Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

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

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Автоматизована система сортування запчастин на підприємстві на базі Ардуіно з використанням Azure Custom API
(тема)

Виконав:
студент 2 курсу, групи АУТПм-21-1

Андрейко Є.Р
(прізвище, ініціали)

Спеціальності 151 Автоматизація та комп`ютерно-інтегровані технології
(код і повна назва спеціальності)

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

Освітня програма Автоматизоване управління технологічними процесами
(повна назва освітньої програми)

Керівник проф. Цимбал О.М.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри КІТАМ _____ Невлюдов І. Ш.
(підпис) (прізвище, ініціали)

2022 р.

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

20.12. 2022 Є.Р. Андрейко

Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій

Кафедра комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки

Рівень вищої освіти другий (магістерський)

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

Тип програми освітньо-професійна

Освітня програма Автоматизоване управління технологічними процесами
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«___»_____2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студентові Андрейко Євгену Руслановича
(прізвище, ім'я, по батькові)

1. Тема роботи: Автоматизована система сортування запчастин на підприємстві на базі Ардуіно з використанням Azure Custom API затверджена наказом по університету від 07.11.2022р. № 1463Ст

2. Термін подання студентом роботи: .12.2022 р.

3.Вихідні дані до роботи: Модель автоматизованої системи сортування за допомогою реалізації технічного зору. Реалізація моделі у вигляді макета та коду.

4 Перелік питань, що потрібно опрацювати в роботі:

4.1 Аналіз предметної області

4.2 Огляд технологій та методів систем сортування

4.3 Огляд методів детекції об'єктів

4.4 Аналіз та вибір контролера

4.5 Розробка макета

4.6 Розробка програмного забезпечення для отримання та обробки зображення

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютернихілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Демонстраційний матеріал у вигляді презентації (*.pptx) – 15 с.

6. Консультанти розділів роботи (п. 6 включається до завдання за наявності консультантівзгідно 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 технічного завдання	2.11.2022	виконано
2	Аналіз літератури за темою кваліфікаційної роботи	12.11.2022	виконано
3	Огляд технологій та методів систем сортування	14.11.2022	виконано
4	Огляд методів детекції об'єктів	16.11.2022	виконано
5	Аналіз та вибір контролера	25.11.2022	виконано
6	Розробка макета	02.12.2022	виконано
7	Розробка програмного забезпечення для отримання та обробки зображення	03.12.2022	виконано
10	Подання роботи на перевірку	.12.2022	виконано
11	Оформлення пояснювальної записки	.12.2022	виконано
12	Подання роботи на рецензію	.12.2022	виконано
13	Подання на підпис роботи зав. кафедри	.11.2022	виконано
14	Подання кваліфікаційної роботи в ЕК	22.11.2022	виконано

Дата видачі завдання 1 листопада 2022 р.

Студент_____Андрейко Є. Р.
(підпис)

Керівник роботи_____ проф. Цимбал О.М.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 73 с., 37 рис., 24 джерел.

СИСТЕМА СОРТУВАННЯ, АВТОМАТИЗАЦІЯ СОРТУВАННЯ, ПРОМИСЛОВІ СКЛАДИ, ПЛАТИ КЕРУВАННЯ, ТЕХНОЛОГІІ ТА МЕТОДИ ТЕХНІЧНОГО ЗОРУ,

Об’єкт дослідження – процес керування мобільною роботизованою платформою.

Предмет дослідження – застосування комп’ютерного зору для сортування запчастин на автоматизованому виробництві.

Методи дослідження – системний аналіз, дослідження операцій, , методи і засоби автоматизованого сортування, машинного зору та об’єднання їх обох, аналіз та проектування систем зору, проектування і програмування систем сортування.

В процесі виконання кваліфікаційної роботи потрібно вирішити теоретичні та практичні задачі:

– аналіз складських систем промислових підприємств;

– аналіз існуючих складських систем сортування звичайних та автоматизованих;

– аналіз методів та технологій для реалізації технічного зору.

– розробка макета для реалізації автоматичного сортування;

– розробка та проєктування скрипта для отримання зображення з камер;

– розробка програми для обробки зображення;

– використання Azure для навчання та реалізації зору.

Робота виконана згідно [1–5].

ABSTRACT

The explanatory note: 73 pages, 37 figures, 24 sources of information.

SORTING SYSTEM, AUTOMATION OF SORTING, INDUSTRIAL WAREHOUSES, CONTROL BOARDS, TECHNOLOGIES AND METHODS OF TECHNICAL VISION,

The object of the research is sorting automation tools, methods and technologies for implementing automated machine vision.

The subject of research is the use of computer vision for sorting spare parts in automated production.

Research methods – system analysis, operations research, methods and means of automated sorting, machine vision and combining them both, analysis and design of vision systems, design and programming of sorting systems.

In the course of the qualification work, theoretical and practical tasks must be solved:

– analysis of warehouse systems of industrial enterprises;
– analysis of existing conventional and automated warehouse sorting systems;
– analysis of methods and technologies for the implementation of technical vision.
– development of a layout for the implementation of automatic sorting;
– development and design of a script for receiving images from cameras;
– development of a program for image processing;
– using Azure for vision training and implementation.
– the work was performed according to [1–5].

ЗМІСТ

ВСТУП

Автоматизація управління підприємств сприяє його розвитку, в першу чергу як введення інноваційних технологій, що в свою чергу впливає на швидкість та якість виконання роботі, що безумовно приведе до підвищення економічних показників підприємства.

Актуальність впровадження автоматизованої системи сортування за допомогою машинного зору важлива інновація яку завдяки принципам проєктування та програмування, а на сам перед абстракції , можливо використовувати для великої кількості потреб, наприклад для сортування будь який деталей з метою відділення однієї групи від іншої або за для пошуку дефектів у однієї групи і тд.

Мета роботи – вдосконалення ефективності процесів складської логістики підприємства шляхом розробки автоматизованої системи сортування будь яких запчастин за допомогою.

Одна з труднощів, пов’язаних з сучасним використанням робототехніки на складі, полягає в нездатності пересуваються по підлозі машин зчитувати інформацію з етикеток товарів, розташованих на верхніх ярусах стелажів.

Рік від року можливості складської робототехніки збільшуються, що дозволяє новим зразкам завойовувати симпатії власників складських господарств, що дивляться на речі консервативно і звикли використовувати традиційну спецтехніку, керовану оператором. На сьогоднішній час комп’ютерний зір досить розвинена галузь інформаційних технологій, але напрацювання з цієї галузі рідко використовуються у промислових маштабах. Тому актуальним є дослідження питань ідентифікації мобільним об'єктом свого положення в просторі, а також комп'ютерна емуляція навколишнього середовища.

Об’єктом дослідження є мобільна платформа, яка у її промисловій версії може бути використана на виробництвах різного типу і виконувати різноманітні операції.

Предмет дослідження – розробка вдосконаленого програмного забезпечення.

Мета атестаційною роботи – розробити апаратне забезпечення мобільного робота, як частини транспортно-складської системи, та дослідити системи орієнтування на площі, за допомогою сенсорів або інших компонентів.

Для досягнення мети потрібно розв’язати наступні завдання:

- ознайомитися з технологіями машинного зору;
- ознайомитися з сенсорами для реалізації зору;
- проаналізувати технічну літературу по темі проєкта;
- проаналізувати систем технічного зору в робототехніці;
- розглянути варіанти технічної модернізації платформи;
- розробити програмне забезпечення для мобільної платформи

1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Аналіз технічного завдання

Використання зорової інформації в системах керування роботів привертає найбільшу увагу розробників та дослідників, за для того, щоб провести аналогію з зором людини, особливо з боку розв'язання завдань розпізнавання та ідентифікації. Проте завдання інтеграції засобів комп'ютерного та технічного зору у системи керування, візуального розпізнання та обробки інформації про предмети з якими потрібно взаємод»іяти, адаптації роботи засобів керування на основі візуальної інформації є до сих пір такими, що не отримали свого повного розв'язання. Очевидним є протиріччя між сучасним рівнем розробки технологій і методів комп`ютерного зору та рівнем їх впровадження у системи інтелектуального керування мобільними та маніпуляційними роботами. Таким чином, тематика дослідження, спрямованого на розробку моделей та методів розпізнання, обробки та керування мобільними автоматизованими системами, є своєчасною та актуальною.

В даній роботі розглянемо методи детекції об'єктів на зображенні.

Системи зору тепер вважаються невід'ємною частиною багатьох промислових процесів, тому що вони можуть пропонувати швидкі, точно відтворені можливості контролю. Наприклад, у харчовій промисловості, де система технічного зору відіграє вирішальну роль у процесах, коли швидкість і точність надзвичайно важливі і допомагають забезпечити конкурентну перевагу для виробників. Сам харчовий продукт перевіряється на предмет контролю і якості порцій, а також на якість упаковки і маркування. Крім того, на фармацевтичному ринку потрібні найвимогливіші системи бачення, які не тільки перевіряють продукти, але також перевіряють використання та налаштування систем, забезпечуючи правильне дозування і контроль за процесами виготовлення ліків [6]. Бачення приносить користь багатьох секторах виробництва, де робототехніка є однією з головних областей застосування. Одним з ключових моментів є той факт,

що її можна розглядати як технологію включення в систему управління роботом. Роботи гарні в повторюваних завданнях, але погано враховують швидко мінливі параметри, тому, коли розташування продукту змінюється, робот– система не спрацьовує. Система технічного зору дозволяє роботам «бачити» об’єкт і обчислити його X – і Y– позиції. Останнім часом роботи стали застосовуватися з можливістю дво – і трьох– бачення. Таким чином, ним стала доступна і третя координата, як правило, висота об’єкта. Список систем датчиків зображень, програмних пакетів і діапазон інтелектуальних камер постійно зростає, тому для будь– якого додатка існує технічне зір у системах управління роботами. З появою недорогих багатоядерних процесорів система розширила свої горизонти (рис. 1.1).

Рисунок 1.1 – Абстрактний приклад системи та системі оцінок

Двоетапні методи (two-stage methods), вони ж методи, що базуються на регіонах (region-based methods) – підхід, розділений на два етапи. На першому етапі селективним пошуком або за допомогою спеціального шару нейронної мережі виокремлюються регіони інтересу (regions of interest, RoI) - області, які з високою вірогідністю містять усередині себе об'єкти. На другому етапі обрані регіони розглядаються класифікатором для визначення приналежності до вихідних класів і регрессором, що уточнює місце розташування обмежувальних рамок.

Одноетапні методи (one-stage methods) – підхід, що не використовує окремий

алгоритм для генерації регіонів, натомість передбачаючи координати певної кількості обмежувальних рамок із різними характеристиками, такими, як результати класифікації та ступінь упевненості, а надалі коригуючи місце розташування рамок [6].

1.2 Аналіз методів детекції об’єктів

Відстеження положення є поєднанням апаратних засобів і програмного забезпечення, яке дозволяє визначити абсолютне положення об'єкта в просторі. Ця технологія є критично важливою для досягнення ефекту занурення у віртуальну реальність. У поєднанні з відстеженням орієнтації стає можливим вимірювати та передавати всі 6 ступенів свободи (6-DoF) реального світу. У ході роботи з технологіями віртуальної реальності в нашій компанії ми отримали певний досвід у цьому питанні і хотіли б їм поділитися, розповівши про існуючі способи відстеження положення для віртуальної реальності, а також плюси і мінуси того чи іншого рішення. Невелика класифікація, сукупність методів і підходів до розв'язання цієї задачі можна розділити на кілька груп:

Акустичні – акустичні прилади стеження використовують ультразвукові (високочастотні) звукові хвилі для вимірювання положення та орієнтації цільового об'єкта. Для визначення положення об'єкта вимірюється час прольоту (time-of-arrival) звукової хвилі від передавача до приймачів або різниця фаз синусоїдальної звукової хвилі при прийомо-передачі. Компанія Intersense розробляє пристрої відстеження позиції на основі ультразвуку. Акустичні трекери зазвичай мають низьку швидкість оновлення, викликану низькою швидкістю звуку в повітрі. Інша проблема полягає в тому, що швидкість звуку в повітрі залежить від таких факторів зовнішнього середовища, як температура, барометричний тиск та вологість

Радіочастотні – методів заснованих на радіочастотах безліч. Багато в чому за принципами визначення становища вони схожі на акустичні методи відстеження (відмінність лише в природі хвилі). Найбільш перспективними на даний момент є методи UWB (Ultra-Wide Band), але навіть у кращих рішеннях на основі UWB

точність досягає лише порядку сантиметрів (DW1000 від DecaWave, Dart від Zebra Technologies, Series 7000 від Ubisense та інші);

Магнітні – магнітний трекінг заснований на вимірюванні інтенсивності магнітного поля у різних напрямках. Як правило, у таких системах є базова станція, яка генерує змінне або постійне магнітне поле. Оскільки сила магнітного поля зменшується зі збільшенням відстані між точкою вимірювання та базовою станцією, можна визначити місце розташування контролера. Якщо точка вимірювання обертається, розподіл магнітного поля змінюється різними осях, що дозволяє визначити орієнтацію. Точність даного методу може бути достатньо високою в контрольованих умовах, однак магнітне відстеження піддається перешкодам від струмопровідних матеріалів поблизу випромінювача або датчика, від магнітних полів, створюваних іншими електронними пристроями і феромагнітними матеріалами в просторі відстеження;

Оптичні – це методи являють собою сукупність алгоритмів комп'ютерного зору і пристроїв, що відстежують, в ролі яких виступають камери видимого або інфрачервоного діапазону, стерео-камери і камери глибини. Залежно від вибору системи відліку виділяють два підходи для відстеження положення (рис 1.2).

