical and modern architectures. The choice of the YOLOv26 model for implementing the machine vision system is justified.

In the third section, the software of the machine vision system is developed, and the modules for receiving and processing the video stream are implemented.

As a result, software was developed that provides automatic detection and classification of products by degree of readiness. The testing confirmed the operability of the developed system and the possibility of its use for automated monitoring of agricultural crops and as a component of mobile robotic platforms.

ЗМІСТ

Перелік скорочень…….. 7
Вступ…..……..... 8
1 Аналіз технічного завдання…..……..…….. 10
1.1 Роль систем технічного зору в мобільній робототехніці……...……10
1.2 Особливості застосування мобільних платформ для автоматизованого моніторингу……. 11
1.3 Обробка зображень та виявлення об’єктів…….……..……14
1.4 Сфери застосування систем комп’ютерного зору…….....…… 17
2 Алгоритми та моделі розпізнавання об’єктів……..……..……... 21
2.1 Класичні методи розпізнавання……..……. 21
2.2 Двоступеневі методи детекції об’єктів.……..……..…….. 22
2.3 Одноступеневі методи детекції об’єктів..……...…….... 25
2.4 Transformer-методи…….……...……..…….. 27
3 Розробка програмного забезпечення для СТЗ мобільної платформи…... 28
3.1 Аналіз задачі та вимоги до системи...……...28
3.1.1 Актуальність та мета розробки…….……..……..... 28
3.2.2 Вибір апаратного забезпечення системи……...29
3.2 Проєктування програмно-апаратного комплексу..……..……... 29
3.2.1 Вибір програмних компонентів……..……... 29
3.2.2 Вибір апаратного забезпечення системи……...31
3.3 Формування та підготовка набору даних..……...32
3.3.1 Збір та розмітка даних……33
3.3.2 Створення конфігураційного YAML-файлу датасету……. 34
3.4 Навчання моделі розпізнавання об’єктів..……... 35
3.4.1 Підготовка програмного середовища…… 35
3.4.2 Встановлення параметрів для навчання моделі……... 36
3.4.3 Навчання моделі……..... 37

3.5 Створення модуля розпізнавання..……..……..…….... 39
3.6 Донавчання моделі…….…… 42
3.7 Розрахунок стійкості програмного модуля…….…… 46
3.8 Безпека праці……..…… 48
Висновки……. 50
Перелік джерел посилання…… 51
Додаток А Лістинг програмного коду……..56
Додаток Б Демонстраційний матеріал……..65

ПЕРЕЛІК СКОРОЧЕНЬ

СТЗ – система технічного зору;

CNN – Convolutional Neural Networks (згорткові нейронні мережі);

OpenCV – Open Source Computer Vision Library (бібліотека комп'ютерного зору з відкритим кодом);

RPN – Region Proposal Network (мережа пропозицій регіонів);

SIFT – Scale-Invariant Feature Transform (масштабо-інваріантне перетворення ознак);

SURF – Speeded-Up Robust Features (прискорені стійкі ознаки);

YOLO – You Only Look Once (ви дивитеся лише раз).

ВСТУП

Стрімкий розвиток інформаційних технологій, робототехніки та методів штучного інтелекту сприяє широкому впровадженню систем технічного зору в різних сферах діяльності людини. Такі системи дозволяють автоматизувати аналіз візуальної інформації, забезпечуючи високу швидкість і точність виявлення, класифікації та відстеження об'єктів. Завдяки цьому вони активно використовуються у промисловості, транспорті, медицині, логістиці, системах безпеки та сільському господарстві [1].

Особливий інтерес становить застосування систем технічного зору у складі мобільних роботизованих платформ. Оснащення мобільних платформ камерами та сучасними алгоритмами комп'ютерного зору дозволяє їм автономно отримувати інформацію про навколишнє середовище, розпізнавати об'єкти, оцінювати їхній стан і приймати рішення відповідно до поставлених завдань [2].

Одним із перспективних напрямів використання мобільних платформ є аграрна галузь. Зростання потреби у підвищенні ефективності сільськогосподарського виробництва стимулює впровадження інтелектуальних систем, здатних автоматично контролювати стан рослин і плодів. Використання систем технічного зору дає змогу здійснювати моніторинг посівів, оцінювати врожайність, визначати ступінь стиглості плодів та своєчасно виявляти можливі відхилення під час їх розвитку [3].

Однією з практично важливих задач є автоматичне виявлення та класифікація виробів за ступенем готовності. Ручне оцінювання потребує значних витрат часу, залежить від людського фактора та є малоефективним при роботі з великими площами.

Метою кваліфікаційної роботи є підвищення ефективності процесу автоматичного розпізнавання об’єктів у системі технічного зору мобільної платформи шляхом використання сучасної моделі YOLO та її донавчання.

Об'єкт розробки – процес автоматизованого розпізнавання об'єктів у системах технічного зору мобільних платформ.

Предмет розробки – методи, алгоритми та програмні засоби детекції й класифікації виробів за ступенем готовності на основі аналізу зображень.

Для досягнення поставленої мети необхідно вирішити такі задачі:

– проаналізувати сучасні системи технічного зору та методи детекції об'єктів;

– дослідити сучасні алгоритми комп'ютерного зору та обґрунтувати вибір моделі для реалізації системи;

– спроєктувати структуру програмного забезпечення системи технічного зору мобільної платформи;

– реалізувати програмне забезпечення для виявлення та класифікації помідорів за ступенем стиглості;

– провести тестування розробленої системи та оцінити ефективність її роботи.

Практичне значення одержаних результатів полягає у створенні програмного забезпечення системи технічного зору, яке може бути використане як складова мобільних роботизованих платформ для автоматизованого моніторингу сільськогосподарських культур, оцінювання ступеня стиглості плодів, контролю врожайності та подальшого розвитку інтелектуальних аграрних систем.

Результати роботи відповідають цілям сталого розвитку (ЦСР), а саме цілі № 12: забезпечення переходу до раціональних моделей споживання і виробництва [4].

Кваліфікаційна робота оформлена відповідно до вимог ДСТУ 3008:2015 [5] та методичних рекомендацій з написання кваліфікаційної роботи [6].

1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Роль систем технічного зору в мобільній робототехніці

Під системою технічного зору розуміють сукупність програмних і апаратних засобів, що забезпечують отримання, обробку та аналіз візуальної інформації з метою автоматичного виявлення, класифікації або вимірювання характеристик об’єктів. На відміну від традиційних систем відеоспостереження, технічний зір не обмежується передачею зображення оператору, а виконує його інтелектуальну інтерпретацію та формує рішення на основі отриманих даних [7].

Особливої актуальності системи технічного зору набули у сфері мобільної робототехніки. Автономні роботи, безпілотні літальні апарати та інші мобільні платформи повинні оперативно сприймати навколишнє середовище, виявляти необхідні об’єкти, оцінювати їх розташування та приймати рішення без постійного втручання людини. У таких умовах камера стає одним із найважливіших сенсорів, що забезпечує отримання інформації про простір, у якому функціонує система [8].

Використання технічного зору дозволяє мобільним платформам виконувати широкий спектр завдань, серед яких автономна навігація, виявлення перешкод, розпізнавання об’єктів, контроль виробничих процесів, інспекція інфраструктури, моніторинг територій та оцінка стану навколишнього середовища. Сучасні алгоритми аналізу зображень забезпечують можливість роботи не лише у лабораторних умовах, а й у складних реальних сценаріях із мінливим освітленням, різноманітними погодними умовами та неоднорідним фоном [9].

Важливою особливістю мобільних систем є обмеженість обчислювальних ресурсів. На відміну від серверних рішень, які можуть використовувати потужні графічні процесори, автономні платформи часто

працюють на компактних одноплатних комп’ютерах або вбудованих процесорах. Це висуває підвищені вимоги до швидкодії алгоритмів, їхньої енергоефективності та здатності працювати в режимі реального часу.

Системи технічного зору в поєднанні з мобільними платформами активно застосовуються у складській логістиці, пошуково–рятувальних операціях, охоронних комплексах, екологічному моніторингу та промисловій автоматизації. В усіх цих випадках основною вимогою є надійне виявлення необхідних об’єктів із мінімальним часом затримки, що робить розвиток сучасних алгоритмів комп’ютерного зору одним із ключових напрямів розвитку робототехнічних систем [10].

Таким чином, технічний зір став невід’ємною складовою сучасних мобільних платформ. Його використання забезпечує можливість автоматичного аналізу навколишнього середовища та прийняття рішень на основі візуальних даних, що відкриває широкі перспективи для створення інтелектуальних систем різного призначення, у тому числі для задач моніторингу та розпізнавання об’єктів у природних умовах [11].

1.2 Особливості застосування мобільних платформ для автоматизованого моніторингу

Однією з ключових переваг мобільних платформ є можливість здійснювати аналіз інформації безпосередньо під час руху. На відміну від стаціонарних систем відеоспостереження, вони дозволяють отримувати дані з різних ракурсів і охоплювати значні площі, що особливо важливо для відкритих територій або протяжних об’єктів. Водночас така мобільність ускладнює обробку даних, оскільки програмне забезпечення повинно враховувати різні фактори, які впливають на якість отриманого зображення [11].

У більшості сучасних рішень саме камера виступає основним джерелом інформації про навколишнє середовище (рис. 1.1). Вона забезпечує високу

інформативність при відносно невеликій вартості обладнання та може використовуватися як окремо, так і разом з іншими сенсорами. На основі отриманих зображень система виконує пошук об’єктів, визначає їх розташування, оцінює розміри або інші характеристики, необхідні для подальшого прийняття рішень.



Рисунок 1.1 – Приклад використання мобільних роботизованих платформ для моніторингу сільськогосподарських насаджень [12, 13]

Для мобільних платформ характерні обмеження за масою, енергоспоживанням та продуктивністю обчислювального обладнання. Це означає, що алгоритми комп’ютерного зору повинні забезпечувати не лише високу точність, але й достатню швидкодію для роботи в режимі реального часу. На практиці затримка між отриманням зображення та видачею результату має бути мінімальною, особливо якщо результати аналізу використовуються для навігації або автоматичного керування [11].

Суттєвий вплив на якість роботи системи мають умови експлуатації. Під час роботи на відкритій місцевості камера може стикатися зі змінним природним освітленням, різкими тінями, відблисками, опадами, запиленістю або частковим перекриттям об’єктів. Крім цього, сама мобільна платформа може перебувати у русі, що призводить до появи розмиття кадру або зміни кута спостереження. Тому алгоритми аналізу повинні бути стійкими до подібних факторів і коректно працювати навіть за несприятливих умов.

Особливий інтерес становить використання мобільних платформ у задачах автоматизованого моніторингу рослинності та сільськогосподарських культур. Роботизовані системи можуть пересуватися між рядами насаджень або здійснювати спостереження за допомогою встановлених камер, формуючи інформацію про стан рослин і плодів. Це дозволяє автоматизувати процес збору статистики, оцінки врожайності та виявлення потенційних проблем на ранніх стадіях [12].

Прикладом такої задачі є виявлення плодів томатів у природному середовищі. Незважаючи на відносну простоту для людини, автоматичне розпізнавання помідорів супроводжується низкою труднощів. Плоди можуть частково закриватися листям або стеблами, перебувати в тіні, мати різний ступінь стиглості та відрізнятися за кольором. Додаткову складність створює неоднорідний фон і значна кількість дрібних деталей рослини. Саме тому системи комп’ютерного зору повинні бути здатними працювати в умовах природної варіативності сцени, а не лише у лабораторному середовищі.

Водночас доцільно розглядати задачу розпізнавання томатів як окремий випадок ширшої проблеми автоматичного виявлення об’єктів на мобільних платформах. Аналогічні методи можуть застосовуватися для ідентифікації інших плодів, елементів інфраструктури, транспортних засобів, дефектів поверхонь або технологічних об’єктів. Така універсальність забезпечує можливість адаптації програмного забезпечення до різних прикладних задач без суттєвої зміни загальної архітектури системи [13].

Розвиток сучасних засобів комп’ютерного зору сприяє створенню інтелектуальних мобільних комплексів, здатних виконувати не лише спостереження, а й попередній аналіз отриманих даних безпосередньо на борту. Це зменшує потребу у передачі великих обсягів відео на віддалені сервери, скорочує час реагування системи та підвищує ефективність автономної роботи.

Таким чином, мобільні платформи з інтегрованими системами технічного зору є перспективним напрямом розвитку сучасної автоматизації.

Їх використання забезпечує можливість оперативного моніторингу навколишнього середовища, локалізації об’єктів та підтримки прийняття рішень у різних сферах діяльності. Для задач виявлення плодів томатів такі системи можуть стати основою автоматизованих рішень, що підвищують ефективність спостереження та збору інформації в реальних польових умовах.