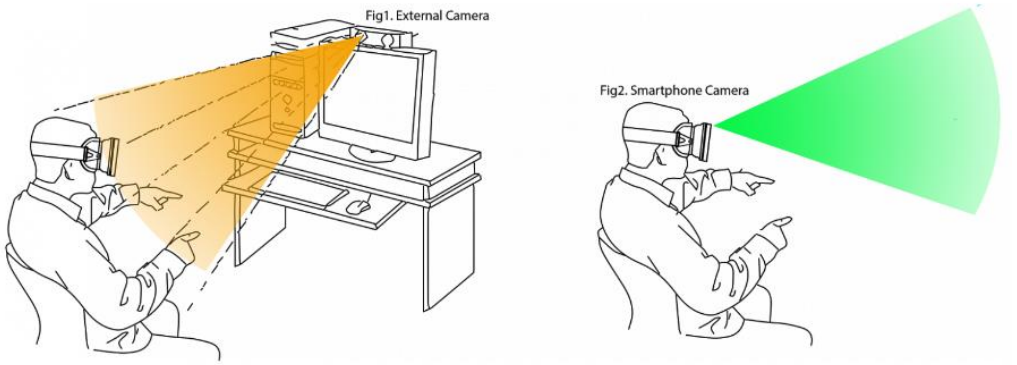

Рисунок 1.2 – Два підходи оптичного метода [7]

Outside-in підхід має на увазі присутність нерухомого зовнішнього спостерігача (камера), що визначає положення об'єкта, що рухається, за характерними точками. Використовується в Oculus Rift (Constrellation), PSVR, OSVR та безлічі Motion Capture систем. Inside-out підхід передбачає наявність на

об'єкті оптичного сенсора, що рухається, завдяки якому можливо відстежувати рух щодо нерухомих точок в навколишньому просторі. Використовується Microsoft Hololens, Project Tango (SLAM), SteamVR Lighthouse (гібридний варіант, тому є базові станції).

Завдання Perspective-n-Point (PnP), при оптичному відстеженні визначення положення об'єкта у просторі вирішується так звана завдання PnP (Perspective-n-Point) (рис1.3), коли за перспективною проекції об'єкта на площину сенсора камери необхідно визначити положення об'єкта в 3D-просторі.

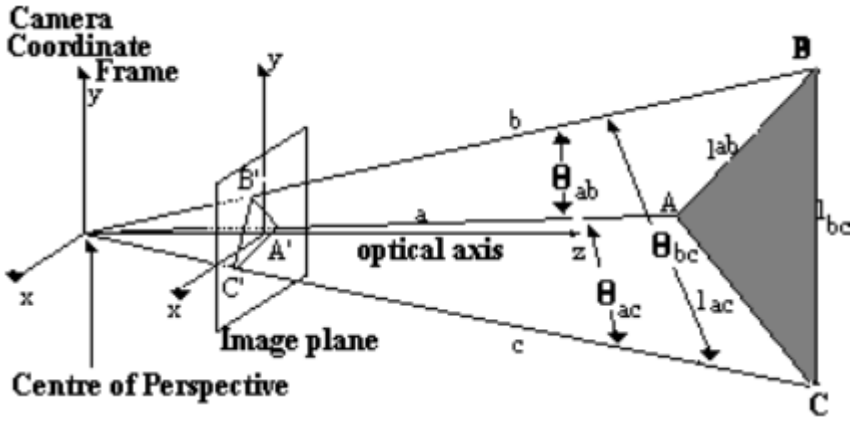

Рисунок 1.3 – PnP завдання [7]

Для заданої 3D-моделі об'єкта та 2D-проекції об'єкта на площину камери вирішується система рівнянь. Внаслідок чого виходить безліч можливих рішень. Кількість рішень залежить від кількості точок у 3D-моделі об'єкта. Однозначне рішення для визначення 6-DoF положення об'єкта можна отримати як мінімум при 4 точках.

Для трикутника виходить від 2 до 4 можливих рішень (рисунок 1.4), тобто положення не може бути однозначно визначено.

Рішення пропонується досить великою кількістю алгоритмів, реалізованих у вигляді бібліотек (POSIT, posest, OpenCV (solvePnP)).

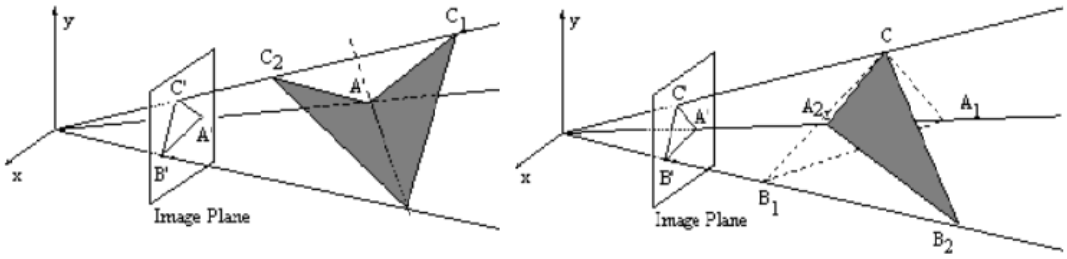

Рисунок 1.4 – Можливі рішення для трикутника [7]

SLAM – Simultaneous Localization and Mapping (рисунок 1.5), метод одночасної локалізації та побудови карти (SLAM) – це найбільш популярний спосіб позиціонування в робототехніці (і не тільки), який застосовується для відстеження положення у просторі.

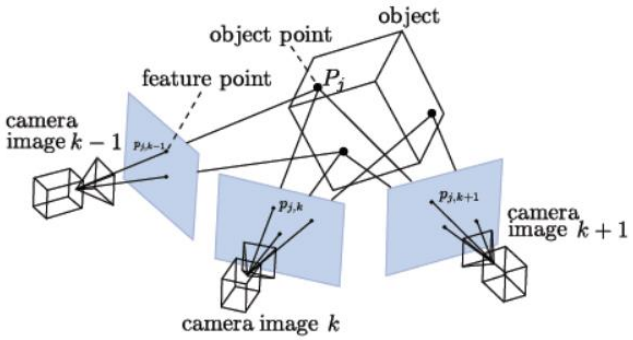

Рисунок 1.5 – Приклад SLAM метода [7]

Алгоритм складається з двох частин: перша – складання карти невідомого навколишнього простору на основі вимірювань (дані з одометра або стерео-камери), друга – визначення свого місця розташування (локалізація) у просторі на основі порівняння поточних вимірювань з наявною картою простору. Цей цикл безперервно перераховується, при цьому результати одного процесу беруть участь у обчисленнях іншого процесу. Найбільш популярні методи вирішення задачі включають фільтр частинок і розширений фільтр Калмана [7].

Насправді SLAM – це досить широка тема, а не лише один певний алгоритм, і розбір усіх існуючих рішень на цю тему тягне на окрему статтю.

SLAM зручний для мобільних рішень віртуальної та доповненої реальності.

Також залежно від наявності спеціальних оптичних маркерів виділяють окремо:

– безмаркерний трекінг, як правило, будується на складних алгоритмах з використанням двох і більше камер, або стерео камер із сенсорами глибини;

– трекінг із використанням маркерів передбачає заздалегідь задану модель об'єкта, яку можна відстежувати навіть із однією камерою. Маркерами зазвичай є джерела інфрачервоного випромінювання (як активні, так і пасивні), а також видимі маркери на кшталт QR-кодів. Такий вид трекінгу можливий лише в межах прямої видимості маркера;

– інерційні (рис 1.6) – сучасні інерційні вимірювальні системи (IMU) на основі MEMS-технології дозволяють відстежувати орієнтацію (roll, pitch, yaw) у просторі з великою точністю та мінімальними затримками [7].

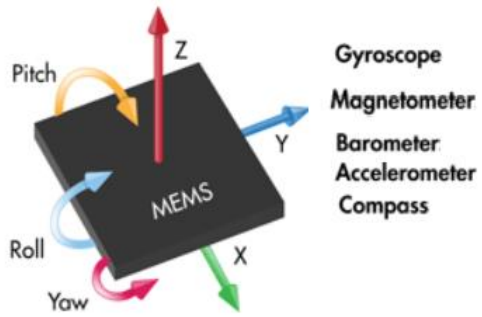

Рисунок 1.6 – Інерційна вимірювальна технологія [7]

Завдяки алгоритмам «sensor fusion» на основі комплементарного фільтра або фільтра Калмана (ефективний рекурсивний фільтр, що оцінює вектор стану динамічної системи, використовуючи ряд неповних та зашумлених вимірів) дані з гіроскопа та акселерометра успішно коригують один одного та забезпечують точність як для короткочасних вимірювань, так і для тривалого періоду.

Проте визначення координат (переміщення) рахунок подвійного інтегрування лінійного прискорення (dead reckoning), обчисленого з сирих даних з акселерометра, не задовольняє вимогам точності на тривалих періодах часу. Акселерометр сам по собі дає дуже зашумлені дані, і при інтегруванні помилка

збільшується з часом квадратично. Вирішити цю проблему допомагає комбінування інерційного підходу до трекінгу з іншими методами, які періодично коригують так званий дрифт акселерометра.

– гібридні – так як жоден з методів не є бездоганним, і всі вони мають свої слабкі місця, найрозумніше комбінувати різні методи відстеження. Так інерційний трекінг (IMU) може забезпечити високу частоту поновлення даних (до 1000 Гц), тоді як оптичні методи можуть дати стабільну точність у тривалі періоди часу (коригування дрифту). Гібридні методи відстеження засновані на алгоритмах Sensor Fusion, найбільш популярним є розширений фільтр Калмана (EKF – Extended Kalman Filter) [8].

Людське сприйняття висуває високі вимоги до точності (~1мм) та затримок (< 20 мс) в обладнанні. Оптичні та інерційні методи найбільш близькі до цих вимог, і найчастіше використовуються спільно, доповнюючи один одного. Розглянемо базові принципи, у яких побудовані перелічені вище методи.

Двоетапні методи:

– Region-CNN (R-CNN, Region-based Convolutional Network) – алгоритм, заснований на згорткових нейронних мережах (рис 1.7). Замість того, щоб використовувати для пошуку зображень ковзаючі вікна фіксованого розміру, на першому кроці алгоритм намагається знайти селективним пошуком регіони - прямокутні рамки різних розмірів, які, ймовірно, містять об'єкт. Це забезпечує більш швидке та ефективне знаходження об'єктів незалежно від розміру об'єкта, відстані до камери, кута зору. Сумарна кількість регіонів для кожного зображення, згенерованих на першому кроці, приблизно дорівнює двом тисячам. Знайдені регіони за допомогою афінних перетворень набувають розміру, який потрібно подати на вхід CNN. Також замість афінних перетворень можна використовувати паддинги, або розширювати обмежувальні рамки до розмірів, необхідних для входу CNN. Як CNN найчастіше використовують архітектуру CaffeNet, що витягує для кожного регіону близько 4096 ознак.

На останньому етапі вектори ознак регіонів обробляють SVM, що проводять класифікацію об'єктів, по одній SVM на кожен домен [9].

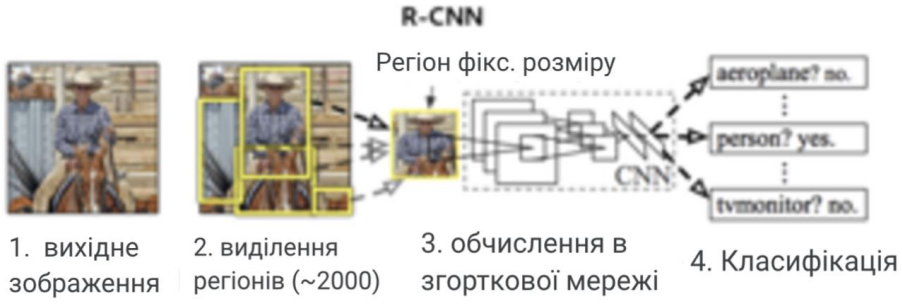

Рисунок 1.7 – Схема роботи алгоритму R-CNN [9]

Селективний пошук, своєю чергою, теж можна навчати за допомогою лінійної регресії параметрів регіону – ширини, висоти, центру. Цей метод, названий bounding-box regression, дає змогу точніше виділити об'єкт. Як дані для регресії використовуються ознаки, отримані в результаті роботи CNN.