1.3 Обробка зображень та виявлення об’єктів

Обробка цифрових зображень є одним із ключових етапів функціонування сучасних систем технічного зору. Її основна мета полягає у виділенні корисної інформації із вхідних даних та формуванні ознак, необхідних для подальшого аналізу й прийняття рішень. Залежно від поставленої задачі програмне забезпечення може виконувати покращення якості зображення, локалізацію об'єктів, їх класифікацію, сегментацію або відстеження [14].

Традиційно процес аналізу зображень складається з декількох послідовних етапів. Спочатку відбувається отримання цифрового зображення з камери або іншого сенсора. Після цього можуть застосовуватися алгоритми попередньої обробки, спрямовані на зменшення шумів, корекцію яскравості та контрастності, фільтрацію або зміну кольорового простору. На наступних етапах виконуються пошук характерних ознак, виділення контурів або текстур та безпосереднє визначення наявності необхідних об'єктів.

Перші системи комп'ютерного зору ґрунтувалися переважно на класичних алгоритмах цифрової обробки зображень. Для виявлення об'єктів використовувалися методи порогової обробки, аналіз гістограм, пошук границь за допомогою операторів Собеля або Кенні, морфологічні перетворення, а також алгоритми шаблонного зіставлення. Такі методи демонстрували високу ефективність лише за умови стабільного освітлення, однорідного фону та незмінної форми об'єктів [15].

З розвитком обчислювальної техніки почали активно використовуватися дескриптори локальних ознак, серед яких широкого поширення набули Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Oriented FAST and Rotated BRIEF (ORB) та Histogram of Oriented Gradients (HOG). Вони дозволяли описувати характерні особливості зображення у вигляді компактних векторів та використовувалися для пошуку відповідностей між об'єктами, розпізнавання сцен і побудови систем навігації. Однак такі методи також вимагали ручного вибору ознак та втрачали ефективність при значних змінах масштабу, ракурсу або освітлення [16].

Наступним етапом розвитку стали алгоритми машинного навчання, у яких класифікація виконувалася на основі попередньо сформованих ознак. Для цього широко використовувалися метод опорних векторів (Support Vector Machine – SVM), дерева рішень, ансамблеві методи та інші математичні моделі. Хоча вони забезпечували кращі результати порівняно з традиційними алгоритмами, якість роботи значною мірою залежала від правильності вибору характеристик об'єкта, що вимагало участі фахівців та додаткової підготовки даних [17].

Суттєвий прорив у розвитку комп'ютерного зору відбувся із появою глибокого навчання. На відміну від класичних методів, нейронні мережі здатні автоматично формувати інформативні ознаки без необхідності їх ручного проектування. Це дозволило значно підвищити точність розпізнавання та забезпечити стійкість до змін масштабу, перспективи, освітлення і складності фону.

Особливу роль у розвитку сучасних систем відіграли згорткові нейронні мережі (Convolutional Neural Networks – CNN), які спеціально адаптовані для роботи із зображеннями. Їх архітектура дозволяє автоматично виділяти просторові залежності та формувати складні ознаки, починаючи від простих контурів і текстур до високорівневих представлень об'єктів. Саме цей підхід сьогодні лежить в основі більшості систем детекції.

У задачах мобільної робототехніки особливе значення має не лише точність, а й швидкість роботи алгоритмів. Платформа повинна аналізувати відеопотік практично без затримок, оскільки результати розпізнавання можуть використовуватися для навігації, уникнення перешкод або прийняття керуючих рішень. Тому при виборі методів необхідно враховувати компроміс між обчислювальною складністю та якістю виявлення [18].

У природних умовах задача автоматичного пошуку об'єктів додатково ускладнюється впливом зовнішніх факторів. Для прикладу, при виявленні плодів томатів система повинна коректно працювати за різних умов освітлення, наявності тіней, часткового перекриття листям, зміни кольору в процесі дозрівання та значної кількості схожих елементів на зображенні. У таких ситуаціях традиційні алгоритми, що базуються лише на кольорових або геометричних характеристиках, часто демонструють нестабільні результати.

Сучасні методи на основі глибокого навчання дозволяють враховувати складні взаємозв'язки між елементами зображення та виконувати пошук об'єктів навіть у випадках, коли вони частково приховані або мають значні відмінності у зовнішньому вигляді [19].

Важливою тенденцією останніх років стало створення компактних моделей, оптимізованих для роботи на вбудованих обчислювальних пристроях. Такі рішення забезпечують можливість виконання аналізу безпосередньо на мобільній платформі без необхідності постійного підключення до хмарних сервісів або високопродуктивних серверів.

Отже, розвиток методів комп'ютерного зору демонструє поступовий перехід від простих алгоритмів цифрової обробки до складних моделей штучного інтелекту, здатних автоматично аналізувати великі обсяги візуальної інформації. Для мобільних платформ, що виконують задачі пошуку та розпізнавання об'єктів у реальному середовищі, сучасні методи глибокого навчання є найбільш перспективним напрямом розвитку завдяки поєднанню високої точності, гнучкості та можливості роботи в складних експлуатаційних умовах.

1.4 Сфери застосування систем комп’ютерного зору

Розвиток технологій комп’ютерного зору та штучного інтелекту суттєво розширив можливості автоматизації процесів у різних галузях людської діяльності. Якщо раніше аналіз візуальної інформації переважно виконувався людиною, то сучасні алгоритми дозволяють автоматично виявляти, класифікувати та відстежувати об’єкти з високою точністю і швидкістю. Це сприяє підвищенню ефективності виробничих процесів, зменшенню витрат часу та мінімізації впливу людського фактора [20].

Однією з найважливіших сфер застосування систем комп’ютерного зору є промислова автоматизація. На виробничих підприємствах такі системи використовуються для контролю якості продукції, перевірки геометричних параметрів деталей, виявлення дефектів поверхонь та автоматичного сортування виробів. Завдяки високій швидкодії вони здатні аналізувати продукцію безпосередньо на конвеєрі, що дозволяє оперативно виявляти невідповідності та запобігати випуску бракованої продукції.

У транспортній галузі комп’ютерний зір використовується для розпізнавання дорожніх знаків, смуг руху, транспортних засобів і пішоходів. Такі технології є важливою складовою сучасних систем допомоги водієві та автономного керування транспортом. Аналіз відеопотоку в реальному часі дозволяє своєчасно реагувати на зміни дорожньої обстановки та підвищувати рівень безпеки [21].

Не менш важливим напрямом є застосування систем технічного зору в логістичних комплексах та автоматизованих складах. Роботизовані платформи використовують камери для орієнтації у просторі, розпізнавання товарів, зчитування маркування та контролю правильності комплектування замовлень. Це сприяє скороченню часу виконання складських операцій і зменшенню кількості помилок.

Значний інтерес викликає використання комп’ютерного зору в медицині, де алгоритми аналізують рентгенівські знімки, комп’ютерну

томографію, магнітно-резонансні дослідження та інші медичні зображення. Автоматизовані системи допомагають лікарям виявляти патологічні зміни на ранніх стадіях та використовуються як інструмент підтримки прийняття рішень.

Останніми роками особливо активно розвивається застосування систем комп’ютерного зору у сільському господарстві. Підвищення вимог до ефективності виробництва та необхідність оптимального використання ресурсів стимулюють впровадження інтелектуальних технологій для моніторингу посівів і автоматизації окремих етапів вирощування культур. У таких системах камери та алгоритми аналізу зображень дозволяють оперативно отримувати інформацію про стан рослин, розвиток плодів і можливі відхилення від нормального процесу росту [22].

Мобільні роботизовані платформи, оснащені системами технічного зору, можуть здійснювати пересування між рядами насаджень і виконувати безперервний візуальний контроль. Під час руху вони формують цифрові зображення, які аналізуються програмними засобами для локалізації окремих об’єктів, оцінки їхнього стану або визначення кількісних характеристик. Завдяки цьому можливо автоматизувати збір інформації, який раніше потребував значних трудових витрат.

Однією з перспективних задач є автоматичне виявлення плодів овочевих культур, зокрема томатів. Подібні системи можуть використовуватися для оцінювання врожайності, контролю дозрівання плодів, формування статистичних звітів та підтримки прийняття рішень щодо подальших агротехнічних заходів. Виявлення томатів на цифрових зображеннях дозволяє визначити їх приблизну кількість, просторове розташування та рівень зрілості без необхідності ручного обстеження великих площ [22].

Разом із тим розпізнавання томатів у природному середовищі пов’язане з рядом труднощів. Плоди можуть бути частково приховані листям або стеблами, знаходитися у тіні чи під прямим сонячним промінням, а також

мати різне забарвлення залежно від стадії дозрівання. Додатково на точність роботи впливають неоднорідний фон, наявність інших рослин та зміна кута спостереження внаслідок руху мобільної платформи.

У зв’язку з цим сучасні системи повинні бути достатньо універсальними та адаптивними, щоб забезпечувати стабільну роботу за реальних умов експлуатації. Важливою перевагою мобільних платформ є можливість виконання аналізу безпосередньо на місці збору інформації. Це дозволяє оперативно отримувати результати, не передаючи великі обсяги даних на віддалені сервери, що особливо актуально для польових умов із нестабільним мережевим покриттям.

Крім задач моніторингу, технології комп’ютерного зору можуть використовуватися як складова автономних роботизованих комплексів для збору врожаю [23]. У таких випадках система технічного зору виконує подвійне завдання: визначає місцезнаходження плоду та надає інформацію для планування подальших дій виконавчих механізмів. Навіть якщо робот не здійснює безпосередній збір урожаю, результати автоматичного розпізнавання можуть бути використані для побудови карт продуктивності ділянок, прогнозування врожайності або планування логістичних процесів.

Водночас принципи, що застосовуються при виявленні томатів, можуть бути легко адаптовані до інших задач. Аналогічні алгоритми використовуються для локалізації яблук, перцю, огірків, ягід, фруктів та інших об’єктів природного походження. Це свідчить про універсальність сучасних методів комп’ютерного зору та їхню здатність працювати з різними типами сцен без кардинальної зміни програмної архітектури.

Окремої уваги заслуговує інтеграція систем технічного зору з мобільними платформами невеликого розміру. Завдяки розвитку компактних обчислювальних модулів стало можливим виконання складних алгоритмів аналізу зображень безпосередньо на борту робота. Це відкриває перспективи створення автономних інтелектуальних систем, здатних виконувати моніторинг, навігацію та розпізнавання об’єктів у режимі реального часу без

постійного зв’язку із зовнішньою інфраструктурою.

Отже, сучасні системи комп’ютерного зору знаходять широке застосування в багатьох галузях, однак одним із найбільш перспективних напрямів є їх використання на мобільних роботизованих платформах для автоматизованого моніторингу об’єктів. Зокрема, задачі виявлення плодів томатів демонструють практичну цінність таких технологій і підтверджують доцільність подальшого розвитку методів автоматичного аналізу цифрових зображень у реальних умовах експлуатації.

2 АЛГОРИТМИ ТА МОДЕЛІ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ

2.1 Класичні методи розпізнавання

Перші системи детекції об'єктів базувалися на класичних алгоритмах комп'ютерного зору, у яких основна увага приділялася ручному виділенню ознак зображення. Подібні методи не використовували нейронні мережі, а працювали на основі математичних алгоритмів аналізу контурів, градієнтів, текстур та геометричних характеристик об'єктів [11].

Одним із найперших підходів був метод ковзного вікна (Sliding Window – SW). Його принцип полягав у послідовному переміщенні невеликої області по всьому зображенню з подальшою перевіркою кожної області класифікатором. Для врахування різних розмірів об'єктів зображення додатково масштабувалося. Хоч цей підхід і був простішим в реалізації, він, в свою чергу, мав високі обчислювальні витрати та низьку продуктивність виявлення для складних фонів та цілей з високою щільністю [24].

Важливим етапом розвитку класичних методів став алгоритм Віоли–Джонса, реалізований у вигляді Гаарових каскадів. Даний підхід використовував Гаароподібні ознаки та каскад слабких класифікаторів, які дозволяли швидко відкидати області, що не містять необхідного об'єкта, виходячи з різниці яскравості зображення в прямокутних областях [24]. Основною перевагою алгоритму була висока швидкість роботи навіть на слабкому обладнанні. Саме тому метод широко використовувався для детекції облич у цифрових камерах та ранніх системах відеоспостереження. Проте точність роботи алгоритму значно знижувалась при складному освітленні, зміні ракурсу або перекритті об'єкта.

Ще одним поширеним класичним підходом був метод HOG + SVM. У ньому використовувались гістограми орієнтованих градієнтів (Histogram of Oriented Gradients), які описували контури та структуру об'єкта обчислюючи

інформацію про напрямок локального градієнта зображення, а класифікація виконувалась за допомогою алгоритму Support Vector Machine [25]. Перевага цього методу полягала в його надійності, особливо для виявлення пішоходів. Однак метод усе ще залежав від ручного виділення ознак і погано адаптувався до складних сцен.

Також широкого поширення набули методи масштабно-інваріантного перетворення ознак (SIFT) та прискорених стійких ознак (SURF). Вони дозволяли знаходити ключові точки на зображенні та описувати їх спеціальними дескрипторами, стійкими до зміни масштабу та повороту. Такі алгоритми активно застосовувались у робототехніці, навігації та системах доповненої реальності. SURF реалізує вищу швидкість та кращу продуктивність, що робить його придатним для застосувань у реальному часі [26].

Основним недоліком класичних методів була обмежена універсальність та складність роботи зі складними реальними сценами, що стало причиною переходу до методів глибокого навчання.

2.2 Двоступеневі методи детекції об’єктів

Подальший розвиток систем детекції пов’язаний із появою глибоких нейронних мереж та двоступеневих (two-stage) алгоритмів. Головна особливість таких методів полягає у тому, що процес детекції виконується у два окремі етапи. На першому етапі система визначає потенційні області, у яких може знаходитися об’єкт, а на другому – класифікує знайдені області та уточнює їх координати [27].

Першим успішним алгоритмом цього типу став R-CNN (Regions with CNN Features). У даному підході спочатку генерувалися області-кандидати за допомогою селективного пошуку, після чого кожна область окремо передавалась у згорткову нейронну мережу для класифікації. Алгоритм продемонстрував значне підвищення точності порівняно з класичними

методами, однак мав низьку швидкість роботи через необхідність багаторазового запуску нейронної мережі. Для усунення цього недоліку було розроблено Fast R-CNN. У ньому згорткова нейронна мережа застосовувалась до всього зображення лише один раз, після чого області–кандидати вирізались уже з отриманої карти ознак. Це дозволило значно підвищити швидкість та точність роботи алгоритму [28.].

Наступним етапом розвитку став Faster R-CNN, у якому була реалізована мережа пропозицій регіонів (Region Proposal Network – RPN). На відміну від попередніх версій, генерація областей–кандидатів виконувалась безпосередньо нейронною мережею, одночасно виконуючи виявлення об'єктів та пропозицію регіонів [29]. Завдяки цьому вдалося значно покращити як швидкість, так і точність детекції. Faster R-CNN і сьогодні залишається одним із найбільш точних алгоритмів детекції об’єктів та використовується у медичних системах (виявлення ракових пухлин, пневмонії, захворювань сітківки ока), аналізі супутникових знімків та промислових системах контролю якості. На рисунку 2.1 наведено порівняння архітектур сімейства R-CNN.

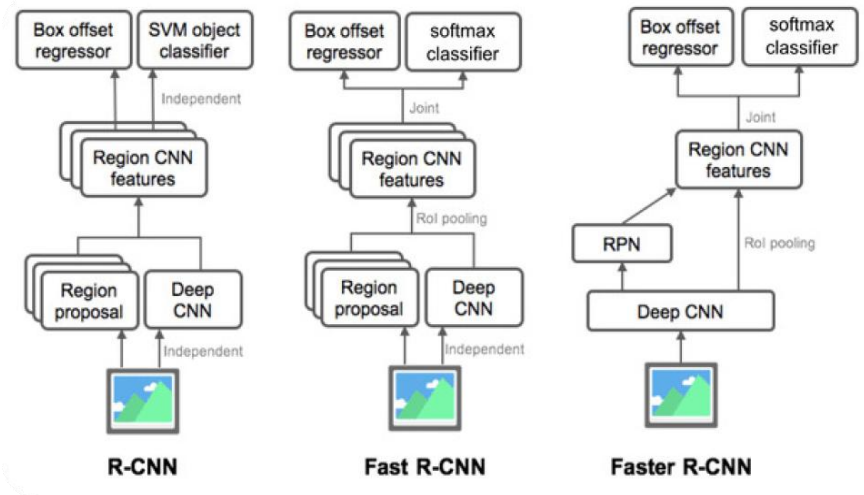


Рисунок 2.1 – Порівняння R-CNN, fast R-CNN та faster R-CNN [29]

Однією з головних проблем детекції об’єктів є робота з різними масштабами. На одному зображенні можуть одночасно бути присутні як

великі об’єкти на передньому плані, так і дуже маленькі – наприклад, люди вдалині, дорожні знаки або дрібні деталі.

Саме для вирішення цієї проблеми була створена архітектура FPN (Feature Pyramid Network). Її основна ідея полягає у побудові піраміди ознак різного масштабу, що складається з двох шляхів: висхідного шляху, де магістраль обробляє вхідне зображення для створення карт ознак з різною роздільною здатністю; менші шари зберігають дрібні просторові деталі для малих об'єктів, тоді як глибші шари зберігають деталі для більших об'єктів. Низхідний шлях потім вибірково аналізує глибші ознаки (наприклад, за допомогою 2-кратної білінійної інтерполяції) до вищої роздільної здатності, реконструюючи піраміду ознак [30]. На рисунку 2.2 наведено архітектуру мережі піраміди ознак.

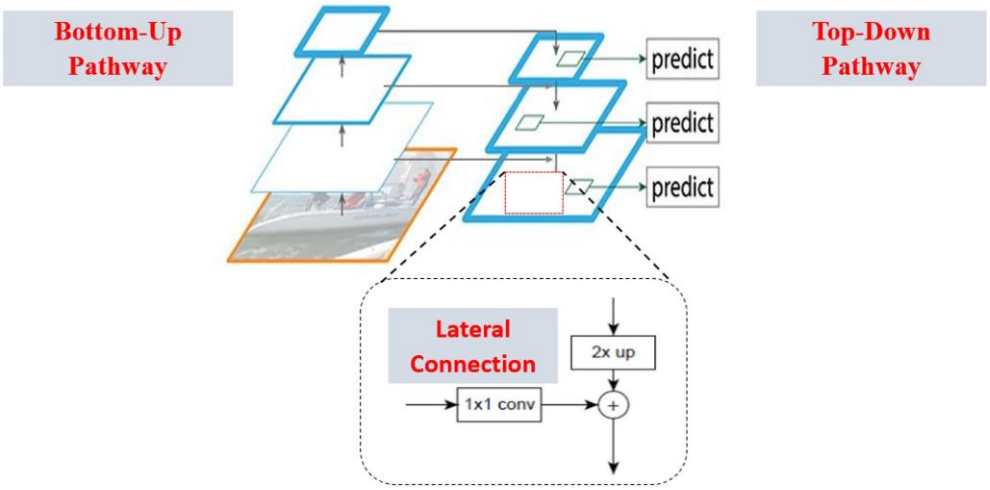


Рисунок 2.2 – Архітектура мережі піраміди ознак [30]

Основною перевагою двоступеневих методів є висока точність детекції та хороша робота з малими або складними об’єктами. Проте такі алгоритми потребують значних обчислювальних ресурсів і працюють повільніше за сучасні одноступеневі моделі.

2.3 Одноступеневі методи детекції об’єктів

Одноступеневі (one-stage) методи були розроблені з метою підвищення швидкості роботи систем детекції та забезпечення можливості роботи у реальному часі. На відміну від двоступеневих підходів, такі моделі виконують локалізацію та класифікацію об’єктів одночасно за один прохід нейронної мережі. В них було прибрано концепцію областей-кандидатів, і тепер, з метою покращити швидкість виявлення, безпосередньо прогнозується категорія та положення обмежувальної рамки на всьому зображенні.

Найвідомішим представником одноступеневих детекторів є сімейство YOLO. Перша версія була представлена у 2016 році Джозефом Редмоном і стала справжнім проривом у сфері виявлення об'єктів в реальному часі [31]. Головна ідея YOLO полягає у поділі зображення на сітку. Кожна клітинка сітки відповідає за прогнозування координатів об’єктів у своїй області, клас та ймовірність його наявності. На відміну від двоступеневих методів, модель аналізує все зображення одразу, а не окремі області. Це дозволяє отримати дуже високу швидкість роботи – десятки кадрів на секунду навіть на відносно слабких графічних картах. Перші версії YOLO мали значні проблеми з виявленням маленьких об'єктів, однак із кожною новою версією архітектура суттєво вдосконалювалась. На рисунку 2.3 зображено архітектуру першої версії моделі YOLO [31].

В YOLOv2 та YOLOv3 були додані anchor boxes, batch normalization, multi-scale detection та покращені backbone-мережі. Особливо важливим став multi-scale detection – модель почала краще працювати з малими об’єктами.

YOLOv4 та YOLOv5 – ці версії були орієнтовані на практичне використання. Було додано CSP connections, мозаїчну аугментацію, PANet/FPN та проведено оптимізації для графічних карт.

Ці моделі стали точнішими, стабільнішими, а також простішими в тренуваннях, завдяки підтримці PyTorch в 5 версії.

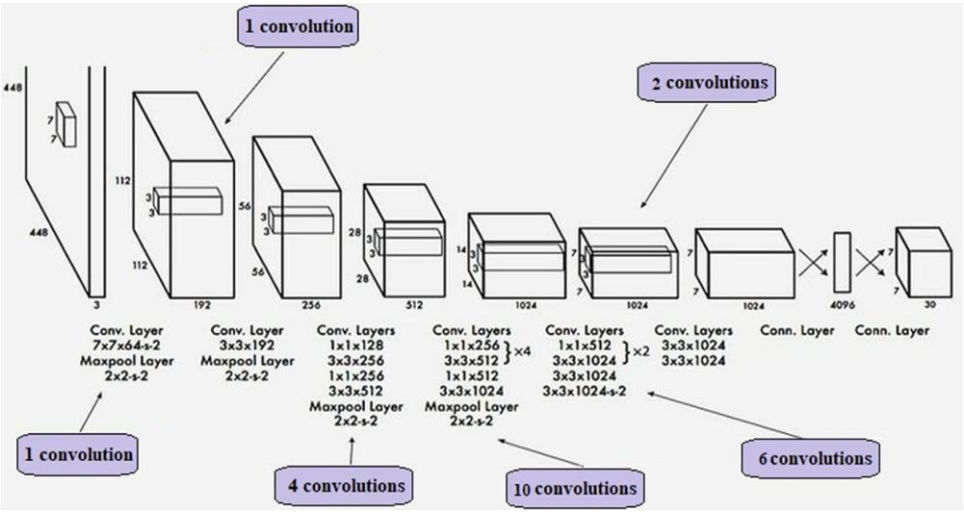


Рисунок 2.3 – Архітектура YOLOv1 [31]

Останні версії YOLO мають можливість виявляти об'єкти, сегментувати, проводити оцінку пози та стежити за об'єктами [32].

Вони активно використовуються у мобільній робототехніці, системах автопілотування, системах відеоспостереження, дронах та промислових системах автоматизації.

Іншим популярним одноступеневим методом є SSD (Single Shot Detector). Особливістю SSD є використання багатомасштабних карт ознак, що дозволяє ефективніше виявляти об’єкти різного розміру. Ключовим нововведенням є одночасне виявлення в кількох масштабах – ранні шари з високою роздільною здатністю виявляють малі об'єкти, тоді як глибші шари з низькою роздільною здатністю виявляють великі об'єкти [33]. Це усуває необхідність окремих етапів генерації кандидатів та вилучення ознак. Даний алгоритм забезпечує хороший баланс між швидкістю та точністю і часто використовується на вбудованих та edge-пристроях.

До сучасних одноступеневих алгоритмів також належить RetinaNet, в якому була представлена функція втрат Focal Loss. Її використання дозволило зменшити проблему дисбалансу між кількістю фонових областей (негативних прикладів) та реальних об’єктів під час навчання моделі.

RetinaNet поєднує Focal Loss з магістраллю ResNet-FPN та двома паралельними підмережами, налаштованими для класифікації та регресії

блоків, що дозволяє отримати точність Faster R-CNN з подібною швидкістю [34].

Головною перевагою одноступеневих методів є швидкість роботи та можливість використання у системах реального часу. Саме тому вони стали найбільш популярними у сучасних системах технічного зору.

2.4 Transformer-методи

Сучасним напрямком розвитку систем детекції об’єктів є використання transformer-архітектур. На відміну від класичних CNN-підходів, transformers використовують механізм attention, який дозволяє моделі аналізувати взаємозв’язки між різними частинами зображення [35].