– Fast R-CNN (рис 1.8). За рахунок того, що в R-CNN для кожного з 2000 регіонів класифікація проводиться окремо, навчання мережі займає великий обсяг часу. Оригінальній версії алгоритму R-CNN для опрацювання кожного тестового зображення було потрібно близько 47 секунд, тому його автори запропонували алгоритм, що поліпшує продуктивність – Fast R-CNN. Його характерною особливістю є подача на вхід CNN не окремих регіонів, а всього зображення відразу для отримання загальної карти ознак. Запропоновані регіони накладаються на загальну карту ознак, і в результаті кількість операцій згортання істотно зменшується. Оскільки регіони мають різний розмір, необхідно привести ознаки до фіксованого розміру за допомогою операції RoIPooling (Region of interest pooling). У рамках RoIPooling регіон ділиться на сітку, розмірність осередків якої збігається з розмірністю виходу, після чого за осередками сітки проводиться вибір максимального значення. Отримані регіони фіксованого розміру далі є входом для повнозв'язного шару, який і здійснює як класифікацію, так і лінійну регресію для

зсуву меж його рамок. Варто зазначити, що у Fast R-CNN використовується спільне навчання SVM для класифікації, CNN і bounding box регресора замість незалежного їхнього навчання – для цього використовується спільна функція втрат [9].

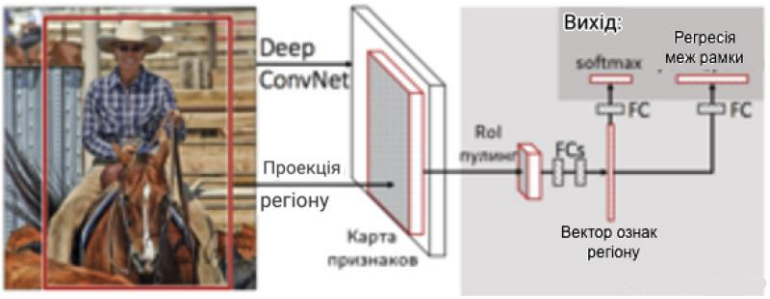

Рисунок 1.8 – Схема роботи алгоритму Fast R-CNN [9]

Одноетапні методи. Сімейство алгоритмів R-CNN використовує передбачення регіонів, що дає змогу забезпечувати хорошу точність, але може бути дуже повільним для деяких сфер, таких, як безпілотне керування автомобілем. Можна виділити ще одне сімейство алгоритмів для детекції зображень, що паралельно розвиваються і не використовують регіони - сімейство алгоритмів швидкої детекції.

SSD – (Single Shot Detector) (рис 1.9) використовує ідею використання пірамідальної ієрархії виходів згорткової мережі для ефективного виявлення об'єктів різних розмірів. Зображення послідовно передається на шари згорткової мережі, які зменшуються в розмірах. Вихід з останнього шару кожної розмірності бере участь в ухваленні рішення щодо детекції об'єктів, таким чином, складається пірамідальна характеристика зображення. Це дає змогу виявляти об'єкти різних масштабів, оскільки розмірність виходів перших шарів сильно корелює з обмежувальними рамками для маленьких об'єктів, а останніх - для великих. На відміну від YOLO, SSD не розбиває зображення на сітку довільного розміру, а передбачає зміщення ключових рамок. Ключові рамки на різних рівнях масштабуються так, що одна розмірність вихідного шару відповідає за об'єкти свого масштабу. У результаті, великі об'єкти можуть бути виявлені тільки на

вищому рівні, а маленькі об'єкти - на низьких рівнях. Як і в інших алгоритмах, функція втрат забезпечує спільний внесок як втрат локалізації, так і втрат класифікації [10].

Рисунок 1.9 – Архітектура нейронної мережі для алгоритму SSD [9].

1.3 Аналіз систем сортування на складі

Жодне велике промислове підприємство чи виробництво не обходиться без сортувальних конвеєрів. Це незамінне автоматизоване обладнання, без якого складно уявити копіткий процес сортування чи комплектації. Оптимізація робочих процесів за рахунок сортувальної конвеєрної системи очевидна, адже можна швидко та чітко розподілити товар чи комплектуючі як за категоріями, так і за місцем призначення для зберігання чи транспортування. Завдяки конвеєру зручність сортування відчувається з першої хвилини роботи. Підвищити продуктивність і забезпечити рентабельність логістичних операцій можна завдяки використанню сортувальних конвеєрів. Значне скорочення ручної роботи та зниження тимчасових витрат відбувається за рахунок автоматизованого сортування товарів, організованого під час руху конвеєра. По суті, сортувальний конвеєр є одним із ключових етапів процесу сортування [10 – 11].

На стадії завантаження лінії, щоб використання обладнання було ефективним, необхідно визначити правильність вибору продукції/товару, періодичність подачі тощо. А ось сама періодичність подачі на сортування товару зазвичай регулюється швидкістю руху конвеєрної стрічки. За допомогою

сканування штрих-коду на упаковці відбувається ідентифікація товару/вантажу, це важливий момент, адже сканер зчитує всю інформацію про вантаж для передачі її сортувальній системі, а вже потім товар йде із загального потоку конвеєра за запрограмованим маршрутом в точку збору, відвантаження, тощо.

Так звані збираючи конвеєри задіяні на етапі збору/комплектації, як правило, представлені кількома лініями, що подають. Потоки та рівномірність надходження вантажу координуються системою збору, незалежно від типу конвеєра (наприклад, роликовий або стрічковий), що дозволяє сортувальній машині працювати безперервно [11].

Етап відвантаження є завершальним, і тут необхідний контроль надходження товару, який своєчасно має видобуватись на виході. Тут зазвичай використовують роликові (гравітаційні) або стрічкові конвеєри як відвантажувальне обладнання. Механізмів для сортування існує досить багато, відрізняються складністю та швидкодією, якщо можна так сказати. Розглянемо деякі види сортування, наприклад, сортування високої швидкості, середньої і так звана низько-швидкісне сортування.

Так от, якщо говорити про низько-швидкісну, то це використовується найпростіше сортувальне обладнання в комплексі зі звичайними роликовими або стрічковими конвеєрами, швидкість сортування яких досягає до 30 од./хв. Використання в таких лініях трансферу сприяє передачі (зштовхуванню) вантажу/товару убік, на інший конвеєр чи жолоб. Тут необхідно враховувати зональність конвеєрної лінії для товару, щоб уникнути зіткнень і збоїв у пересуванні. Як правило, один трансфер може застосовуватись для сортування за двома напрямками. Якщо говорити про сортувальні машини з середньою швидкістю від 30 до 150 од/хв, то серед них дуже популярні дивертори – пристрої з робочими механізмами, що піднімаються, що забезпечує точну орієнтацію вантажу при його зміщенні. Це відбувається так: під дном короба, наприклад, що під'їжджає до місця знімання, піднімаються ролики, що дозволяє підняти його і при великій швидкості переміщення перенаправити у бік скидання під кутом приблизно 30-45º у напрямку руху. Швидкість сортування варіюється від 60 до 150 од./хв. За

допомогою такого обладнання можна сортувати різні вантажі, у тому числі і тендітні. Високошвидкісні сортувальники можуть обробити вже від 150 до 400 од/хв, з цим не впораються звичайні конвеєри, тут підключається спеціалізоване обладнання. Вантаж на такі системи можна класти як вручну, і автоматизовано з допомогою конвеєра [11].

Найбільш поширені з них конвеєри: з лотками, що нахиляються, конвеєр-знімач поперечний стрічковий, з лотками які розсуваються, що зручно для штучних одиничних товарів/вантажів. Ще спеціально для великогабаритних вантажів (ящиків, коробок) існує сортувальник з кулачками, що відхиляють. Така машина може розподіляти товар з одного на кілька конвеєрів та сортує 200-300 од/хв, але залежно від специфіки вантажу та встановлених проміжків між ним.

Під час вибору системи основну увагу слід приділити ефективності сортування. Слід обрати систему, яка може забезпечити найкращу продуктивність:

– прискорити обробку широкого спектра товарів з різними характеристиками (розмір, вага, крихкість);

– скоротити ручну працю, пов'язану із сортуванням продукції;

– уникнення помилок під час комплектації багаторядкових замовлень клієнтів [11].

Сортування може бути ручним, напівавтоматичним або автоматичним (IN і OUT). Автоматичне сортування – найкраще рішення при високій пропускній здатності деталей.

Ручне сортування менш продуктивне і більш трудомістке. Однак його можна підтримувати невеликими елементами автоматизації, наприклад, системою Pick-by-Light. Технологія без паперового комплектування спрощує і прискорює процес комплектування багаторядкових замовлень клієнтів. Система Pick-by-Light представлена у вигляді дисплея з елементами управління та світлодіодним дисплеєм, який активується перед відповідним товаром. Технологія дає змогу обробляти від 250 до 600 рядків на годину. Використання багатоколірного дисплея дає змогу кільком користувачам одночасно обробляти багаторядкове замовлення клієнта або одному технічному спеціалісту одночасно обробляти кілька завдань.

Завдяки впровадженню інноваційної технології Pick-by-Light продуктивність складу подвоюється. Робочі процеси виконуються швидко, а ризик неправильного введення зводиться до нуля. У зв'язку зі збільшенням робочого навантаження і пов'язаним із цим ризиком помилок автоматизація процесів стає все більш важливою. За допомогою конвеєрів партії товарів можна розділити на різні категорії. Обладнання працює з різними видами матеріалу: коробки, ящики, піддони, посилки, конверти, гнучкі пакувальні матеріали. Для сортування за допомогою конвеєрів використовуються різні види обладнання: стрічкові конвеєри, роликові конвеєри з приводом і без приводу, конвеєри спіральні та вертикальні. Кожен тип обладнання використовується для виконання певних завдань. Використання автоматичних систем сортування push-tray дає змогу звести до мінімуму кількість співробітників у процесі комплектування, завдяки чому процес комплектування прискорюється, а ризик помилок знижується. Можливості системи та її конфігурація легко адаптуються до умов, що змінюються [11].

1.4 Висновок до розділу 1

В першому розділі було проаналізовано технічне завдання, та розглянута потрібна література, а саме про системи сортування, автоматизація складських систем, та тільки поверхнево розглянуто про методи та технології реалізації технічного зору. Для отримання відповідних знань для проєктування та реалізації обраного проєкту.

2 ВИБІР ТА ОБГРУНТУВАННЯ КОНТРОЛЕРУ ТА МЕТОДУ РЕАЛІЗІЦІЇ ТЕХНІЧНОГО ЗОРУ

2.1 Технології для визначення місцезнаходження об’єктів у просторі

Рішення для визначення місцезнаходження – це збірний термін, що відноситься до технологій, які використовуються для відстеження положення цільового об'єкта (ресурсу або особи) в реальному або найближчому часі, зазвичай, в межах обмеженої ділянки. Технології рішень для визначення місцезнаходження використовуються спільно з системами позиціонування для збору даних та надання оперативної інформації та контексту, що дозволяє компаніям удосконалювати бізнес-рішення на основі зібраних даних про місцезнаходження.

Концепція рішення для визначення місцезнаходження існує вже з тих пір, як люди вперше захотіли відстежувати положення об'єктів, людей і товарів, і сягає часу появи технології радіолокації. Рішення для визначення місцезнаходження почали розвиватися, коли на торговій виставці ID EXPO в 1998 був введений термін RTLS (система позиціонування в реальному часі). Цей термін був створений для опису та виділення нової технології, яка не тільки забезпечувала можливості автоматичної ідентифікації за допомогою активних RFID-міток, але й додавала можливість перегляду розташування на екрані комп'ютера. Саме на цій виставці було показано перші приклади комерційних систем RTLS на основі радіолокації. Хоча ці можливості раніше використовувалися військовими та урядовими установами, ця технологія була надто дорогою для застосування в комерційних цілях. На початку 1990-х років у США були встановлені перші комерційні системи RTLS, які працювали на основі передавальних міток. З того часу з'явилося більш просунуте обладнання для визначення місцезнаходження, а також розвиваються дані та програмне забезпечення для визначення місцезнаходження. Лідери галузі в даний час розробляють методи прогнозної аналітики та впроваджують робочі алгоритми, щоб допомогти компаніям ще більш підвищити продуктивність [12].

Інтернет речей (IoT) являє собою мережу фізичних пристроїв, оснащених датчиками та можливостями зв'язку, що дозволяють об'єктам встановлювати з'єднання та обмінюватися даними без прямої взаємодії з людиною. Типовий сценарій використання Інтернету речей включає програми для смартфона, здатні контролювати термостати в будинку. Можливість легко відслідковувати розташування за допомогою датчиків стала важливою точкою даних для багатьох пристроїв IoT. Дані про місцезнаходження можна використовувати для управління активами в будь-якій кількості організацій шляхом підключення до центральної платформи IOT, що забезпечує новий рівень доступності даних та керованості, а також сприяє більш зваженому прийняттю рішень. У контексті IoT та зростаючої взаємопов'язаності додаткове використання RTLS створює новий вимір раніше недоступних даних. Кінцеві користувачі з'ясують, що немає універсальних рішень визначення місцезнаходження. Наприклад, деяким користувачам потрібно точно відстежувати ресурс у час у межах великого складу, тоді як іншим, мабуть, потрібно знати приблизне місце розташування ресурсу, наприклад, у якому ряду перебуває продукт у гіпермаркеті. Важливість місцезнаходження в реальному часі може залежати від цінності ресурсу або фізичного етапу процесу, на якому знаходиться об'єкт. У деяких ситуаціях користувачеві потрібно знати лише останню контрольну точку, яку минув об'єкт. У багатьох випадках для досягнення результатів виконання процесу чи робочого циклу потрібні кілька технологій. [12]

Для автоматичного позиціонування доступні різні технології. Тип обраної технології зазвичай залежить від двох основних факторів:

– точність розташування: деякі технології можуть відстежувати переміщення об'єктів аж до воріт або зони. Це корисно для тих об'єктів, переміщення яких потрібно відстежувати у простих випадках, наприклад, при надходженні палети на склад або відправлення зі складу. Інші технології можуть відстежувати розташування об'єкта з великою точністю (до метра або навіть менше). Це корисно в тих ситуаціях, коли потрібне точне визначення місцезнаходження;

– швидкість поновлення. Отримання інформації в режимі реального часу

не потрібне для всіх рішень. Наприклад, якщо палету було просканово під час проходження через ворота складу і тепер знаходиться стелаж, постійне оновлення розташування не потрібне. Однак інші рішення можуть надавати оновлені дані про місцезнаходження щогодини або в режимі реального часу, наприклад, у межах хвилин, секунд або менше. Прикладом може бути рішення, що вимагає постійного оновлення при відстеженні розташування транспортного засобу в міру того, як воно переміщається об'єктом [12].

Просторові характеристики, які варто враховувати:

– діапазон, який діапазон об'єктів, що відстежуються, вони розкидані по всьому світу, чи знаходяться на об'єкті, у приміщенні;

– точність, залежить від наявності перешкод, а також часу/кута/сигналу;

– точність, як точно потрібно визначити місце розташування об'єкта (7 м або 25 см);

– частота поновлення інформації, як часто потрібно оновлювати інформацію про місцезнаходження (кожну секунду, кожні 5 хвилин або вручну);

– інфраструктура, існуюча (Wi-Fi), зовнішня (стільникова/супутникова), проста чи складна;

– спосіб застосування, традиційне прокладання кабелів встановлення приймачів або антен або використанням міток та додатків;

– час роботи від батареї, щоденна підзарядка, щорічна заміна батарей, заміна міток кожні 5–10 років;

– функціональна сумісність, для міток, зчитувача/приймача, для механізму позиціонування;

– вартість, вартість міток, інфраструктури, впровадження та обслуговування. [13]

Розглянемо технології визначення місцеположення:

– RFID – RFID розшифровується як радіочастотна ідентифікація. Це автоматична технологія ідентифікації, коли цифрові дані, закодовані в RFID – мітці або смарт-етикетці, захоплюються пристроєм зчитування за допомогою радіохвиль. Простіше кажучи, технологія RFID нагадує технологію зчитування

штрих-кодів, проте для зчитування даних з міток використовуються радіохвилі, а не оптичне сканування штрих-кодів на етикетці. Для RFID не потрібно прямої видимості мітки або етикетки для зчитування даних, що зберігаються. Це одна із ключових особливостей системи RFID. Система RFID є динамічною, тобто є можливість оновлювати дані на схемі. На відміну від RFID-мітки, штрих-код є статичним і не може бути змінений;

– підключивши зчитувач RFID до Інтернету, об'єкти з RFID-мітками або етикетками можна ідентифікувати та відстежувати автоматично;

– пасивні RFID-мітки - не мають джерела живлення та використовують антену та інтегральну схему (ІВ). Зчитувач посилає радіохвилі, які забезпечують живлення ІВ, коли вона знаходиться у зоні дії зчитувача. Такі мітки зазвичай містять лише основну ідентифікаційну інформацію, проте мають малий розмір, тривалий термін служби (понад 20 років) та низьку вартість [13];

– активні RFID-мітки – вимагають наявності в мітках власного джерела живлення (зазвичай це батарея) та передавача для трансляції сигналу на RFID-зчитувач. Вони можуть зберігати більше даних, мають збільшену дальність зчитування та є відмінним вибором для високоточних рішень, що вимагають відстеження в реальному часі. Вони мають великі розміри через вбудовану батарею і зазвичай є більш дорогими. Приймачі зчитують односторонні передачі від активних міток [13];

– надширокосмугові технології (СШП) – такі мітки можуть забезпечити високий рівень точності місцезнаходження на середній відстані за рахунок великої кількості передач для визначення точного розташування в режимі реального часу. Вони забезпечують базову ідентифікацію, а також надають дані про місцезнаходження та інші датчики. Приймачі зчитують односторонні передачі з міток, додають до них позначку часу та перенаправляють на концентратор. Концентратор забезпечує живлення, синхронізацію та провідне мережеве з'єднання приймачів на великих відстанях. Приклади використання включають управління обладнанням (забезпечення використання правильного інструменту на потрібному етапі операції), відстеження руху людей та орієнтацію для забезпечення безпеки,

де необхідний високий рівень точності місцезнаходження [13];

– Bluetooth з низьким енергоспоживанням – такі радіомаяки побудовані на універсальному стандарті Bluetooth і добре підходять для систем з низьким енергоспоживанням, а також доступні для читання пристроїв з підтримкою Bluetooth або смартфонами. Радіомаяки передають дані про місцезнаходження на мобільні пристрої або виділені шлюзи, а потім перенаправляють їх через мережу Wi-Fi або мобільну мережу на платформу (проміжне ПЗ), яка перетворює їх на відомості про місцезнаходження. Приклади включають ситуації, коли систему необхідно встановити без переривання бізнес-процесів (наприклад, в установі охорони здоров'я), тому використовується обладнання співробітників (стільникові телефони, мобільні комп'ютери), щоб спільно визначати місце розташування ваших активів. Ще один варіант використання радіомаяків Bluetooth з низьким енергоспоживанням: розміщення простих в установці точок оплати, які можуть використовуватися встановленими на смартфонах клієнтів програмами від продавців для підвищення рівня взаємодії з покупцями [13];

– Wi-Fi – смартфони є одним із найпоширеніших способів застосування рішень для визначення місця розташування з використанням Wi-Fi. Якщо у вас є смартфон і ви входите до будівлі, мережа Wi-Fi у цій будівлі може використовуватися для відстеження вашого розташування. У мобільних операційних системах є функції, які можуть обмежити таке відстеження та захистити вашу конфіденційність. Однак, якщо ви даєте свою згоду на відстеження (наприклад, як покупці) або використовуєте корпоративні пристрої (як працівники), визначення місця розташування через мережу Wi-Fi може стати недорогим і простим у реалізації варіантом. Оскільки на підприємствах вже існують мережі Wi-Fi, організації можуть скористатися перевагами існуючої мережі для впровадження рішення визначення місцезнаходження. Деяке обладнання, можливо, потрібно додати або перемістити для підвищення ефективності роботи такого рішення. Крім того, такий варіант має низьку точність роботи, тому може не підійти для всіх сценаріїв роботи. Хорошим прикладом застосування є відстеження відвідувачів або співробітників, коли потрібна точність

лише на рівні зон [13];

– ISO 24730 – стандартизований бездротовий протокол 2,4 ГГц існує спільно зі стандартом Wi-Fi/802.11. Зазвичай він працює краще, ніж Wi-Fi, у промислових умовах, складних для передачі РЧ-сигналів з огляду на наявність великої кількості металу, обладнання та інших перешкод, які блокують або відображають сигнал, викликаючи перешкоди. У разі прямої видимості пристрої, які працюють під керівництвом цього протоколу, можуть обмінюватися інформацією з відривом до 1 км. Цей варіант найкраще підходить для відстеження активів, на відміну від інших корпоративних ресурсів, таких як люди або матеріали. Він добре працює при відносно повільній швидкості оновлення з активами, інформацію про місцезнаходження яких потрібно оновлювати щохвилини, а не щомиті. Ця технологія ідеально підходить для стоянок або об'єктів зовнішньої установки, де необхідна точність аж до місця паркування [13];

– GPS – глобальна система позиціонування, що використовує комплекс синхронізованих за часом супутників, кожен з яких транслює сигнал на землю. Приймач GPS або мітка приймає безліч сигналів від декількох супутників, після чого порівнює різницю в часі прийому сигналів для визначення свого розташування. Найкраще підходить для відкритих просторів з прямою видимістю неба. Такий варіант працює найкраще для зовнішнього позиціонування, великих об'єктів, таких як аеропорти, а також місць, де впровадження інфраструктури становить складність. На таких великих площах пристрої GPS можуть використовувати функцію самостійного позиціонування, забезпечуючи хорошу точність при вибірковій або мінімальній установці елементів інфраструктури.

– системи позиціонування – пакет корпоративного програмного забезпечення, що надає інструменти для проектування, конфігурування, експлуатації та усунення несправностей RTLS-рішень. Ця система служить як центральне сховище всіх даних реального часу про місцезнаходження та зв'язок, отриманих інфраструктурою позиціонування RTLS. Корпоративне програмне забезпечення може забезпечувати роботу тисячі об'єктів, користувачів і сотень тисяч ресурсів, що відстежуються, а також має можливість відстежувати

положення мобільних пристроїв. Надає дані іншим корпоративним додаткам, партнерам або клієнтам, а також містить інтерфейс користувача та консоль зі звітами, подіями, оповіщеннями та діями на рівні ресурсів корпорації, а також з можливістю управління системою. З цим програмним забезпеченням вам не потрібно знати джерело даних; воно застосовує ефективні алгоритми згладжування, і його можна вбудовувати у різні додатки. Це дозволяє користувачам об'єднати окремі програми в одну центральну базу даних [13].

Існує чимало архітектури нейронних мереж, що дозволяють розпізнавати сутність. Більшість їх є дворівневі нейронні мережі, коли одна нейронна мережу поєднує у собі кілька простіших. Основний принцип роботи таких нейронних мереж полягає у поділі всього зображення на регіони чи клітини, після чого відбувається аналіз кожного регіону.

У дворівневих нейронних мереж є дві істотні проблеми: зображення аналізується не повністю, а тільки виділені регіони, а швидкість роботи таких нейронних мереж значно нижчі. Перший недолік пов'язані з принципом роботи, згаданим раніше. Найчастіше кожен регіон обробляється послідовно за допомогою згорткової нейронної мережі. Результат, отриманий згортковою нейронною мережею, передається у повнозв'язну для визначення класу та розмірів області, всередині якої знаходиться об'єкт, що шукається. Далі нейронна мережа обробляє наступний регіон і алгоритм повторюється. Однак існує нейронна мережа, яка позбавлена цих недоліків. Вибрана нейронна мережа YOLO (You Only Look Once) являє собою дворівневу нейронну мережу, переважно з згортковими шарами. Забігаючи наперед, можна відзначити, що YOLO здатна обробляти всі регіони, з яких складається зображення, за один раз. Головна причина, через яку було обрано саме цю нейронну мережу, полягає в її ефективності, оскільки вона має кращі значення щодо співвідношення між швидкістю і точністю [13].

YOLO – алгоритм YOLO (рис 2.1), запропонований 2016 року, був першою спробою зробити можливою детекцію об'єктів у реальному часі. У рамках алгоритму YOLO вихідне зображення спочатку розбивається на сітку з N×N комірок. Якщо центр об'єкта потрапляє всередину координат комірки, то ця

комірка вважається відповідальною за визначення параметрів місцезнаходження об'єкта. Кожна комірка описує кілька варіантів розташування обмежувальних рамок для одного й того самого об'єкта. Кожен із цих варіантів характеризується п'ятьма значеннями - координатами центру обмежувальної рамки, її шириною і висотою, а також ступенем упевненості в тому, що обмежувальна рамка містить у собі об'єкт. Також необхідно для кожної пари класу об'єктів і комірки визначити ймовірність того, що комірка містить у собі об'єкт цього класу. Таким чином, останній шар мережі, що ухвалює остаточне рішення про обмежувальні рамки та класифікацію об'єктів, працює з тензором розмірності N×N×(5B+C), де B - кількість передбачуваних обмежувальних рамок для осередку, C - кількість класів об'єктів, визначених від самого початку [14].