Одним із перших успішних алгоритмів цього типу став DETR (Detection Transformer), розроблений компанією Meta. У даному підході згорткова нейронна мережа використовується для отримання карти ознак, після чого transformer-архітектура виконує глобальний аналіз сцени та безпосередньо прогнозує координати й класи об’єктів. Однією з головних особливостей DETR є відмова від anchor boxes та складних етапів post-processing, які використовувались у попередніх алгоритмах.

Transformer-підходи забезпечують високу точність та краще враховують взаємозв’язки між об’єктами на зображенні. Крім того, такі архітектури простіше масштабуються та краще адаптуються до складних сцен. Однак їхнім недоліком є висока обчислювальна складність і потреба у великих наборах даних для навчання.

Сьогодні активно розвиваються покращені версії transformer-моделей, зокрема Deformable DETR, RF-DETR та RT-DETRv2 та інші гібридні архітектури, які поєднують переваги CNN та transformers [32]. Подібні моделі вважаються одним із найперспективніших напрямків розвитку систем комп’ютерного зору та детекції об’єктів.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СТЗ МОБІЛЬНОЇ ПЛАТФОРМИ

3.1 Аналіз задачі та вимоги до системи

3.1.1 Актуальність та мета розробки

Актуальність розробки автоматизованої системи технічного зору мобільної платформи обумовлена стрімким розвитком технологій комп’ютерного зору, робототехніки та автоматизації аграрної галузі. У сільському господарстві все більшого значення набувають системи, здатні автоматично аналізувати стан рослин та врожаю без постійного втручання людини.

Однією з важливих задач аграрного виробництва є контроль ступеня стиглості овочів, зокрема помідорів. Використання систем технічного зору на основі нейромережевих алгоритмів дозволяє автоматизувати процес виявлення та класифікації плодів у режимі реального часу. Це підвищує швидкість аналізу, забезпечує безперервний моніторинг і створює основу для побудови автономних мобільних робототехнічних систем.

Використання автоматизованої системи технічного зору дозволяє:

- зменшити вплив людського фактора;
- підвищити швидкість обробки інформації;
- забезпечити безперервний моніторинг стану рослин;
- підвищити ефективність аграрних процесів;
- знизити витрати на ручне сортування та контроль продукції.

Особливо актуальним є поєднання системи технічного зору з мобільною платформою, оскільки це дозволяє проводити аналіз безпосередньо в умовах теплиць або відкритих полів. Такі системи можуть використовуватися для автоматизованого збору врожаю, сортування продукції, оцінювання стану рослин та оптимізації аграрних процесів.

Метою розробки автоматизованої системи технічного зору мобільної платформи є створення програмного забезпечення, здатного в автоматичному режимі виявляти та класифікувати помідори за ступенем стиглості на основі аналізу відеопотоку, зображень чи відео.

Основною задачею системи є класифікація виявлених об’єктів на три категорії:

– стиглі помідори;
– помідори, що достигають;
– нестиглі помідори.

3.1.2 Вимоги до системи

Головна функція системи – розпізнавання помідорів та їх класифікація за ступенем стиглості. В ході виконання цієї задачі система повинна отримувати зображення (або відеопотік) з камери, виконувати попередню обробку зображення, виявляти та класифікувати помідори, і відображати результати розпізнавання за допомогою кольорових позначок, рамок чи написів, які можуть бути реалізовані обраними в подальшому програмними засобами.

Система повинна мати зрозумілий інтерфейс, забезпечувати стабільну точність виявлення за різних умов освітлення, надійність роботи, підтримувати роботу з різними видами відеокамер. Використовувати засоби, що дозволять запускатися як на Windows, так і на Linux. Підтримувати можливість подальшого навчання моделі на нових даних та мати можливість масштабування і модернізації.

3.2 Проєктування програмно-апаратного комплексу

3.2.1 Вибір програмних компонентів

Для реалізації системи технічного зору мобільної платформи було обрано сучасні програмні засоби та бібліотеки, що забезпечують можливість

розробки, навчання та запуску нейромережевих моделей комп’ютерного зору.

Основною мовою програмування системи було обрано Python [36]. Ця мова широко використовується у сфері штучного інтелекту, машинного навчання та комп’ютерного зору завдяки простоті синтаксису, великій кількості спеціалізованих бібліотек та активній спільноті розробників. Крім того, Python підтримує апаратне прискорення обчислень через CUDA та GPU NVIDIA, що є важливим для роботи моделей YOLO.

Для роботи із зображеннями та відеопотоком використовується бібліотека OpenCV. Вона є однією з найпоширеніших бібліотек комп’ютерного зору та містить велику кількість функцій для обробки графічних даних.

У межах розроблюваної системи OpenCV використовується для:

– отримання відеопотоку з вебкамери;
– зчитування фотографій та відеофайлів;
– попередньої обробки кадрів;
– масштабування зображень;
– відображення результатів детекції;
– роботи з графічним інтерфейсом системи.

Бібліотека забезпечує високу швидкодію та добре інтегрується з Python і нейромережевими моделями.

Для реалізації функцій детекції та класифікації помідорів було обрано модель YOLO з бібліотекою Ultralytics. Дана модель належить до сімейства одноступеневих детекторів об’єктів та забезпечує високу швидкість роботи при достатній точності розпізнавання. Ultralytics дозволяє швидко запускати тренування на власних даних, виконувати перевірку якості моделі та застосовувати її для детекції в реальному часі. На рисунку 3.1 зображено порівняння різних версій YOLO та інших моделей детекції об’єктів.

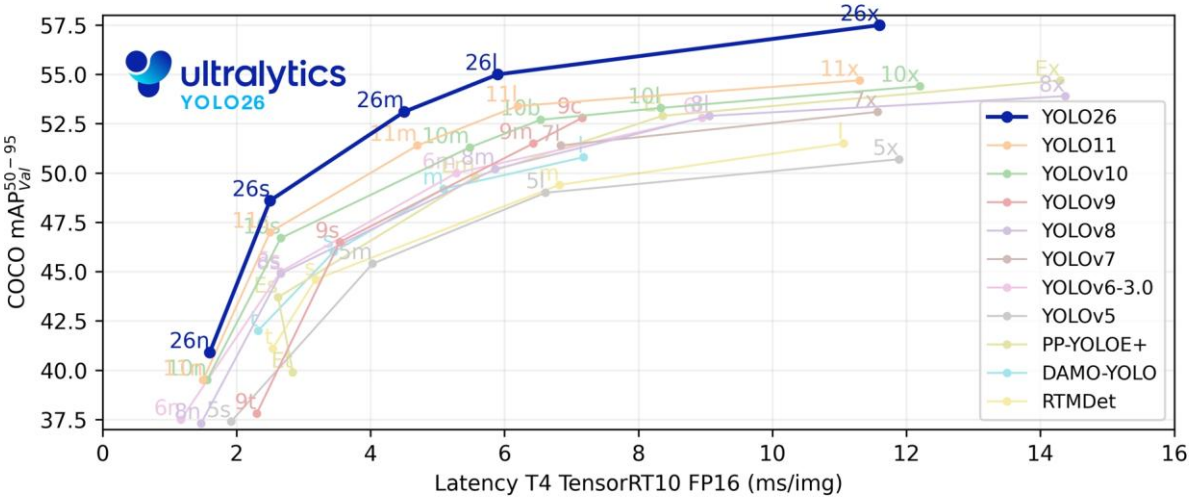


Рисунок 3.1 – Порівняння останніх версій YOLO [32]

Виходячи з даних графіку, нова серія YOLO26 має кращу точність за схожої затримки обробки одного зображення. Розроблена для периферійних та малопотужних пристроїв вона пропонує оптимізований дизайн, який усуває зайву складність, одночасно інтегруючи цільові інновації для забезпечення швидшого та доступнішого розгортання [32]. Її й буде обрано для розробки, оскільки вона пропонує оптимальний баланс між швидкістю роботи та точністю при обмежених обчислювальних ресурсах.

У системі модель буде навчатися на невеликому наборі, до ~500 зображень, з чим може впоратися її small модель без значної втрати продуктивності.

Для організації програмного середовища буде використано платформу Anaconda. Вона має зручне керування бібліотеками та віртуальними середовищами Python [36], стабільну роботу програмного забезпечення і допомагає уникати конфліктів між версіями пакетів.

3.2.2 Вибір апаратного забезпечення системи

Для реалізації системи технічного зору мобільної платформи було обрано апаратне забезпечення, яке має достатню продуктивність для роботи алгоритмів комп’ютерного зору та нейромережевих моделей.

Процесор Intel Core i5-10500H буде забезпечувати виконання задач попередньої обробки зображень, роботи програмного інтерфейсу та взаємодії між компонентами системи.

Для прискорення обчислень використовується дискретна відеокарта NVIDIA GeForce RTX 3050 з підтримкою технології CUDA. Графічний процесор дозволяє значно збільшити швидкість роботи нейромережевої моделі під час аналізу фотографій та відеопотоку. Використання GPU є особливо важливим для роботи системи в режимі реального часу.

Обсяг оперативної пам’яті становить 8 ГБ DDR4, що є достатнім для запуску середовища розробки, бібліотек комп’ютерного зору та обробки вхідних даних.

Для отримання відеоданих використовується вбудована HD вебкамера ноутбука з роздільною здатністю 720p. Вебкамера забезпечує отримання відеопотоку в режимі реального часу та використовується для тестування алгоритмів детекції та класифікації помідорів.

3.3 Формування та підготовка набору даних

Для навчання нейромережевої моделі YOLO було взято набір даних із зображень плантацій з плодами помідорів різного ступеня стиглості. Якість та різноманітність датасету є важливими факторами, що безпосередньо впливають на точність роботи системи технічного зору.

Основною метою підготовки набору даних було створення навчальної вибірки, яка дозволить моделі коректно виявляти помідори та класифікувати їх за ступенем стиглості в різних умовах освітлення, з різних ракурсів, різним фоном та частковим перекриттям іншими об’єктами. Це дозволяє підвищити стійкість моделі до змін умов роботи системи в реальному середовищі.

3.3.1 Збір та розмітка даних

Для навчання нейромережевої моделі було використано готовий набір даних, що містив 277 зображень помідорів різного ступеня стиглості. Початковий датасет був представлений у форматі COCO та містив JSON-файли з анотаціями об’єктів. У файлах анотацій зберігалася інформація про координати обмежувальних рамок (bounding boxes), класи об’єктів та параметри зображень.

Оскільки модель YOLO використовує власний формат розмітки даних, виникла необхідність у перетворенні структури датасету. Для цього було розроблено Python-скрипт, який автоматично виконував конвертацію анотацій із формату COCO у формат YOLO.

```
from ultralytics.data.converter import convert_coco

convert_coco(
    labels_dir="dataset/annotations/",
    save_dir="convdataset/converted/",
    cls91to80=False,
)
```

У процесі перетворення JSON-анотації оброблялися програмно, після чого для кожного зображення створювався окремий текстовий файл із координатами об’єктів та відповідними класами у форматі YOLO.

Після конвертації датасету був написаний додатковий Python-скрипт для автоматизованого розподілу зображень та файлів анотацій між папками train та valid у співвідношенні 90% до 10% (див. додаток А). Такий поділ використовується для формування навчальної та валідаційної вибірок.

Навчальна вибірка використовується безпосередньо для навчання нейромережевої моделі, тоді як валідаційна вибірка застосовується для оцінювання якості роботи моделі під час навчання. Це дозволяє контролювати точність детекції, виявляти перенавчання моделі та оцінювати

її здатність узагальнювати інформацію на нових даних.

Автоматизація процесу підготовки датасету дозволила значно спростити роботу з великим обсягом зображень, зменшити ймовірність помилок під час формування структури набору даних та забезпечити сумісність датасету з моделлю YOLO.

3.3.2 Створення конфігураційного YAML-файлу датасету

Для навчання моделі YOLO необхідно створити конфігураційний файл формату YAML, який містить інформацію про структуру датасету, шляхи до навчальних і валідаційних зображень, а також перелік класів об’єктів.

Для автоматизації процесу був реалізований Python-скрипт, що формує YAML-файл програмно. Він дозволить уникнути помилок при ручному редагуванні та спрощує масштабування датасету при зміні кількості класів або структури папок.

```
import yaml
# Назви класів
classes = [
    "unripe",
    "semi-ripe",
    "fully-ripe"
]

data = {
    "path": "content/data",
    "train": "train/images",
    "val": "validation/images",
    "nc": "3",
    "names": classes
}

with open("/content/data.yaml", "w") as file:
```

```
        yaml.dump(data, file, sort_keys=False)

print("data.yaml створено")
```

3.4 Навчання моделі розпізнавання об’єктів

Після підготовки датасету наступним етапом є навчання моделі розпізнавання об’єктів. На цьому етапі нейронна мережа аналізує набір зображень та відповідних анотацій, поступово навчаючись визначати розташування та клас об’єктів на зображенні. Якість навчання напряму залежить від структури датасету, коректності анотацій, а також вибору архітектури моделі та параметрів тренування.