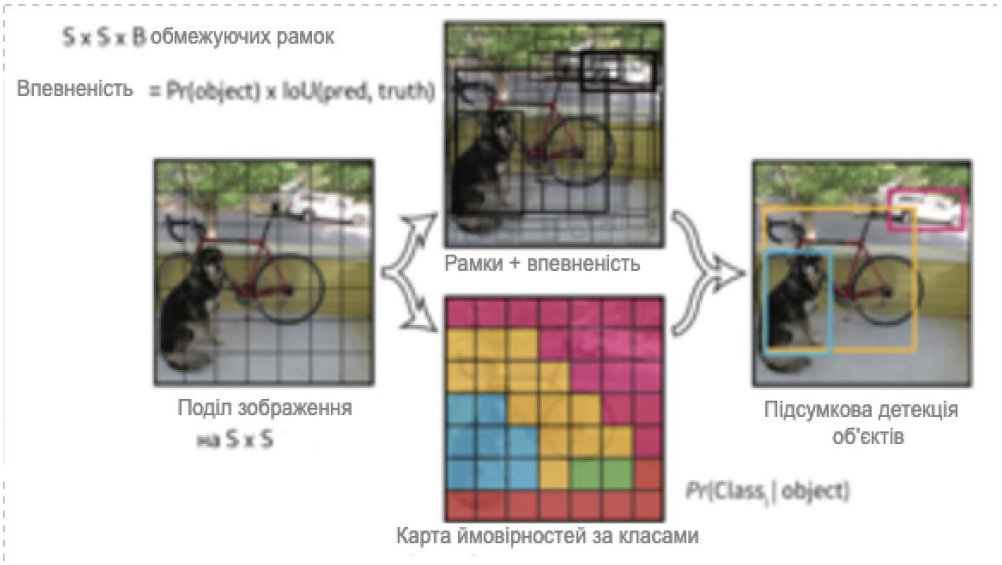

Рисунок 2.1 – Схема роботи алгоритму YOLO [14]

Нейронна мережа YOLOv4 і DL-фреймворк Darknet (C/C++/CUDA) краще за швидкістю FPS і точності AP50:95 і AP50 на датасеті Microsoft COCO, ніж DL-фреймворки та нейронні мережі: Google TensorFlow EfficientDet, FaceBook Detectron RetinaNet PyTorch Yolov3-ASFF, і багато інших. YOLOv4 досягає точності 43.5% AP / 65.7% AP50 на тесті Microsoft COCO при швидкості 62 FPS TitanV або 34 FPS RTX 2070. На відміну від інших сучасних детекторів, YOLOv4 може навчати відеокарта nVidia з 8-16 GB VRAM. Тепер не тільки великі компанії

можуть навчати нейронну мережу на сотнях GPU/TPU для використання великих розмірів mini-batch для досягнення вищої точності, тому ми повертаємо контроль над штучним інтелектом звичайним користувачам, тому що для YOLOv4 велика міні-партія не потрібна, можна обмежитися розміром 2 – 8. YOLOV4- оптимальна від використання real-time, т.к. мережа лежить на кривій оптимальності за Парето на графіку (рис 2.2) AP(accuracy)/FPS(speed) [14-15].

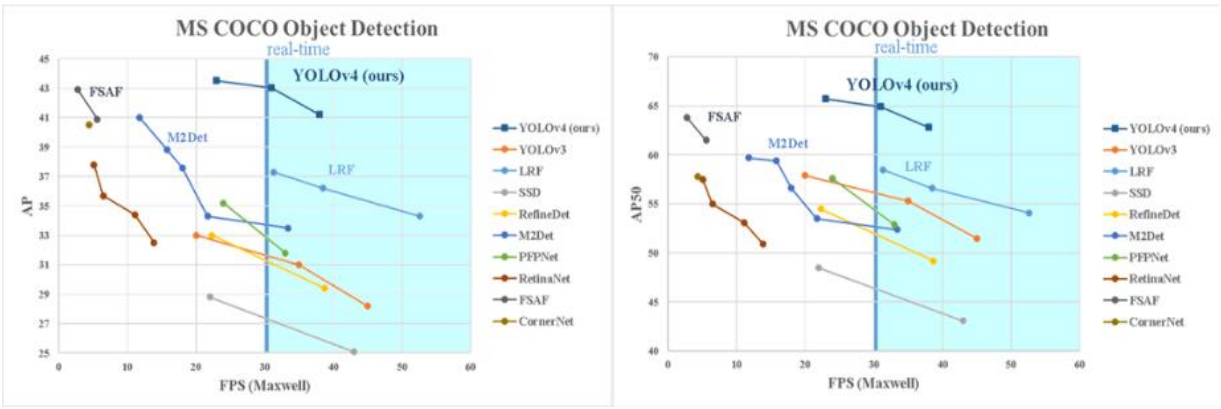

Рисунок 2.2 – Графік точності та швидкості (FPS) [14-15]

Особливості навчання різних нейронних мереж. Навчання EfficientDet з розміром mini-batch = 128 на декількох GPU Tesla V100 32GB, в той час як YOLOv4 була навчена всього на одній GPU Tesla V100 32GB з mini-batch = 8 = batch/subdivisions, і може бути навчена на звичайній ігровій -16GB GPU-VRAM.

Наступний аспект, складність навчання нейронної мережі виявлення власних об'єктів. Скільки часу ви не навчали б інші мережі на одній GPU 1080 Ti, ви не отримаєте заявленої точності, показаної на графіці вище. Більшість мереж (EfficientDet, ASFF) треба навчати на 4 – 128 GPUs (з великим розміром mini-batch використовуючи syncBN) і треба навчати щоразу для кожного дозволу мережі, без виконання обох умов неможливо досягти заявленої ними точності AP або AP50 [14-15].

2.2 Автоматизована система сортування

Автоматична система сортування – дуже корисне рішення для кожної сучасної компанії. Основними її перевагами є:

– підвищена продуктивність;

– можливість модульної конструкції.

Завдяки особливості модульної конструкції можна розробити ефективне рішення для будь-якого складу. Система сортування з висувними лотками може бути переконфігурована в міру необхідності, коли цього вимагають змінювані робочі вимоги:

– наявність систем ідентифікації продукції (ручна або автоматична) і різних типів подачі вантажу на конвеєрну систему: У разі подачі вантажу контейнери подаються прямо або під кутом 45/90 градусів. Завантаження конвеєрної системи автоматичне або ручне;

– низьке енергоспоживання: економічна робота систем - найважливіший фактор зниження експлуатаційних витрат;

– надійна цілодобова робота: система сортування з виштовхувальними лотками вирізняється простотою в експлуатації. Крім того, він не потребує спеціального і дорогого обслуговування [16].

Система сортування зі штовхачем може працювати з різними типами упаковки: пластикові коробки, картонні коробки, конверти, поліетиленові пакети та багато іншого.

Щоб правильно вибрати систему сортування для складу, необхідно враховувати кілька моментів:

– пропускна здатність;

– залежно від цієї характеристики вибирається тип сортувального обладнання [17];

– кількість різних напрямків сортування;

– години роботи;

– площа та обладнання складу;

– екологічні чинники (показник вологості, температура, шумове забруднення)[17].

Крім того, враховуються технічні властивості товарів, що сортуються: розмір і вага, центр ваги, крихкість, коефіцієнт тертя.

Метою сортування деталей є поділ запчастин на групи за однією або кількома ознаками. Коли виробнича компанія отримує товари, головне завдання - розділити їх на різні категорії та направити у відповідні зони складу. З іншого боку, в логістичній компанії товари з різних складських приміщень мають бути зібрані в багаторядкове замовлення клієнта [18].

Впровадження систем автоматизації складських операцій дає:

– економію часу під час комплектації замовлень;

– точний контроль товаропотоку;

– мінімізацію витрат на обслуговування обладнання;

– уникнення помилок під час комплектації замовлень клієнтів.

Автоматизація складських операцій підвищує продуктивність і спрощує робочі процеси [18].

2.3 Вибір та обґрунтування контролера

Контролер для мобільної платформи в даній дипломній роботі буде вибраний в залежності від датчика зору, оскільки деякі розглянуті датчики не можуть взаємодіяти з деякими процесорами контролерів. На даний момент розглядаються такі мікроконтролери як Arduino, Raspberry pi або Iskra JS.

Якщо буде обрано Arduino або Iskra JS, то у поєднанні з ними використовуватиметься troyka shield. Розглянемо деякі характеристики вище вказаних контролерів та шилда.

Iskra JS – флагманська плата із вбудованим інтерпретатором JavaScript.Вона є розвитком платформи Espruino, але сумісна з платами-шилдами Arduino. Якщо для проекту важливі швидкість та комфорт розробки, максимальна сумісність із платами розширення, сенсорами та іншими електронними модулями, Iskra JS – це

оптимальний вибір.

Iskra JS працює на частоті 168 МГц. 32-бітний мікроконтролер ARM Cortex-M4 надає у розпорядження 1 МБ флеш-пам'яті для зберігання прошивки інтерпретатора JavaScript та вашого коду, а також 192 КБ оперативної пам'яті для їхньої роботи [19].

Цього вистачає для обробки JS-коду та вирішення безлічі завдань на кшталт управління роботами, промисловою автоматикою, системами розумного будинку тощо [19].

Характеристики Iskra JS:

– мікроконтролер: STM32F405RG (32-бітний ARM Cortex M4);
– тактова частота: 168 МГц;
– флеш-пам'ять: 1024 кБ;
– SRAM-пам'ять: 192 кБ;
– номінальна робоча напруга: 3,3 В;
– вхідна напруга, що рекомендується: 7–15 В або 3,6–12 В;
– максимальний струм із шини 5V: 1000 мА;
– максимальний струм з шини 3.3V: 300 мА (включаючи живлення мікроконтролера);
– максимальний струм із піна або на пін: 25 мА;
– максимальний сумарний струм із пінів або на піни: 240 мА;
– портів введення-виведення загального призначення: 26;
– портів з підтримкою ШІМ: 22;
– портів з АЦП: 12 (12 біт);
– портів із ЦАП: 2 (12 біт);
– доступні апаратні інтерфейси: 4× UART/Serial, 3× I²C/TWI, 2× SPI;
– габарити: 69×53 мм [19].

Для визначення які варіації Arduino можна використовувати в цій роботі, розглянемо деякі характеристики шилда. Troyka Shield – це плата розширення,яка допомагає підключати велику кількість периферії на кшталт датчиків через

стандартні трипровідні шлейфи. Це дозволяє не вдаватися до паяння абоокремої макетної плати [19].

Характеристики Troyka Shield:

– сумісність: Arduino форм-фактора Uno R3, Mega 2560;
– живлення підключених модулів: 3,3–5 В;
– інтерфейси Troyka (S-V-G): 20 груп контактів;
– інтерфейс I²C: 3 групи контактів;
– інтерфейс SPI: 1 група контактів;
– габарити: 69×53×19 мм [19].

Судячи з характеристик шилда, бачимо, що найкраща взаємодія буде з платами Arduino uno R3 та Arduino, тож розглянемо їх характеристики.

Плата Arduino Uno яку зображено на рисунку 2.3 – апаратна обчислювальна платформа, основними компонентами якої є проста плата введення/виводу [20].

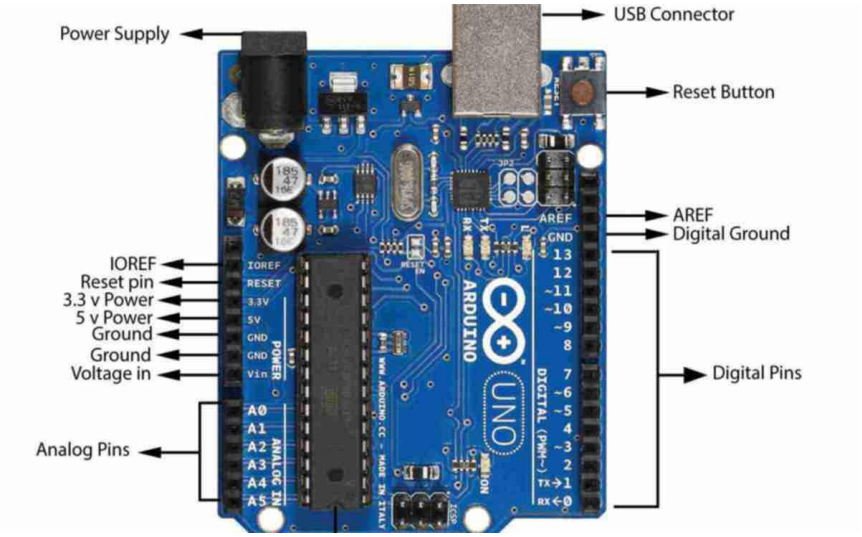

Рисунок 2.3 –Контролер Arduino UNO [20]

Плата Arduino складається з мікроконтролера Atmel AVR (ATmega328 та ATmega168 у нових версіях та ATmega8 у старих) та елементної обв'язки для програмування та інтеграції з іншими схемами. На кожній платі обов'язково присутні лінійний стабілізатор напруги 5 і 16 МГц кварцовий генератор (в деяких

версіях керамічний резонатор). У мікроконтролер попередньо прошитиме завантажувач, тому зовнішній програматор не потрібен [20].

На концептуальному рівні всі плати програмуються через RS-232 (послідовне з'єднання), але реалізація цього способу відрізняється від версії до версії. Плата Serial Arduino містить просту схему, що інвертує, для конвертування рівнів сигналів RS-232 в рівні ТТЛ, і навпаки. Поточні плати, як UNO, програмуються через USB, що здійснюється завдяки мікросхемі конвертера USB-to-serial на кшталт FTDI FT232. У деяких варіантах, таких як Arduino Mini або неофіційної Boarduino, для програмування потрібне підключення окремої плати USB-to-serial або кабелю.

Плати Arduino дозволяють використовувати більшу частину I/O висновків мікроконтролера у зовнішніх схемах. Наприклад, у платі Diecimila доступно 14 цифрових вводів/виводів (рівні «LOW» – 0В і «HIGH» – 5В), 6 з яких можуть видавати ШІМ сигнал, та 6 аналогових входів (0-5В). Ці висновки доступні у верхній частині плати через 0,1-дюймові роз'єми типу «мама». На ринку є кілька зовнішніх плат розширення, відомих як «shields» [20].

Програмне забезпечення можливо писати в інтегрованому середовище розробки Arduino – це кросплатформна програма, що включає редактор коду, компілятор і модуль передачі прошивки в плату.

Середовище розробки засноване на мові програмування Processing і спроектоване для програмування новачками, не знайомими з розробкою програмного забезпечення. Строго говорячи, це C/C++, доповнений деякими бібліотеками. Програми обробляються за допомогою препроцесора, потім компілюється за допомогою AVR-GCC [20].