Процес навчання виконувався із використанням попередньо підготовленого датасету, розділеного на train та validation вибірки, що дозволяє оцінювати якість моделі не лише на тренувальних даних, але й на нових зображеннях, які не брали участі у навчанні.

3.4.1 Підготовка програмного середовища

Перед початком навчання моделі необхідно виконати налаштування програмного середовища та встановити всі необхідні інструменти для роботи з нейронними мережами. Для реалізації системи було використано мову програмування Python, оскільки вона має велику кількість бібліотек для машинного навчання, комп’ютерного зору та роботи з моделями YOLO. Також було встановлено необхідні бібліотеки для обробки зображень, роботи з датасетами та запуску процесу тренування моделі (рис. 3.2).

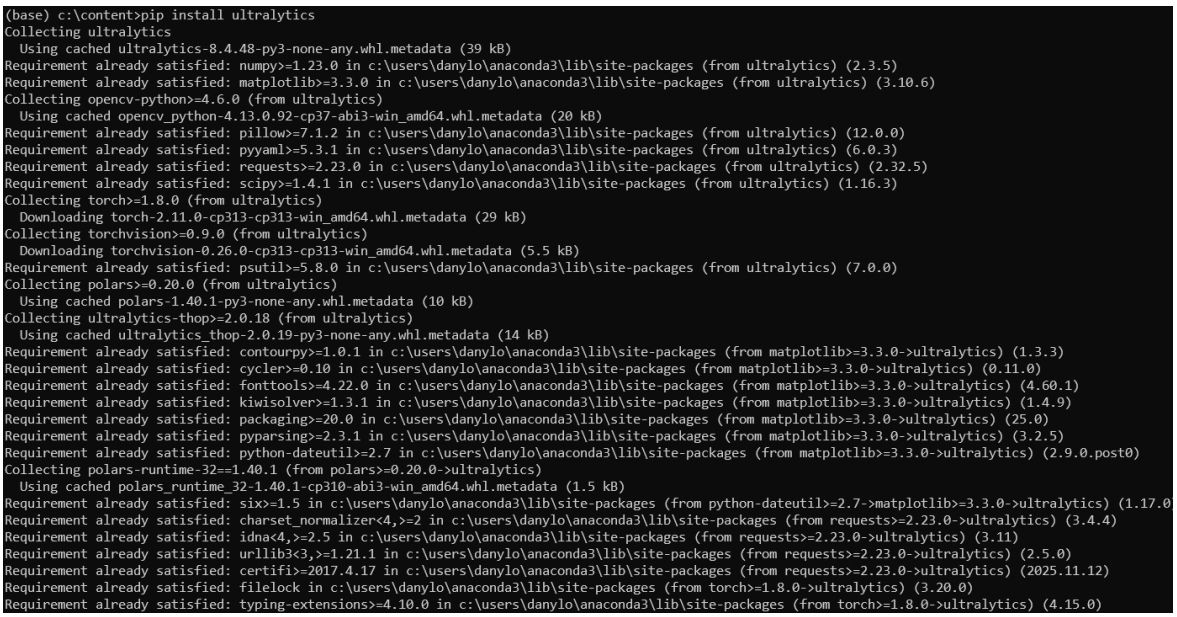


Рисунок 3.2 – Встановлення необхідних бібліотек

Для навчання моделі використовувалась бібліотека Ultralytics, яка забезпечує зручний інтерфейс для роботи з моделями YOLO та дозволяє виконувати тренування, валідацію і тестування нейронної мережі. Також було встановлено CUDA-драйвер для використання графічного процесора.

3.4.2 Встановлення параметрів для навчання моделі

Перед запуском процесу навчання необхідно налаштувати основні параметри моделі, які впливають на швидкість навчання, стабільність роботи нейронної мережі та точність розпізнавання об’єктів. До таких параметрів належать кількість епох навчання, розмір batch, роздільна здатність вхідних зображень та ін.

В нашому випадку параметри навчання підбиралися з урахуванням характеристик апаратного забезпечення та невеликого розміру датасету. Було написано Python-скрипт

```
from ultralytics import YOLO
model = YOLO("yolo26s.pt")
results = model.train(data="data.yaml", epochs=50, imgsz=640, patience=10)
```

для запуску тренування моделі з наступними параметрами:

– epochs = 50 – оптимальна кількість для невеликих датасетів (<1000 зображень);

– imgsz = 640 – стандартний розмір зображення, який забезпечує баланс між деталізацією і швидкістю обробки;

– patience = 10 – параметр зупинки навчання за відсутності покращень протягом 10 епох.

3.4.3 Навчання моделі

Алгоритм навчання проаналізує зображення в каталогах навчання та валідації, а потім розпочне навчання моделі (рис. 3.3). В кінці кожної епохи навчання програма запускає модель на валідаційній вибірці та повідомляє про отриману mAP, точність та повноту. У міру продовження навчання mAP, як правило, має зростати з кожною епохою.

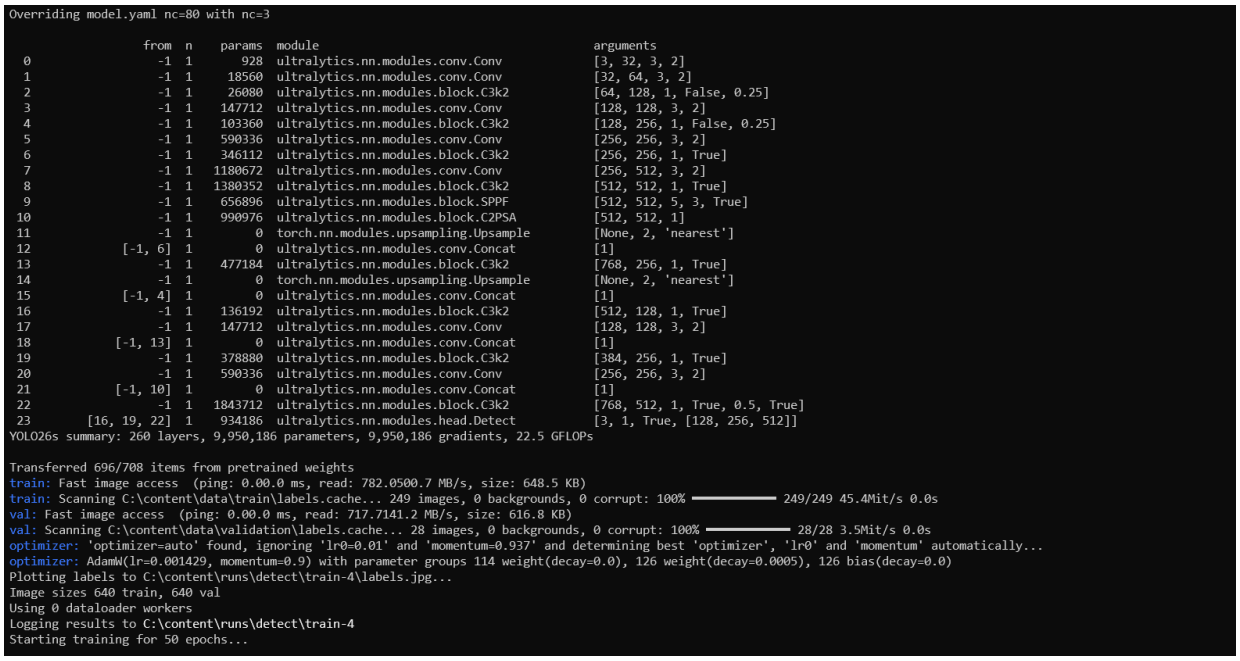


Рисунок 3.3 – Запуск процесу навчання

Процес навчання моделі тривав 36 епох, і був зупинений параметром patience, оскільки покращень не було останніх 10 епох (рис. 3.4).

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      32/50      5.11G      1.167     0.5659    0.00294        101        640: 100% ──────────── 16/16 2.4it/s 6.7s
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ──────────── 1/1 1.9it/s 0.5s
                   all         28        268       0.68      0.875      0.846      0.545

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      33/50      5.11G      1.187     0.5583   0.002857        163        640: 100% ──────────── 16/16 2.3it/s 6.9s
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ──────────── 1/1 3.0it/s 0.3s
                   all         28        268       0.685      0.877      0.832      0.538

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      34/50      5.11G      1.163     0.5295   0.002788        102        640: 100% ──────────── 16/16 2.1it/s 7.6s
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ──────────── 1/1 3.0it/s 0.3s
                   all         28        268       0.788      0.765      0.809      0.533

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      35/50      5.11G       1.16     0.5415   0.002861         58        640: 100% ──────────── 16/16 2.4it/s 6.7s
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ──────────── 1/1 2.4it/s 0.4s
                   all         28        268       0.789      0.786      0.835      0.553

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      36/50      5.11G      1.164     0.5489   0.002746        112        640: 100% ──────────── 16/16 2.2it/s 7.4s
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ──────────── 1/1 2.5it/s 0.4s
                   all         28        268       0.727      0.823      0.826      0.546
EarlyStopping: Training stopped early as no improvement observed in last 10 epochs. Best results observed at epoch 26, best model saved as best.pt.
To update EarlyStopping(patience=10) pass a new patience value, i.e. `patience=300` or use `patience=0` to disable EarlyStopping.

36 epochs completed in 0.099 hours.
Optimizer stripped from /content/runs/detect/train/weights/last.pt, 20.3MB
Optimizer stripped from /content/runs/detect/train/weights/best.pt, 20.3MB