Плата Uno за замовчуванням підтримує три типи пам'яті.

Flash – пам'ять об'ємом 32 кБ. Це основне сховище команд. Коли ви прошиваєте контролер своїм скетчем, він записується саме сюди. 2кБ з даногопулу пам'яті відводиться на bootloader-програму, яка займається ініціалізацієюсистеми, завантаження через USB та запуску скетчу [20].

Оперативна SRAM пам'ять об'ємом 2 кБ. Тут за замовчуванням зберігаються

змінні та об'єкти, що створюються під час роботи програми. Пам'ять ця енергозалежна, при вимкненні живлення всі дані, зрозуміло, зітруться.

Енергонезалежна пам'ять (EEPROM) об'ємом 1кБ. Тут можна зберігати дані, які не зітруться при вимиканні контролера. Але процедура запису та зчитування EEPROM вимагає використання додаткової бібліотеки, яка доступна в Arduino IDE за замовчуванням. Також слід пам'ятати про обмеження циклів перезапису, властивих технології EEPROM [20].

Піни живлення (рисунок 2.3):

– 5V – на цей пін Ардуїно подає 5 В, його можна використовувати для живлення зовнішніх пристроїв;

– 3.3V – на цей пін від внутрішнього стабілізатора подається напруга 3BGND – виведення землі;

– VIN – пін для подачі зовнішньої напруги;

– IREF – пін для інформування зовнішніх пристроїв про робочунапругу плати [20].

Піни Ардуіно використовуються для підключення зовнішніх пристроїв і можуть працювати як у режимі входу (INPUT), так і в режимі виходу (OUTPUT). До кожного входу може бути підключений вбудований резистор 20-50 кОм за допомогою команди pinMode () в режимі INPUT_PULLUP. Допустимий струм на кожному з виходів – 20 мА, не більше 40 мА у піку.

Для зручності роботи деякі піни поєднують кілька функцій:

– піни 0 та 1 – контакти UART (RX та TX відповідно) Піни з 10 по 13 – контакти SPI (SS, MOSI, MISO та SCK відповідно);

– піни A4 та A5 – контакти I2C (SDA та SCL відповідно).

Піни з номерами від 0 до 13 є цифровими. Це означає, що ви можете зчитувати та подавати на них лише два види сигналів: HIGH та LOW. За допомогою ШІМ також можна використовувати цифрові порти для керування потужністю підключених пристроїв [20].

Плата Arduino Mega 2560 призначена для створення проектів, в яких не вистачає звичайних можливостей Arduino Uno. Цей пристрій має максимальну з

усіх плат сімейства Arduino кількість пінів і розширений набір інтерфейсів. Також у Arduino Mega більше вбудованої пам'яті. Плата Mega у повній відповідності зі своєю назвою є на сьогоднішній день найбільшою за розміром та кількістю пінів контролерів Arduino. У порівнянні з нею в Uno набагато менше пінів та пам'яті. Ось список основних відмінностей [20]:

– плата Mega використовує інший мікроконтролер: ATMega 2560.Але тактова частота його дорівнює 16МГц, як і в Уно;

– у платі Mega більша кількість цифрових пінів – 54 замість 14плати Uno. І аналогових – 16/6;

– у плати Mega більше контактів, що підтримують апаратніпереривання: 6 проти 2. Більше Serial портів – 4 проти 1;

– за обсягом пам'яті Uno теж значно поступається Mega. Flash-пам'ять 32/256, SRAM – 2/8, EEPROM – 4/1 [20].

Виходячи з цього можна зробити висновок, що для великих складних проектів з програмами великого розміру і активним використанням різнихкомунікаційних портів краще вибирати Mega. Але ці плати дорожчі за Uno і займають більше місця, тому для невеликих проектів, які не використовують усі додаткові можливості Mega, цілком зійде Uno [20, 21].

Також розглянемо мікропроцесор Raspberry Pi (рис 2.4), для початку дізнаємося що це таке. Raspberry Pi це Невеликий повнофункціональний комп'ютер, який можна підключити до комп'ютерного монітора, клавіатури та миші. Він має всі якості ПК - виділений процесор, пам'ять і графічний драйвер. У нього навіть є власна операційна система під назвою Raspberry Pi OS, яка є оптимізованою версією Linux [20,21].

Raspberry Pi називають одноплатним комп’ютером розміром з кредитну карту. Насправді сама плата трохи крупніше – 85,6 мм × 56 мм × 21 мм – і не має округлених країв, до того ж деякі порти просто стирчать зовні (рис. 2.1).

Важить пристрій всього 54 г. Raspberry Pi доступний у вигляді двох різних моделей, відомих як модель A і модель B.

Відмінності полягають у можливості зниження вартості моделі A завдяки

обмеження деяких функціональних можливостей.

У таблиці представлені технічні можливості та характеристики перших моделей Raspberry Pi A та B.

В лютому 2016 року компанія Raspberry Pi в черговий раз перевершила саму себе, випустивши нову модель Raspberry Pi 3.

У цьому новітньому Raspberry Pi 3 є два гігантських оновлення.

Перше – це наступне покоління чотириядерного процесора Broadcom BCM2837 64–бітної ARMv8, яке збільшує швидкість процесора від 900 МГц на Raspberry Pi 2 до 1,2 ГГц на Raspberry Pi 3.

Другим гігантським оновленням є чіп BCM43143 WiFi, вбудований у Raspberry Pi.

Тут уже не потрібні ніякі WiFi–адаптери, цей Raspberry Pi 3 вже готовий до використання Wi–Fi.

На платі також є Bluetooth Low Energy (BLE), роблячи Raspberry Pi 3 IoT гарним рішенням (BLE підтримка також реалізована).

І нарешті, в цій моделі є модернізоване джерело живлення, що перемикається, та який зріс до 2,5 А замість 2 А, що дозволяє підключати до Вашого Raspberry Pi навіть більш потужні пристрої через USB–порти [22].

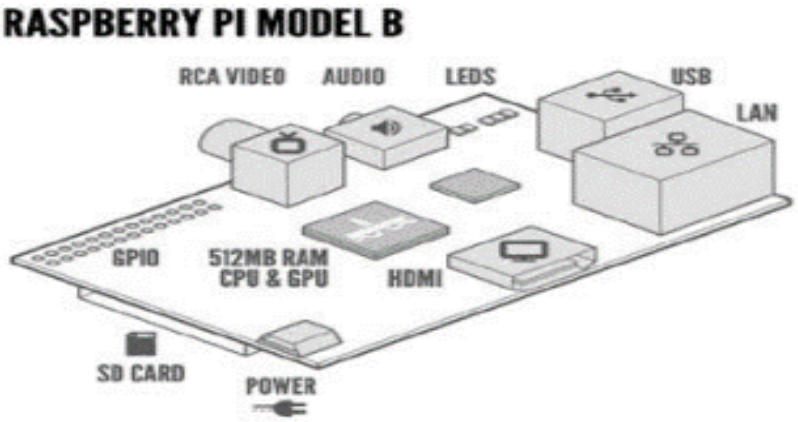

Рисунок 2.4 – Схема Raspberry Pi, модель B [22]

Починаючи з 2013 р. всі Raspberry Pi Model B повинні мати півгігабайта

RAM, але на складах перекупників могли завалятися більш ранні моделі.

Ліцензією на виробництво плат мають компанії Premier Farnell, RS Components і Egoman.

Причому остання випускає плати червоного кольору, які можуть пропонуватися тільки на китайських територіях.

До першої річниці проекту RS Components випустили ювілейну партію плат синього кольору обсягом 1000 штук.

Ці ж компанії мають право продавати Raspberry Pi, а в США поширенням займається Allied Electronics.

Тож всі інші магазини просто закуповують великі партії пристроїв у цієї четвірки і перепродують кінцевим споживачам.

Обидві моделі плат від різних виробників (складанням займаються заводи Sony, Qisda і Egoman) мають деякі несуттєві відмінності, але зрештою вони ідентичні (рис. 2.5) [22].

Рисунок 2.5 – Плата Raspberry Pi [22]

Роз’єм загального призначення введення/виведення (GPIO) – порт для отримання електричних сигналів з Raspberry Pi, таких як датчики для зчитування і управління двигунами. Він складається з двох паралельних рядів штифтів і позначений як P1. Різні моделі Raspberry Pi використовують по–різному через спосіб маршрутизації шрифтів на платі. Композитний відеовихід: цей роз’єм використовується для підключення Raspberry Pi до композитного відеоз’єднання

(стандартне телебачення) за допомогою кабелю RCA [22].

Аудіовихід – чорний 3,5 мм роз’єм на верхньому правому куті плати. Вихід HDMI – порт використовується для передачі цифрового відео на монітор комп’ютера. Вихід HDMI також може скерувати аудіо, так що можна не використовувати порт аудіовиходу [22].

USB порт(и) – дозволяють підключати USB–аксесуари (такі як клавіатура і миша і зовнішніх пристроїв зберігання даних) для системної плати. Модель А має тільки один USB–порт для зниження витрат. Модель В має два порти USB. Порт Ethernet (тільки для моделі В) – призначений для підключення Raspberry Pi до Ethernet мережі і для доступу в Інтернет.

Оперативна пам’ять – об’єм ОЗП не змінився і залишився таким, як і на попередніх моделях – 1 Гб.

Роз’єм для камери послідовного інтерфейсу (CSI) – тонкий роз’єм чорного кольору між роз’ємом Ethernet і виходом HDMI для підключення невеликої камери, наприклад, веб–камери. CSI роз’єми можна придбати в магазині Raspberry Pi.

Живлення: на лівій нижній стороні – мікроroз’єм живлення USB. Живлення забезпечується через мікроджерело живлення USB, яке підключається до цього порту. Роз’єм дисплею послідовного інтерфейсу (DSI) являє собою тонкий роз’єм для підключення високошвидкісних дисплеїв. Він використовується для підключення невеликої ЖК – панелі безпосередньо до Raspberry Pi. Можна використовувати його як базу для сенсорного введення даних.

Raspberry Pi не пропонує сховище, але можливо використовувати карти microSD для зберігання будь-якої ОС (Raspberry Pi, Ubuntu Mate і т. д.). Raspberry Pi також підтримує підключення через Bluetooth, Ethernet та Wi-Fi, тому його також можна використовувати для передачі файлів через Інтернет. Дизайн проекту Raspberry Pi та програмне забезпечення не мають відкритого вихідного коду.

Оскільки плата Raspberry Pi є не що інше, як цілий комп'ютер усередині друкованої плати, її часто називають одноплатним комп'ютером або SBC. Фонд Raspberry Pi постійно оновлюється та покращується. З моменту свого випуску він став популярним вибором для застосування у робототехніці, моніторингу погоди,

IoT та багатьох інших електронних систем [22].

Функціонал який може виконувати :

– чудова програмна реалізація;

– 64-бітний чотириядерний процесор;

– велика оперативна пам'ять (остання платня Raspberry Pi 4 Model B має до 8 ГБ оперативної пам'яті);

– швидкість процесора – 700 МГц – 1,5 ГГц;

– Raspberry Pi має 40 контактів вводу/виводу;

– його можна підключити до Інтернету;

– він може запускати всі види програм (включаючи MS Office та електронну пошту);

– він містить усі: ЦП (центральний процесор), ДП (графічний процесор), порт Ethernet, контакти GPIO (введення/виведення загального призначення) та роз'єм джерела живлення [22].

Тепер можна розглянути ключові відмінності між платами Arduino та Raspberry Pi:

– плати Arduino – це мікроконтролери (не повноцінні комп'ютери), а плати Raspberry Pi – мікропроцесори.

– Raspberry Pi має власну операційну систему, а у плат Arduino її немає. Плата Arduino працює за простими інструкціями, наданими IDE (інтегрованим середовищем розробки).

– Raspberry Pi підтримує Інтернет, плати Arduino не підтримують Інтернет.

– плати Arduino дешевші, плати Raspberry Pi трохи дорожчі.

2.4 Azure

API-інтерфейси є критично важливою частиною успіху програми, особливо коли робота йде в рамках великої складної служби хмарних обчислень, такої як Microsoft Azure. Зокрема, можливість створювати власні API-інтерфейси в Azure забезпечує таку необхідну узгодженість між різними – і, що важливіше, різноманітними – групами розробників, компонентами додатків, мовами програмування та стилями архітектури, що становлять ваше програмне середовище.

Microsoft Azure – хмарна платформа компанії Microsoft. Надає можливість розробки, виконання програм та зберігання даних на серверах, розташованих у розподілених дата-центрах.

Користування баченням Azure. Azure Custom Vision – це служба розпізнавання зображень, яка дозволяє створювати, розгортати та покращувати власні моделі ідентифікаторів зображень. Ідентифікатор зображення застосовує мітки до зображень відповідно до їх візуальних характеристик. Кожна мітка є класифікацією або об'єктом.

Виявлення об'єктів – це вид комп'ютерного зору, в якому модель навчена для виявлення присутності та розташування одного або кількох класів об'єкта на зображенні. Наприклад, системі вилучення з підтримкою штучного інтелекту в магазині продуктів може знадобитися визначення типу та розташування товарів, що купуються клієнтом (рис 2.6) [23].