Validating /content/runs/detect/train/weights/best.pt...
Ultralytics 8.4.49 🚀 Python-3.12.13 torch-2.10.0+cu128 CUDA:0 (Tesla T4, 14913MiB)
YOLO26s summary (fused): 122 layers, 9,466,341 parameters, 0 gradients, 20.5 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ──────────── 1/1 2.8it/s 0.4s
                   all         28        268      0.805      0.806      0.851      0.567
                unripe         27        188      0.921      0.934      0.965      0.605
             semi-ripe         16         37      0.695       0.74      0.697      0.476
            fully-ripe         22         43        0.8      0.743      0.891       0.62
Speed: 0.2ms preprocess, 6.4ms inference, 0.0ms loss, 0.4ms postprocess per image
```

Рисунок 3.4 – Результати навчання

На рисунках 3.5 і 3.6 зображено результати з валідаційної вибірки з мітками та результатами розпізнавання об’єктів з ймовірностями.

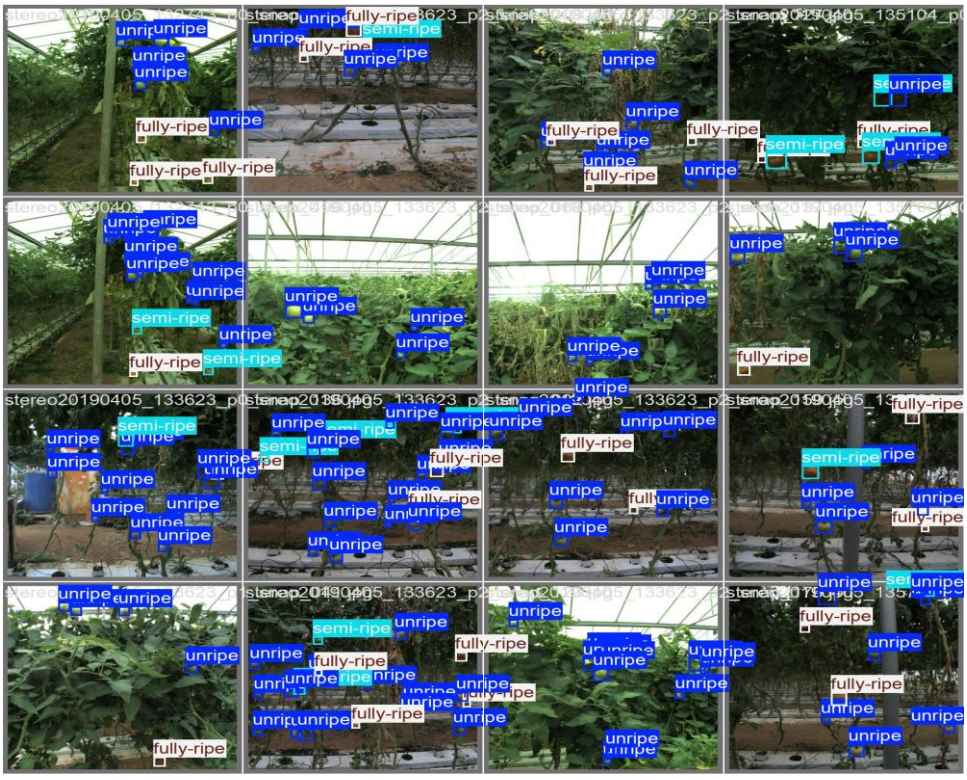


Рисунок 3.5 – Зображення з мітками валідаційного набору



Рисунок 3.6 – Зображення з результатами розпізнавання

3.5 Створення модуля розпізнавання

Було розроблено Python-скрипт (див. додаток А), який забезпечує завантаження натренованої моделі, аналіз вхідних зображень або відеопотоку та відображення результатів детекції у режимі реального часу.

У якості основи було використано модель YOLO, навчена на підготовленому наборі даних із зображеннями томатів різного ступеня стиглості. Після завершення навчання модель експортувалась у файл `best.pt`, який надалі використовується програмою для виконання розпізнавання.

Розроблений модуль підтримує три режими роботи:

– аналіз окремих фотографій;

– обробку відеофайлів;

– роботу з USB-камерою у реальному часі.

Після запуску програми користувач обирає необхідний режим роботи.

Для фотографій та відео додатково задається шлях до файлу, після чого виконується передача зображення до моделі YOLO. У результаті роботи алгоритму формується новий кадр із нанесеними рамками навколо знайдених об’єктів, а також підписами класів і значенням достовірності розпізнавання.

У процесі розробки було реалізовано автоматичне масштабування великих зображень для коректного відображення результатів на екрані користувача. Це дозволило уникнути проблем із частковим відображенням кадру при роботі із зображеннями високої роздільної здатності.

На рисунку 3.7 зображено приклад розпізнавання томатів на статичному фото.

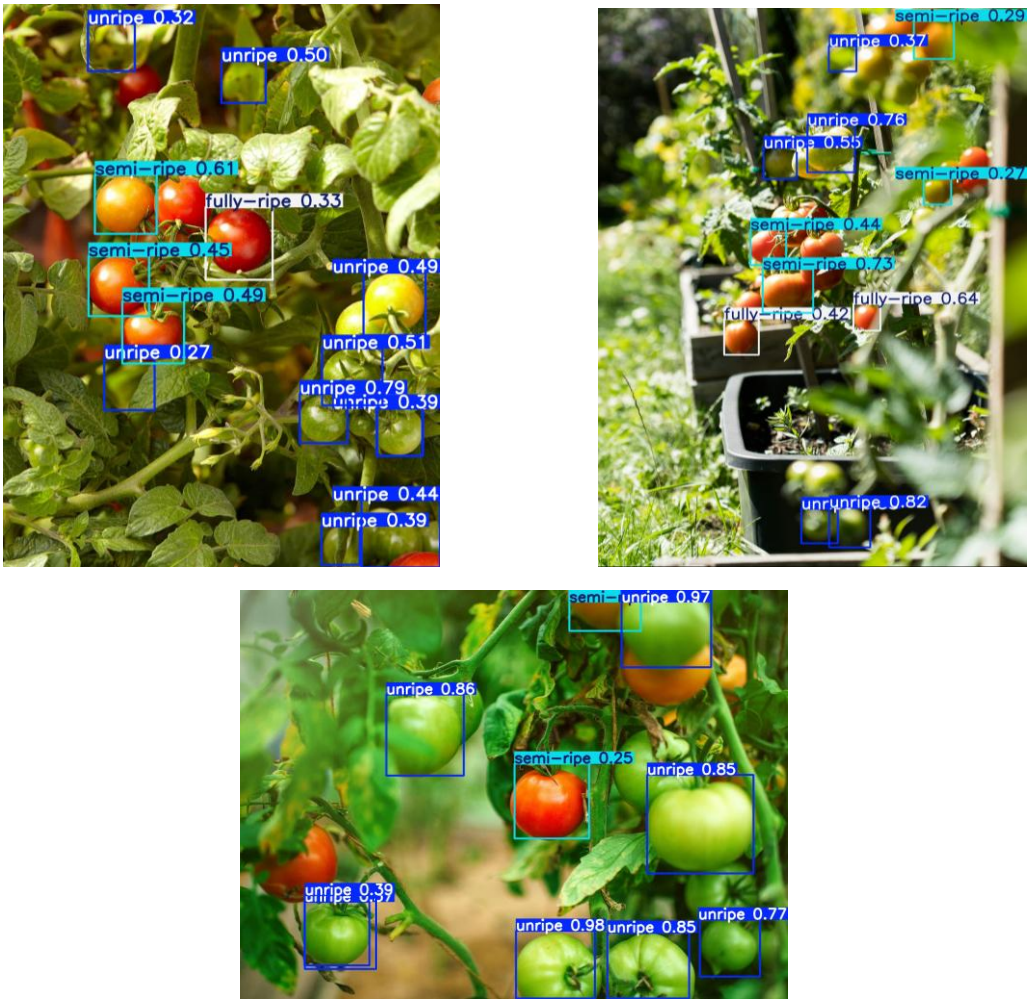


Рисунок 3.7 – Приклад розпізнавання томатів на фото

На рисунку 3.8 зображено фрагмети відео, на якому була проведена детекція томатів.

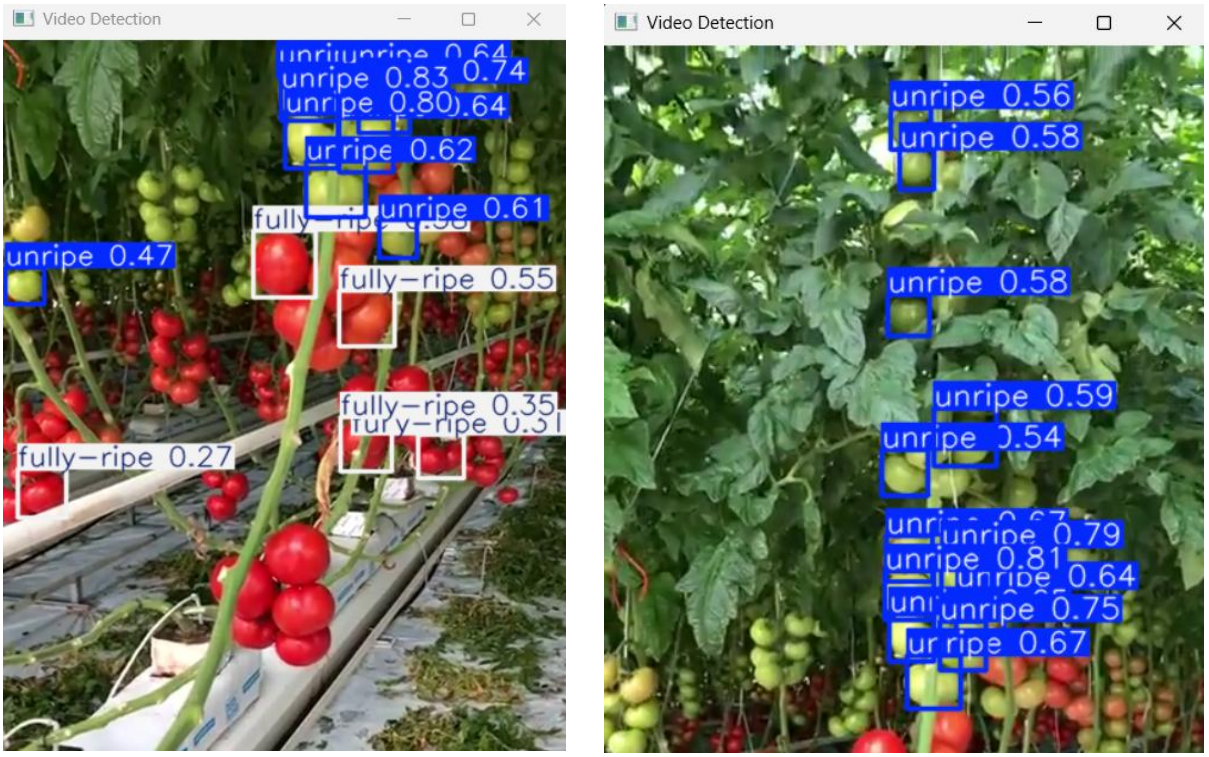


Рисунок 3.8 – Приклад розпізнавання томатів на відео

Для демонстрації прикладу використання вбудованої веб-камери для детекції, я використав зображення томатів, що показував з телефону (рис. 3.9).

У такому сценарії система змогла частково коректно визначати об’єкти, хоча якість детекції була нестабільною через низку факторів. Зокрема, впливали відблиски на екрані телефону, зменшена різкість зображення при зйомці з веб-камери, а також додаткова затримка та компресія відеопотоку. Попри це, модель продемонструвала здатність знаходити ключові ознаки об’єктів у реальному часі, що підтверджує її працездатність у наближених до реальних умовах.

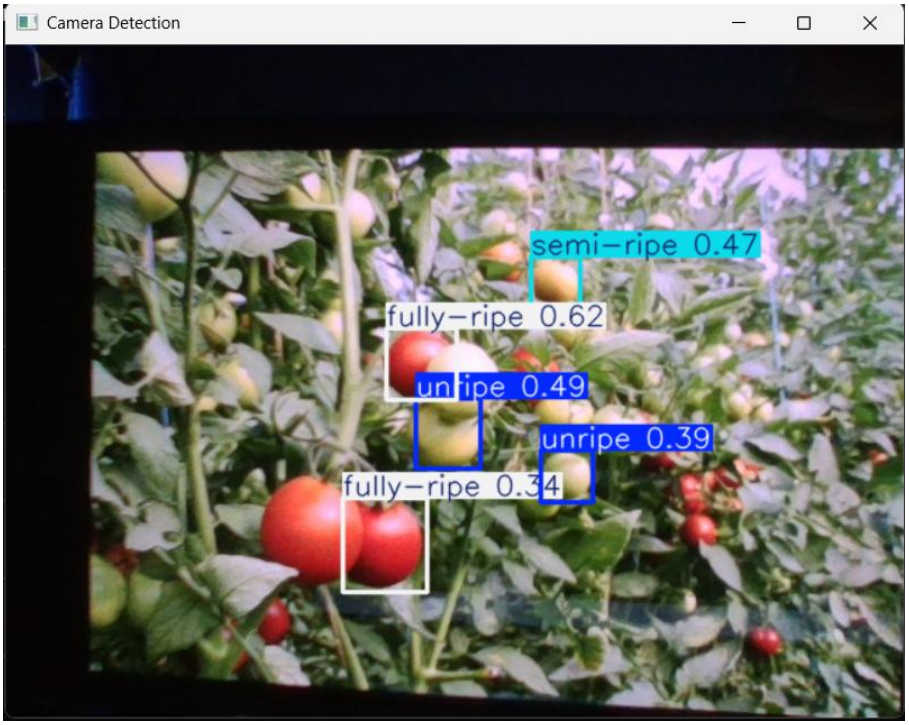


Рисунок 3.9 – Приклад розпізнавання томатів через веб-камеру

У результаті проведеного тестування модель продемонструвала загалом хороші результати розпізнавання томатів різного ступеня стиглості. Система коректно працювала як із зображеннями, так і з відеопотоком у реальному часі, що підтверджує можливість її практичного використання. Водночас було помічено, що повністю дозрілі томати визначаються дещо гірше, ніж інші класи. Це може бути пов’язано з недостатньою кількістю відповідних прикладів у навчальному наборі або схожістю кольорів і форми між окремими категоріями. Для покращення точності було прийнято рішення розширити датасет та провести додаткове навчання моделі на прикладі зображень, які допоможуть краще визначати стиглі помідори, їх скупчення та за відсутності перешкод.

3.6 Донавчання моделі

В ході попереднього тестування моделі розпізнавання, було виявлено недостатню точність в роботі з достигшими томатами, тому задля покращення якості детекції було зібрано новий набір з 1012 зображень,

переважно скупчень достигших томатів, які частково перекриваються або добре помітні в кадрі.

Для розмітки даних на зображеннях було обрано платформу Label Studio (рис. 3.10). Для коректного навчання моделі класи розмітки повинні співпадати за порядком з минулим набором.

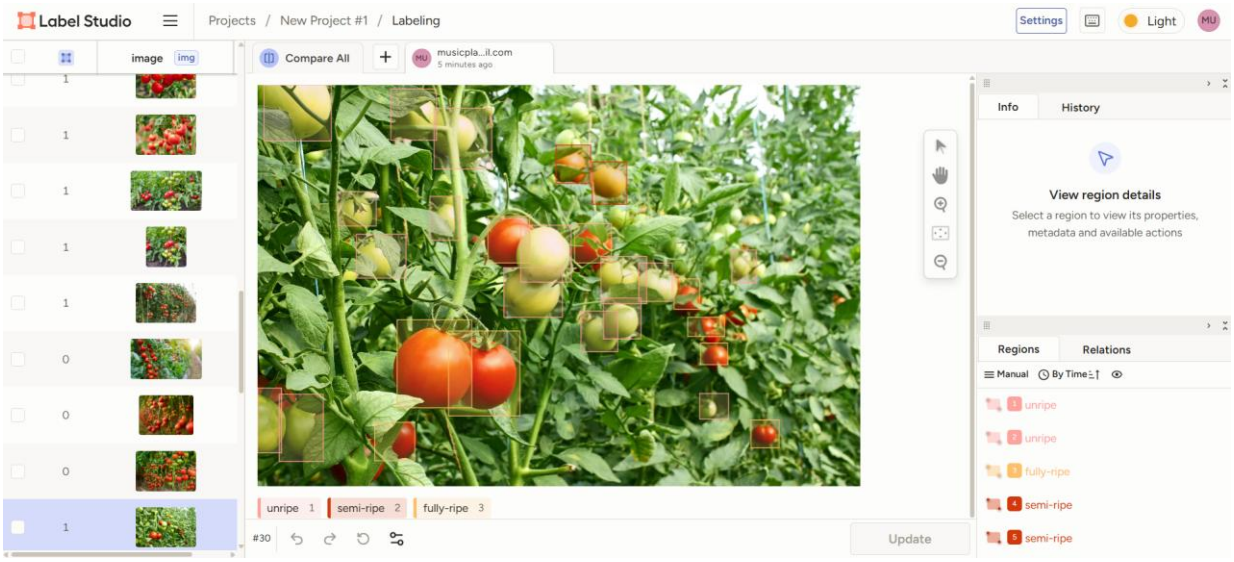


Рисунок 3.10 – Процес розмітки класів об’єктів

Після експорту YOLO-файлу з Label Studio, повторюємо попередні процедури з розділення на тренувальні та валідаційні папки набору даних, створення YAML-файлу датасету та запуску навчання вже готової моделі best.pt на новому датасеті (рис. 3.11).

Також, для подальшого навчання було використано хмарне середовище Google Colab, яке добре підходить для виконання Python–коду та надає безкоштовний доступ до потужної відеокарти Nvidia Tesla T4, що має 16 гігабайт відеопам’яті, та активно використовуєть в датацентрах і машинному навчанні.

Це дозволило суттєво скоротити час навчання моделі та допомагає швидше знаходити та виправляти недоліки датасету.

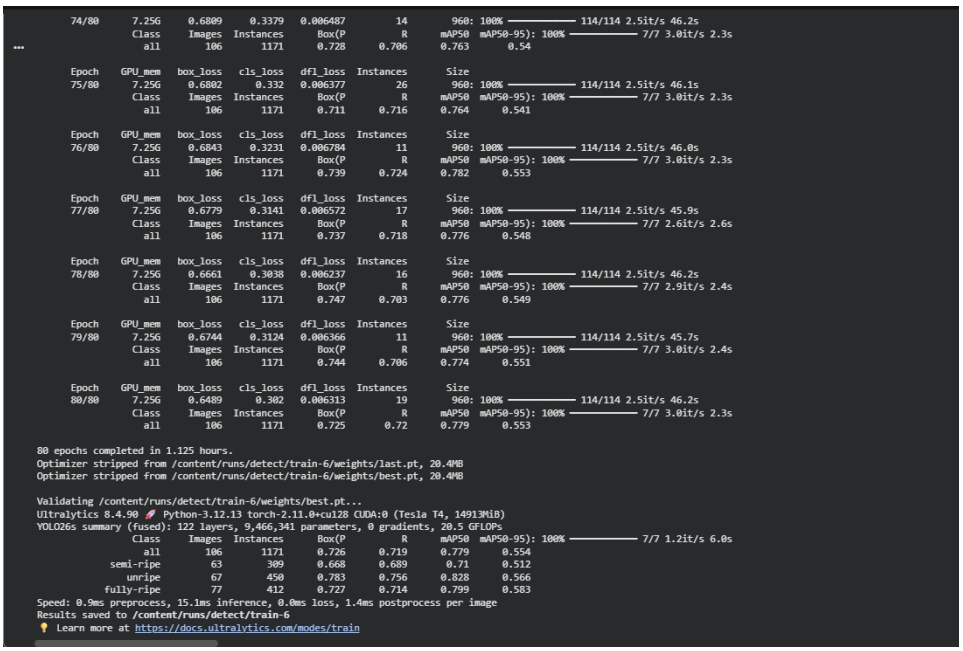


Рисунок 3.11 – Процес донавчання моделі в Google Colab

Оскільки датасет великий, було встановлено параметри epochs = 80, patience = 25, imgsz = 960 (для кращого виявлення дрібних об'єктів) та lr0 = 0,001 щоб модель не "забувала" попередньо вивчені ознаки.

Для наочності покращення результатів, тестування донавчаної моделі було проведено на тих самих фото (рис. 3.12), відео (рис. 3.13) та фото показаним на usb-камеру (рис. 3.14).

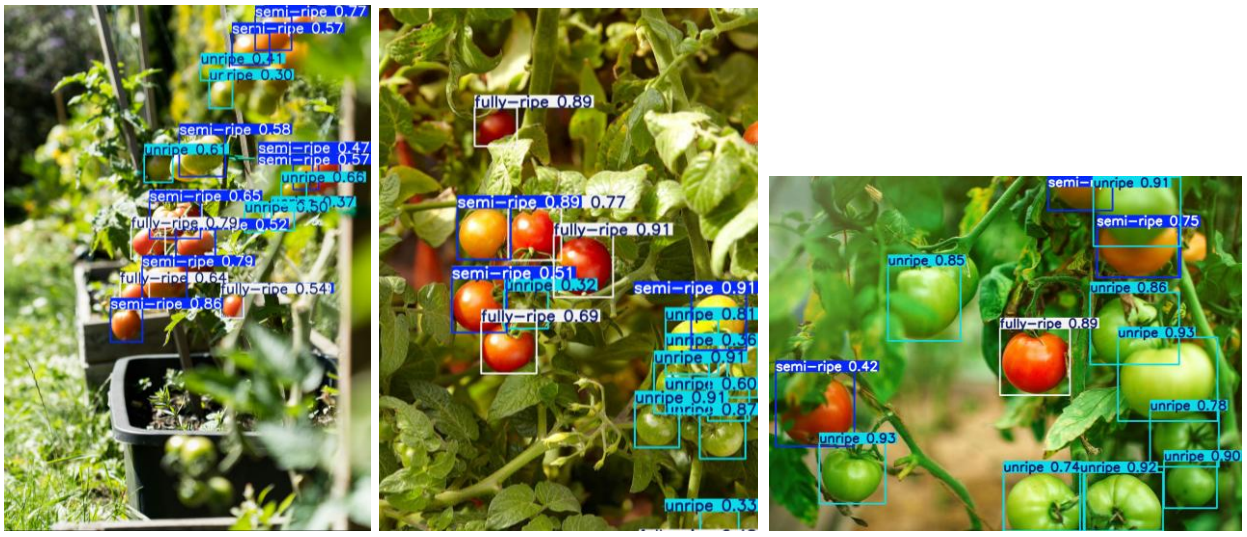


Рисунок 3.12 – Приклад роботи донавчаної моделі розпізнавання на фото

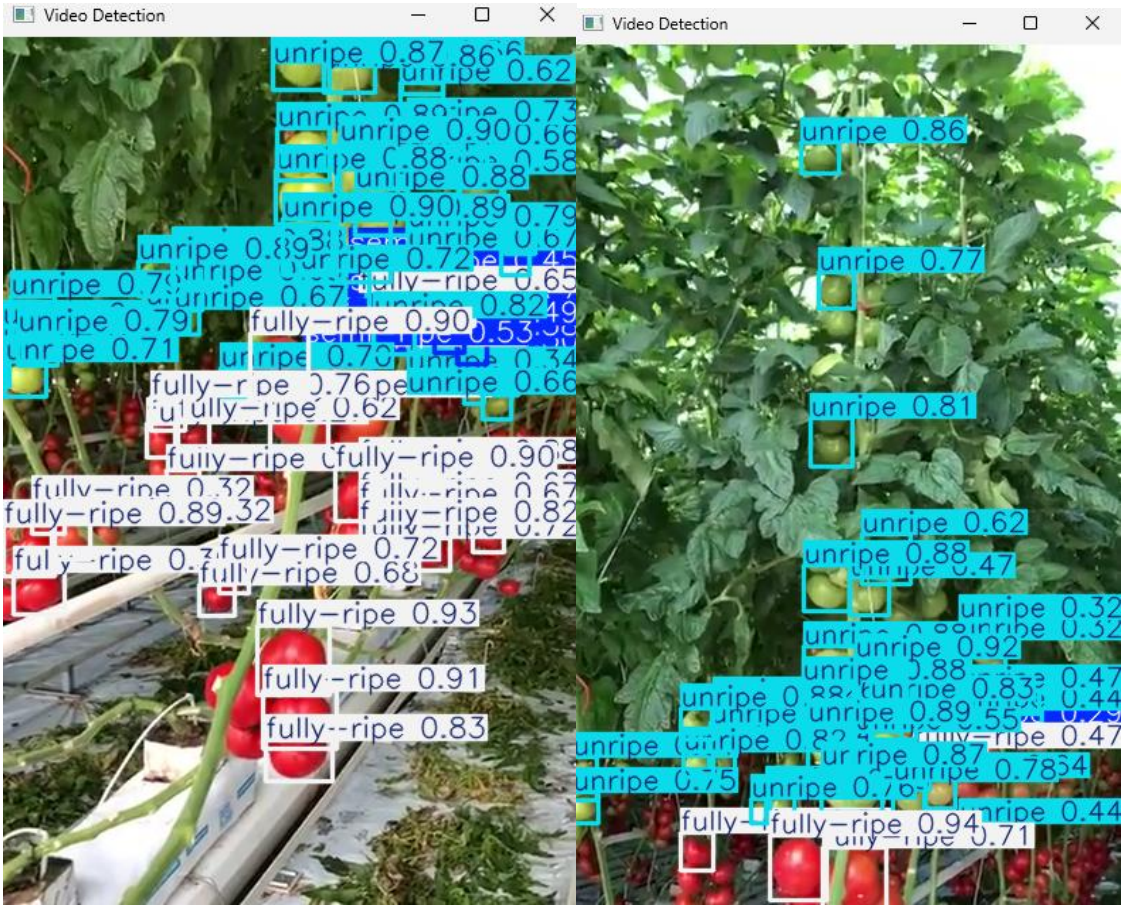


Рисунок 3.13 – Приклад роботи донавчаної моделі розпізнавання на відео

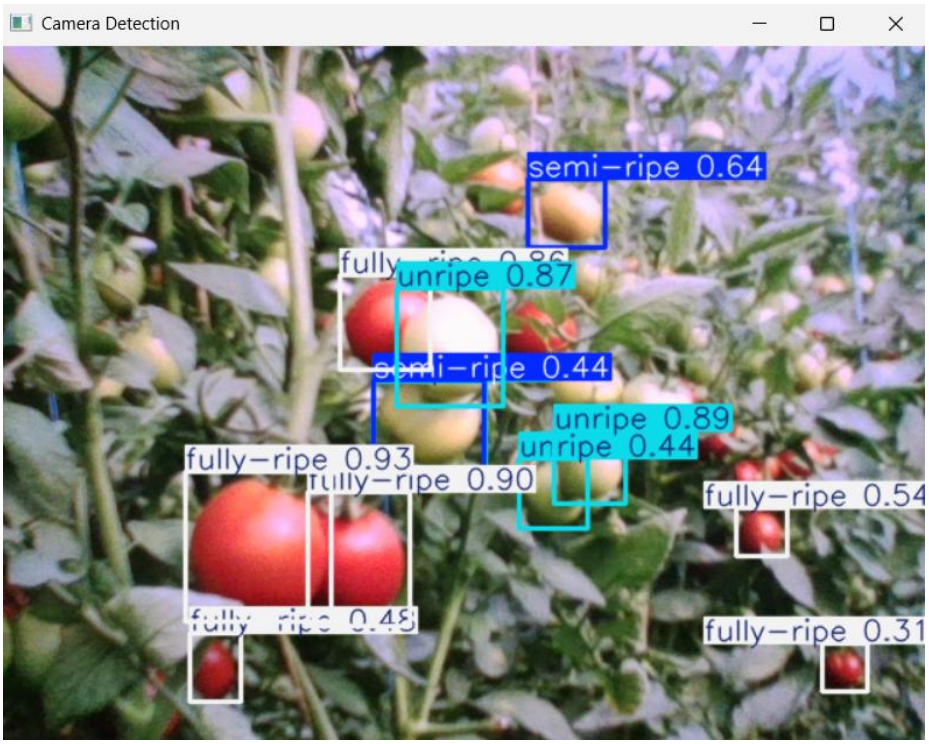


Рисунок 3.14 – Приклад роботи донавчаної моделі розпізнавання з фото на usb-камері

За результатами донавчання моделі на розширеному наборі даних вдалося покращити якість розпізнавання помідорів, зокрема у випадках, коли плоди добре проглядаються та не перекриваються листям. Додавання нових прикладів дозволило підвищити узагальнювальну здатність моделі та забезпечити більш стабільну роботу в різних умовах зйомки. У подальшому точність системи може бути підвищена шляхом розширення набору навчальних даних, включення більшої кількості зображень із різними умовами освітлення та ракурсами, а також проведення додаткового донавчання моделі на нових даних.

Крім того, за наявності більш потужного апаратного забезпечення можливим є використання моделей сімейства YOLO26 більшого розміру, що потенційно дозволить підвищити точність розпізнавання. У межах даної роботи було обрано модель YOLOv26s, оскільки її продуктивності достатньо для виявлення невеликої кількості плодів, що знаходяться поблизу мобільної платформи та підлягають подальшому збору.

3.7 Розрахунок стійкості програмного модуля

Надійність роботи програмного забезпечення є однією з основних вимог до систем технічного зору, що функціонують у режимі реального часу. Під час роботи мобільної платформи програмний модуль повинен безперервно обробляти відеопотік, виконувати детекцію плодів томатів та визначати ступінь їх стиглості незалежно від короткочасних зовнішніх впливів. Тому одним із показників працездатності системи є її стійкість.

Для оцінювання стійкості використано алгебраїчний критерій Гурвіца, який є одним із рекомендованих методів дослідження стійкості систем автоматичного управління.

Для дослідження розглянемо модель системи, яка описується передатною функцією (3.1):

$$W(s) = \frac{5}{s^3 + 6s^2 + 11s + 6}. \quad (3.1)$$

Знаменник передатної функції утворює характеристичне рівняння (3.2):

$$D(s) = s^3 + 6s^2 + 11s + 6 = 0. \quad (3.2)$$

Коефіцієнти характеристичного рівняння наведено в таблиці 3.1.

Таблиця 3.1 – Коефіцієнти характеристичного рівняння

Коефіцієнт	Значення
a_0	1
a_1	6
a_2	11
a_3	6

Відповідно до критерію Гурвіца формується матриця (3.3):

$$H = \begin{pmatrix} 6 & 6 & 0 \\ 1 & 11 & 0 \\ 0 & 6 & 6 \end{pmatrix} \quad (3.3)$$

Для підтвердження стійкості необхідно, щоб усі головні визначники матриці були додатними.

Перший визначник (3.4):

$$\Delta_1 = 6. \quad (3.4)$$

Оскільки $\Delta_1 > 0$, перша умова критерію виконується.

Другий визначник (3.5):

$$\Delta_2 = \begin{vmatrix} 6 & 6 \\ 1 & 11 \end{vmatrix} = 6 \cdot 11 - 1 \cdot 6 = 60. \quad (3.5)$$

Отже, $\Delta_2 > 0$.

Третій визначник (3.6):

$$\Delta_3 = \begin{vmatrix} 6 & 6 & 0 \\ 1 & 11 & 0 \\ 0 & 6 & 6 \end{vmatrix} = 360. \quad (3.6)$$

Таким чином, $\Delta_3 > 0$.

Оскільки всі визначники Гурвіца додатні, то система є стійкою.

Проведений розрахунок підтверджує, що досліджувана система після виникнення зовнішніх збурень повертається до усталеного режиму роботи без втрати працездатності. Для розробленого програмного модуля це означає стабільне функціонування алгоритмів обробки зображень та виявлення плодів томатів під час роботи мобільної платформи. Отриманий результат свідчить про можливість використання розробленого програмного забезпечення як складової системи технічного зору роботизованої платформи.

3.8 Безпека праці

Під час розроблення, навчання та тестування програмного забезпечення системи технічного зору основним робочим місцем є персональний комп'ютер. Тому особлива увага повинна приділятися створенню безпечних та комфортних умов праці, які сприяють збереженню працездатності користувача та зменшенню негативного впливу виробничих факторів [37].

Робоче місце повинно бути обладнане відповідно до вимог ергономіки. Монітор необхідно розташовувати на відстані 50-70 см від очей, при цьому верхня межа екрана повинна знаходитися приблизно на рівні очей або трохи нижче. Спина та руки повинні знаходитись в природньому положенні для уникнення надмірного навантаження на суглоби та перевтоми.

Важливим фактором є забезпечення достатнього рівня освітленості робочого місця. Рекомендовано використовувати комбіноване природне та штучне освітлення, яке не створює відблисків на поверхні монітора. Недостатнє або надмірне освітлення спричиняє швидку втому очей та зниження концентрації уваги.

Особливу увагу слід приділяти електробезпеці. Комп'ютерне обладнання повинно бути справним, підключеним до мережі через справні кабелі та розетки із захисним заземленням. Забороняється виконувати будь-які роботи з підключення або ремонту обладнання під напругою, а також використовувати пошкоджені електричні кабелі чи несправні блоки живлення [37].

Під час навчання нейромережевої моделі значне навантаження припадає на центральний та графічний процесори комп'ютера, що супроводжується виділенням тепла. Для забезпечення стабільної роботи обладнання необхідно підтримувати ефективне охолодження ноутбука, не перекривати вентиляційні отвори та періодично очищати систему охолодження від пилу.

Дотримання наведених вимог безпеки праці забезпечує комфортні умови роботи, знижує ризик виникнення професійної втоми та сприяє надійній експлуатації комп'ютерного обладнання під час розроблення і тестування програмного забезпечення [37].

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено програмне забезпечення системи технічного зору мобільної платформи для автоматичного виявлення та класифікації плодів томатів за ступенем стиглості.

Під час виконання роботи було проведено аналіз сучасних методів детекції об’єктів, розглянуто класичні алгоритми, двоступеневі та одноступеневі моделі, а також сучасні transformer-архітектури. На основі проведеного аналізу для реалізації системи було обрано модель YOLOv26, яка забезпечує оптимальне поєднання швидкодії та точності при роботі на пристроях з обмеженими обчислювальними ресурсами.