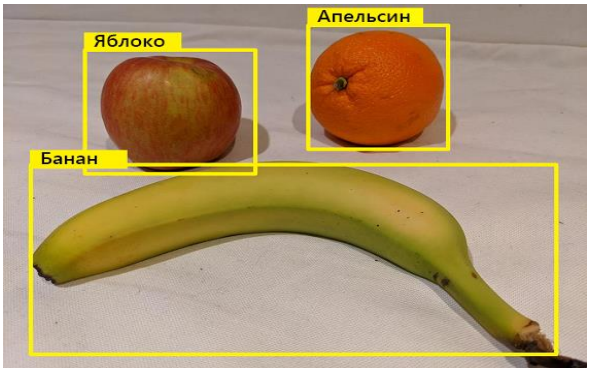

Рисунок 2.6 – Приклад розпізнання об’єктів [23]

Прогноз виявлення об'єктів складається з двох компонентів:

– Мітка класу кожного знайденого об'єкта на зображенні. Наприклад, можна з'ясувати, що на зображенні є одне яблуко і два апельсини.

– Розташування кожного об'єкта на зображенні, позначене як координати прямокутника, що обмежує, що охоплює об'єкт [23].

Щоб навчити модель виявлення об'єктів, можна використовувати когнітивну службу візуального розпізнавання користувача. Щоб використовувати службу Візуальне розпізнавання користувача, необхідно підготувати два типи ресурсів Azure:

Ресурс навчання (використовується для навчання моделей). Це може бути [23]:

– ресурс Cognitive Services;

– ресурс Користувальницьке візуальне розпізнавання (навчання);

– ресурс прогнозування, що використовується клієнтськими програмами для отримання прогнозів з моделі.

Це може бути:

– ресурс Cognitive Services;

– ресурс Користувальницьке візуальне розпізнавання (прогнозування) [23]. Також можна використовувати один ресурс Cognitive Services як для навчання, так і прогнозування. Крім того, можна комбінувати типи ресурсів (наприклад, за допомогою ресурсу Візуальне розпізнавання користувача (навчання) для навчання моделі, яка потім публікується за допомогою ресурсу Cognitive Services) [23].

Існує ряд різних методів, інструментів та методів, які можна використовувати для створення API в Azure. Деякі з них включають розширення панелі керування порталу Azure, Visual Studio Code, інтерфейс командного рядка Azure, PowerShell та шаблон Azure Resource Manager. Щоб навчити модель виявлення об'єктів, можна використовувати портал служби Візуальне розпізнавання користувача для надсилання та маркування зображень перед навчанням, оцінкою, тестуванням та публікацією моделі. Крім того, можна використовувати REST API або пакет SDK для написання коду, який виконує завдання навчання.

Найбільша різниця між навчанням моделі класифікації зображень та навчанням моделі виявлення об'єктів – це маркування зображень з тегами. Хоча для класифікації зображень потрібно, щоб один або кілька тегів застосовувалися до всього зображення, для виявлення об'єктів потрібно, щоб кожна мітка мала тег і область, що визначає прямокутник, що обмежує, для кожного об'єкта на зображенні. Портал служби Візуальне розпізнавання користувача надає графічний інтерфейс, який можна використовувати для позначення навчальних зображень. Найпростішим варіантом створення міток (рис 2.8) для виявлення об'єктів є використання інтерактивного інтерфейсу на порталі служби Візуальне розпізнавання користувача. Цей інтерфейс автоматично пропонує області, що містять об'єкти, яким можна призначити теги або налаштувати шляхом перетягування прямокутника, що обмежує, щоб приєднати об'єкт до мітки.

Крім того, після додавання позначки до вихідного пакета зображень модель можна навчити. Наступні позначки нових зображень можуть використовувати переваги засобу смарт-маркувальника на порталі. Цей засіб може пропонувати не лише області, а й класи об'єктів, які вони містять. Якщо все-таки використовувати засіб створення міток, відмінний від порталу служби Візуальне розпізнавання користувача, може знадобитися налаштувати вихідні дані відповідно до одиниць вимірювання, очікуваних API служби Візуальне розпізнавання користувача. Обмежуючі прямокутники визначаються чотирма значеннями, що представляють ліві (X) і верхні (Y) координати верхнього лівого кута прямокутника, що обмежує, а також його ширину і висоту.

Ці значення виражаються як пропорційних значень щодо розміру вихідного зображення [23].

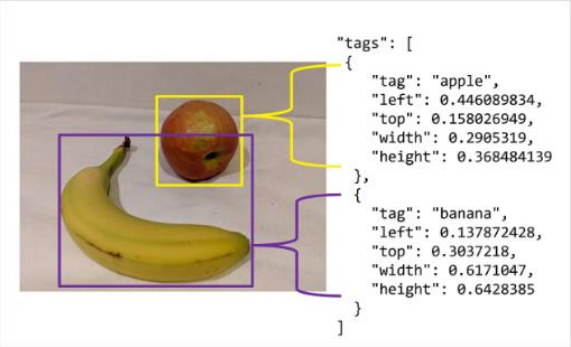

Рисунок 2.8 – Приклад відомості про позначки для об'єктів на зображенні у форматі JSON. [25]

Наприклад, розглянемо цей обмежуючий прямокутник – Зліва: 0,1, Зверху: 0,5, Ширина: 0,5, Висота: 0,25. Це визначає поле, в якому значення ліворуч знаходиться 0,1 (одна десята) від лівого краю зображення, а значення зверху - 0,5 (половина висоти зображення) зверху. Поле має половину ширини та чверть висоти всього зображення.

2.5 Висновки до розділу 2

В другому розділі було розглянуто різноманітні види та способи реалізації систем технічного зору, також було приведено декілька прикладів для більш точного розуміння та остаточного рішення що до вибору потрібного та більш відповідного методу та технології для реалізації проєкта. Також в цьому розділі було розглянуто декілька контрелерів з прикладами, та наведенням плюсів та мінусів кожного, для вибору.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МОБІЛЬНОЇ ПЛАТФОРМИ

3.1 Розробка макету платформи

Ідея цього проєкту з’явилася ще коли я працював на заводі і мене змусили сортувати запчастини, реалізувати це мене підштовхнув проект «Splash» . Splash – це проект створення автоматизованого сміттєвого бака, який автоматично сортує сміття, визначаючи що можна пустити на вторинну переробку.

Принцип роботи: у верхній частині бака встановлена камера, знімки з якої виходять скриптом на Python і далі програмою на C# відправляються на сервіс Azure Custom Vision API, де модель машинного навчання визначає тип матеріалу, що викидається. Використовуючи повертається значення, контролер Arduino включає двигун, який повертає кришку, направляючи сміття у відповідне відділення [24].

Для реалізації проєкта я використав один з принципів проєктування та програмування, який називається «абстракція». У контексті ОВП абстракція – це узагальнення даних та поведінки для типу, що знаходиться вище за поточний клас з ієрархії. Переміщаючи змінні чи методи з підкласу до суперкласу, ви узагальнюєте їх. Це загальні поняття, і вони застосовні у Java. Але мова додає також поняття абстрактних класів та абстрактних методів. Якщо не вдаватись в термінологію програмування то можна сказати, що абстракція – це спосіб виділити набір значущих характеристик об'єкта, виключаючи з розгляду не значущі. Відповідно, абстракція – це набір всіх таких характеристик.

Тож, говорячи про проєкт, я також взяв за основу бак в якому відбувається процес сортування. Для створення баку було використано Arduino Uno, ультразвуковий датчик та сервопривід. Нижче приведена схема поєднання (рис. 3.1)

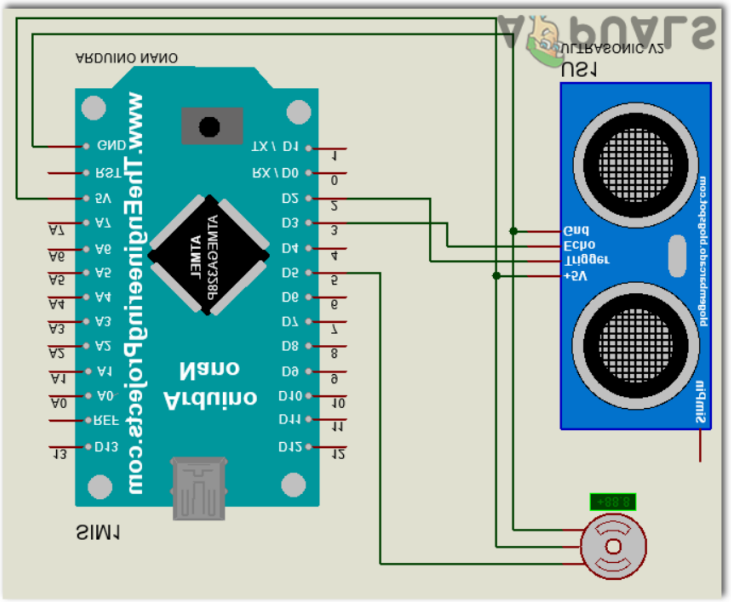

Рисунок 3.1 –Схема підключення

Також мною був написаний код, який зображено нижче (рис 3.2 – 3.4) для автоматичного відкриття контейнера.

```
int duration, distance; // Объявляем переменные для хранения расстояния и типа ультразвукового сигнала
Servo servo; // Объявляем объект для серводвигателя
```

Рисунок 3.2 – Оголошення змінних та об’єктів

```
int const trigPin = 2; // Соединяем контакт 2 Arduino с триггером ультразвукового датчика
int const echoPin = 3; // Соединяем вывод 3 arduino с эхом ультразвукового датчика
```

Рисунок 3.2 – Підключення компонентів до пінів ардуіно

Далі буде описано блок коду void setup.

```
Serial.begin (9600); // установка скорости передачи микроконтроллера
pinMode (trigPin, OUTPUT); // триггерный вывод будет использоваться как выходной
pinMode (echoPin, INPUT); // Пин эхо будет использоваться в качестве входа
servo.attach (5); // Подключить серводвигатель к контакту 5 Arduino
```

Рисунок 3.3 –Ініціалізація контактів

Void setup – це функція, у якій ми ініціалізуємо контакти плати Arduino, які використовуватимуться як INPUT чи OUTPUT. Тригерний висновок використовуватиметься як вихід, а луна-контакт — як вход. Ми використовували об'єктний сервопривід для підключення двигуна до висновку Arduino. Контакт може використовуватися для надсилання сигналу ШІМ. Швидкість передачі даних також встановлюється у цій функції. Швидкість у бодах – це швидкість передачі бітів за секунду, з якою мікроконтролер зв'язується із зовнішніми пристроями.

Далі блок «void loop» – це функція, яка знову і знову запускається у циклі. У цьому циклі ультразвукова хвиля відправляється в довкілля та приймається назад. Пройдена відстань вимірюється з використанням часу, витраченого сигналом, щоб залишити датчик і повернутися до нього. Потім умова застосовується до відстані відповідно.

```
{
  digitalWrite (trigPin, HIGH); // отправка ультразвукового сигнала в окружающую среду
  delay (1000);
  digitalWrite (trigPin, LOW); // Измеряем импульсный вход в эхо-контакт
  duration = pulseIn (echoPin, HIGH); // Расстояние равно половине продолжительности
  distance = (duration / 2) / 29,1; // если расстояние меньше 0,5 метра и больше 0
  (0 или меньше означает превышение диапазона)

  if (distance <= 50 && distance >= 0)
  {
  servo.write (50);
  delay (3000);
  }

    if else
      {
      servo.write (160);
      }
}
}
```

Рисунок 3.4 – Блок коду void loop

3.2 Програмне забезпечення для отримання зображення за допомогою Python

Для отримання зображення з камери було написано скрипт на python з використанням бібліотеки OpenCV, який буде наведено нижче. OpenCV – це open source бібліотека комп'ютерного зору, яка призначена для аналізу, класифікації та обробки зображень. Широко використовується у таких мовах як C, C++, Python та Java.

Перед тим як перейти до коду, потрібно трохи розібратися з теорією. Кожне зображення складається із набору пікселів. Піксель – це будівельний блок зображення. Якщо уявити зображення як сітки, кожен квадрат у сітці містить один піксель, де точці з координатою (0, 0) відповідає верхній лівий кут зображення. Наприклад, уявимо, що у нас є зображення з роздільною здатністю 400x300 пікселів. Це означає, що наша сітка складається з 400 рядків та 300 стовпців. У сукупності у зображенні є 400*300 = 120000 пікселів.

У більшості зображень пікселі представлені двома способами: у відтінках сірого (рис 3.5) та у колірному просторі RGB. У зображеннях у відтінках сірого кожен піксель має значення між 0 і 255 де 0 відповідає чорному, а 255 відповідає білому. А значення між 0 і 255 приймають різні відтінки сірого, де значення ближче до 0 темніші, а значення ближче до 255 світліші.