У процесі розробки було сформовано власний набір даних із розміткою зображень томатів трьох категорій стиглості, виконано навчання нейромережевої моделі та проведено її подальше донавчання на розширеному наборі даних. Це дозволило покращити якість розпізнавання об'єктів та підвищити стабільність роботи моделі за різних умов зйомки.

Було реалізовано програмне забезпечення, що забезпечує обробку зображень, відеофайлів та відеопотоку з камери в режимі реального часу, виконує детекцію плодів, визначає їхній ступінь стиглості та відображає результати розпізнавання у вигляді обмежувальних рамок і текстових позначень.

Результати тестування підтвердили працездатність розробленої системи та її придатність для використання у складі мобільної роботизованої платформи. Запропоноване рішення може бути використане для автоматизованого моніторингу насаджень, оцінювання ступеня стиглості плодів та як складова систем автономного збору врожаю.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Мартишко, В. М., Гуменюк, Ю. О., Деркач, О. П., Герасимчук, Г. А. (2026). Механізація і автоматизація садівництва: шлях до сталого та ефективного виробництва. Наукові нотатки, (85), 100-110.

2. Мітков, В. Б. (2025). Використання роботизованих систем у рослинництві: аналіз практик прецизійного землеробства скандинавських країн. Науковий вісник Таврійського державного агротехнологічного університету, 15(2), 83–92.

3. Дідур, Г. І., Шевченко, А. А., Чорний, М. О., Мотрук, С. О., & Мокрінчук, О. С. (2025). Управління розвитком ресурсного потенціалу аграрних підприємств в умовах впливу глобальних змін навколишнього середовища. Актуальні питання економічних наук, (17).

4. Указ Президента України No722/2019 «Про Цілі сталого розвитку України на період до 2030 року». – Режим доступу: https://www.president.gov.ua/documents/7222019-29825.

5. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. Структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. - 29 с.

6. 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. Токарєва, С.П. Новоселов, О.В Сичова. Харків: ХНУРЕ, 2023. 64 с.

7. Проектування мобільних маніпуляційних роботів : монографія / І. Ш. Невлюдов, А. О. Андрусевич, В. В. Євсєєв та ін. ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків : ХНУРЕ, 2022. – 427 с.

8. Nevliudov I., Yevsieiev V., Maksymova S., Demska N., Kolesnyk K.,

Miliutina O. Mobile Robot Navigation System Based on Ultrasonic Sensors, 2023 IEEE XXVIII International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED), Tbilisi, Georgia, 2023, pp. 247-251.

9. Козачко, О., Сигута, А., Войтік, А. (2025). Математичне моделювання процесів машинного навчання в завданнях розпізнавання зображень. Вісник Херсонського національного технічного університету, 2(4 (95)), 76-84.

10. Жлалі, Ж. Т., Джола, Є. В., & Манойленко, О. П. (2025, November). Машинний зір: як формувалося комп'ютерне «сприйняття світу». In Інноватика в освіті, науці та бізнесі: виклики та можливості. Київський національний університет технологій та дизайну.

11. Невлюдов І. Ш. Технічне та програмне забезпечення розробки малогабаритного мобільного робота : монографія / І. Ш. Невлюдов, В. В. Євсєєв, Д. В. Гурін // Кривий Ріг: Криворізький фаховий коледж Державного некомерційного підприємства «Державний університет «Київський авіаційний інститут», 2025. – 355 с.

12. Riccardo Bertoglioю, Veronica Carini, Stefano Arrigoni, Matteo Matteucci. A Map-Free LiDAR-Based System for Autonomous Navigation inVineyards. URL: https://www.researchgate.net/publication/372163272_A_Map-Free_LiDAR-Based_System_for_Autonomous_Navigation_in_Vineyards (date of access: 13.05.2026).

13. Kadeghe G. Fue, Wesley M. Porter, Edward M. Barnes, Glen C. Rains. An Extensive Review of Mobile Agricultural Robotics for Field Operations: Focus on Cotton Harvesting. URL: https://www.mdpi.com/2624-7402/2/1/10 (date of access: 13.05.2026).

14. Mahalingam, T., Subramoniam, M. (2021). A robust single and multiple moving object detection, tracking and classification. Applied computing and informatics, 17(1), 2-18.

15. Tverdomed, V., Dmytro, Z., Kokriatska, N., & Lukoševičius, V. (2024).

The Detection of Railheads: An Innovative Direct Image Processing Method. Sustainability, 16(12), 5109.

16. Krig, S. (2025). Local feature descriptors. In Computer Vision Metrics: Survey, Taxonomy, and Analysis of Computer Vision, Visual Neuroscience, and Visual AI (pp. 141-193). Singapore: Springer Nature Singapore.

17. M. Alghassab (2026). Mathematical Modeling for the Accurate Prediction and Classification of Renewable Energy Outputs via Ensemble Learning," in IEEE Access, vol. 14, pp. 17206-17220.

18. Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., Parmar, M. (2024). A review of convolutional neural networks in computer vision. Artificial Intelligence Review, 57(4), 1.

19. Li, Z., Wang, Y., Zhang, N., Zhang, Y., Zhao, Z., Xu, D. (2022). Deep learning-based object detection techniques for remote sensing images: A survey. Remote Sensing, 14(10), 2385.

20. Du, K., Bobkov, A. (2023). An overview of object detection and tracking algorithms. Engineering Proceedings, 33(1), 22.

21. Mohanty, A., Mohapatra, A. G., Mohanty, S. K. (2025). Real-time traffic monitoring with AI in smart cities. In Internet of Vehicles and Computer Vision Solutions for Smart City Transformations (pp. 135-165). Cham: Springer Nature Switzerland.

22. Safari, Y., Nakatumba-Nabende, J., Nakasi, R., Nakibuule, R. (2024). A review on automated detection and assessment of fruit damage using machine learning. IEEE access, 12, 21358-21381.

23. Liu, J., Liu, Z. (2024). The vision-based target recognition, localization, and control for harvesting robots: A review. International Journal of Precision Engineering and Manufacturing, 25(2), 409-428.

24. Colhon, M., Stoian, G., Dinu, C. C., Popîrlan, C., Tudorache, C. M., Tudor, I. V., Dănciulescu, D. (2023, September). Sliding Window Method for Imputing Missing Data. In 2023 22nd RoEduNet Conference: Networking in Education and Research (RoEduNet) (pp. 1-6).