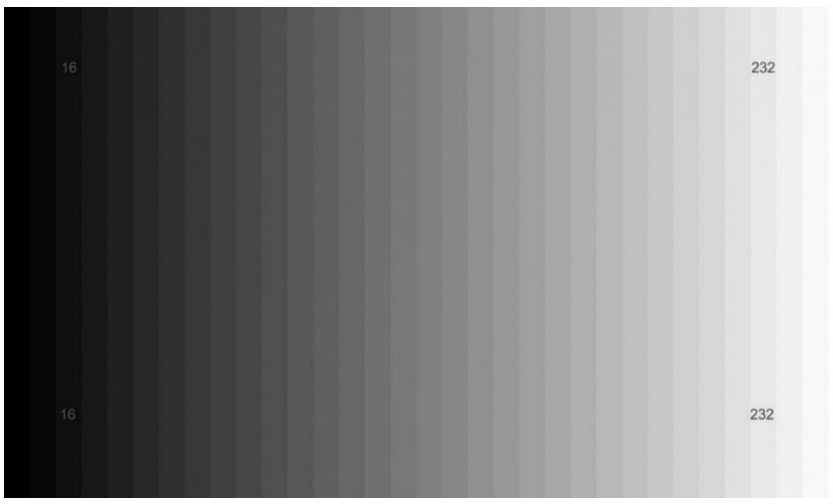

Рисунок 3.5 – Відтінки сірого

Кольорові пікселі зазвичай представлені в колірному просторі RGB (red, green, blue — червоний, зелений, синій), де одне значення для червоної компоненти, одне для зеленої та синій. Кожна з трьох компонентів представлена цілим числом в діапазоні від 0 до 255 включно, яке вказує як «багато» кольору міститься. Виходячи з того, що кожна компонента представлена в діапазоні [0,255], для того, щоб уявити насиченість кожного кольору, нам буде достатньо 8-бітного цілого без знакового числа. Потім ми об'єднуємо значення всіх трьох компонентів у кортеж виду (червоний, зелений, синій). Наприклад, щоб отримати білий колір, кожна з компонентів повинна дорівнювати 255: (255, 255, 255). Тоді, щоб отримати чорний колір, кожна з компонентів повинна бути рівною 0: (0, 0, 0). Нижче наведені поширені кольори (рис 3.6), представлені у вигляді кортежів RGB.

Рисунок 3.6 – Поширенні кольори у форматі RGB кортежів

Для початку були імпортовані потрібні бібліотеки (рис 3.5).

```
import cv2
import time
```

Рисунок 3.5 – Імпорт бібліотеки для часу та для OpenCV

Модуль cv2 – це і є OpenCV. Його ми будемо використовувати для аналізу зображень та відео.

В наступній частині коду ми прочитаємо зображення за допомогою методу «read» (рис 3.6). Ця функція захоплює, декодує та повертає наступний відеокадр.

[out] зображення відеокадр повертається сюди. Якщо жодного кадру не було захоплено, зображення буде порожнім.

False, якщо не було захоплено жодного кадру.

Метод/функція поєднує VideoCapture::grab() і VideoCapture::retrieve() в одному виклику. Це найзручніший спосіб для читання відеофайлів або захоплення даних із декодування та повертає щойно захоплений кадр. Якщо жодного кадру не було захоплено (камеру було від’єднано або у відеофайлі більше немає кадрів), метод повертає false, а функція повертає порожнє зображення (з cv::Mat, перевірте його за допомогою Mat::empty()) .

В API C функції cvRetrieveFrame() і cv.RetrieveFrame() повертають зображення, що зберігається всередині структури захоплення відео. Забороняється змінювати або випускати зображення! Ви можете скопіювати кадр за допомогою cvCloneImage, а потім робити з копією все, що хочете.

Після цього для відображення зображень використовується метод «imshow» (рис 3.6). Перший аргумент – заголовок зображення, а другий – сама змінна зображення.

Функція imshow відображає зображення у вказаному вікні. Якщо вікно було створено з прапорцем cv::WINDOW_AUTOSIZE, зображення відображається в оригінальному розмірі, однак воно все ще обмежене роздільною здатністю екрана. В іншому випадку зображення масштабується відповідно до вікна. Функція може

масштабувати зображення в залежності від його глибини:

– Якщо зображення 8-бітне без знаку, воно відображається як є.

– Якщо зображення 16-бітне без знаку, пікселі діляться на 256. Тобто діапазон значень [0,255*256] відображається на [0,255].

– Якщо зображення є 32-розрядним або 64-розрядним із плаваючою комою, значення пікселів множаться на 255. Тобто діапазон значень [0,1] відображається на [0,255].

– 32-розрядні цілі зображення більше не обробляються через неоднозначність необхідного перетворення. Перетворення на 8-розрядну без знакову матрицю за допомогою спеціальної попередньої обробки, специфічної для контексту зображення.

Якщо вікно було створено з підтримкою OpenGL, cv::imshow також підтримує ogl::Buffer , ogl::Texture2D і cuda::GpuMat як вхідні дані.

Якщо вікно не було створено до цієї функції, передбачається створення вікна з cv::WINDOW_AUTOSIZE.

Якщо потрібно показати зображення, яке перевищує роздільну здатність екрана, вам потрібно буде викликати namedWindow(" ", WINDOW_NORMAL) перед imshow. За цією функцією має слідувати виклик cv::waitKey або cv::pollKey для виконання робочих завдань графічного інтерфейсу користувача, необхідних для фактичного показу даного зображення та змусити вікно реагувати на події миші та клавіатури. Інакше зображення не відображатиметься, і вікно може заблокуватися. Наприклад, waitKey(0) відображатиме вікно нескінченно до будь-якого натискання клавіші (це підходить для відображення зображення). waitKey(25) відобразить кадр і зачекає приблизно 25 мс для натискання клавіші (придатне для покадрового відображення відео). Щоб видалити вікно, використовуйте cv::destroyWindow.

Далі використовуємо метод «imWrite» (рис 3.6), він зберігає зображення у вказаний файл.

Функція imwrite зберігає зображення у вказаний файл. Формат зображення вибирається на основі розширення імені файлу. Загалом за допомогою цієї функції можна зберегти лише 8-бітні одноканальні або 3-канальні (з порядком каналів

«BGR») зображення, за такими винятками:

– 16-розрядні зображення без підпису (CV_16U) можна зберігати у форматах PNG, JPEG 2000 і TIFF;

– 32-розрядні зображення з плаваючою точкою (CV_32F) можна зберігати у форматах TIFF, OpenEXR і Radiance HDR; 3-канальні (CV_32FC3) зображення TIFF будуть збережені з використанням кодування високого динамічного діапазону LogLuv (4 байти на піксель);

– зображення PNG з альфа-каналом можна зберегти за допомогою цієї функції. Для цього створіть 8-бітне (або 16-бітне) 4-канальне зображення BGRA, де альфа-канал йде останнім. Абсолютно прозорі пікселі повинні мати значення альфа-каналу 0, повністю непрозорі пікселі мають мати значення 255/65535 ;

– кілька зображень (вектор Mat) можна зберегти у форматі TIFF.

Якщо формат зображення не підтримується, зображення буде перетворено на 8-бітове без знаку (CV_8U) і збережено таким чином.

```
test, frame = cap.read()
cv2.imshow("Preview",frame)
cv2.imwrite("frame%d.jpg" % count, frame)
time.sleep(4)
```

Рисунок 3.6 – Використання основних методів

Далі йде умова для зупинення, в якій сказано, що код зупиниться коли буде натиснута будь яка клавіша, та повністю заповнений одиницями байт (0xFF), в ній використовується функція «waitKey». Чекає на натиснуту клавішу. Функція waitKey чекає ключову подію нескінченно (коли delay≤0) або протягом милі секунд затримки, якщо вона позитивна. Оскільки ОС має мінімальний час між перемиканням потоків, функція не чекатиме точної затримки мс, вона чекатиме принаймні затримки мс, залежно від того, що ще запущено на вашому комп’ютері в цей час. Він повертає код натиснутої клавіші або -1, якщо жодна клавіша не була натиснута до закінчення зазначеного часу. Щоб перевірити натискання клавіші, але

не чекати її, використовуйте pollKey. Функції waitKey (рис 3.7) і pollKey є єдиними методами у HighGUI, які можуть отримувати та обробляти події графічного інтерфейсу, тому один із них потрібно періодично викликати для нормальної обробки подій, якщо тільки HighGUI не використовується в середовищі, яке піклується про обробку подій.

Функція працює, лише якщо створено хоча б одне вікно HighGUI і воно активне. Якщо є кілька вікон HighGUI, будь-яке з них може бути активним.

```
13        if cv2.waitKey(1) & 0xFF == ord('q'):
14            break
```

Рисунок 3.7 – Умова зупинення виконання

Далі завдяки функції «release» (рис 3.8) закриваємо відеофайл або пристрій захоплення.

Метод автоматично викликається наступним VideoCapture::open і деструктором VideoCapture.

Та за допомогою функції «destroyAllWindows» (рис 3.8) знищує всі вікна HighGUI.

Ця функція s знищує всі відкриті вікна HighGUI.

```
16    cap.release()
17    cv2.destroyAllWindows()
```

Рисунок 3.8 – Функції для завершення роботи

3.3 Навчання розпізнання об’єкта за допомогою Azure

Azure custom vision – ця служба дає змогу навчити модель машинного навчання за допомогою зображень об'єктів. Для подальшого використання навчену модель для розпізнавання схожих об'єктів, як забезпечується захопленням камери підключеної до комп'ютера.

Azure Користувальницьке візуальне розпізнавання — це Microsoft Cognitive Service, яка дозволяє розробникам створювати власні класифікатори образів. Потім ці класифікатори можна використовувати з новими зображеннями для розпізнавання чи класифікації об'єктів у новому зображенні. Служба надає простий, простий у використанні веб-портал для спрощення процесу.

Спочатку зайдемо на головну сторінку служби Візуальне розпізнавання користувача.

Далі авторизуючись та згоджуючись з усіма ліцензійними узгодженнями потрапляємо в розділ «Проєкти» і створюємо свій проект (рис 3.9)

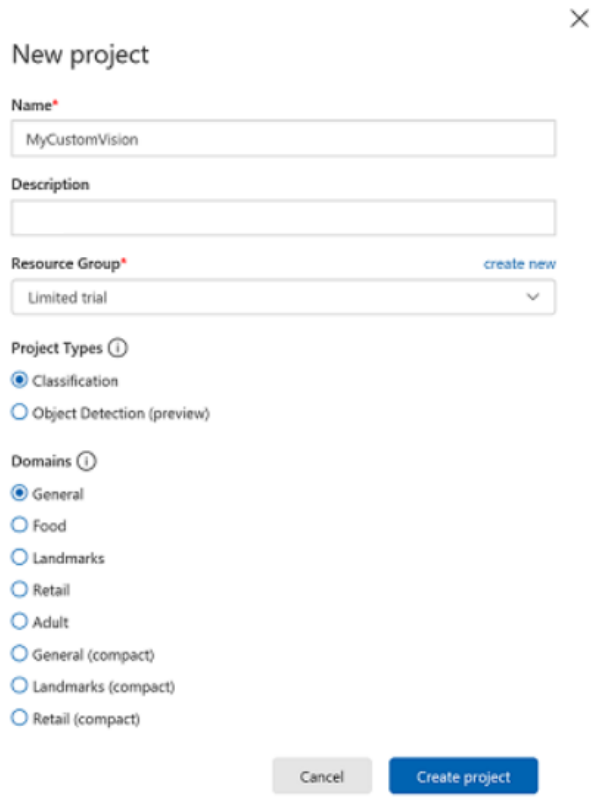

Рисунок 3.9 – Створення проєкту

Щоб ресурс розпізнав об'єкт знадобиться від 5 до 10 зображень (рис 3.10). В цьому проєкті для прикладу використовую комп’ютерну мишу та клавіатуру.

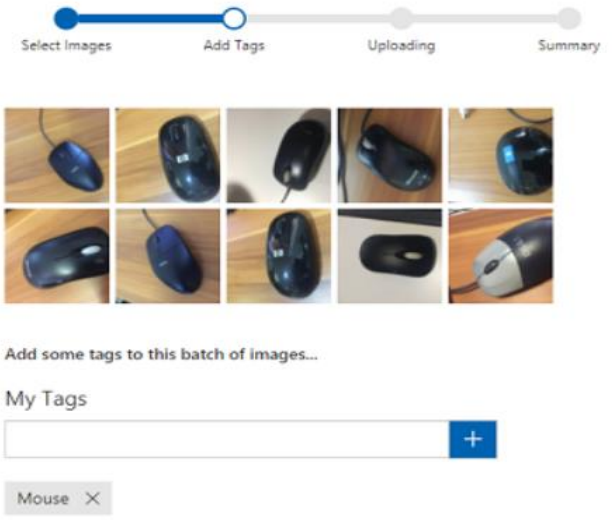

Рисунок 3.10 – Приклад додавання зображення для розпізнання

Після додавання зображення обох об'єктів починаємо першу ітерацію навчання. Після створення можна побачити дві кнопки з ім'ям «Зробити за замовчуванням» та «URL-адресу прогнозування». Спочатку натискаємо «Зробити за замовчуванням» , а потім тиснемо URL-адресу прогнозування (рис3.11). URL-адреса кінцевої точки, надана з цього параметра, встановлюється залежно від того, яка ітерація позначена за промовчанням. Таким чином, якщо пізніше зробити нову ітерацію та оновіть її за замовчуванням, не доведеться змінювати код.

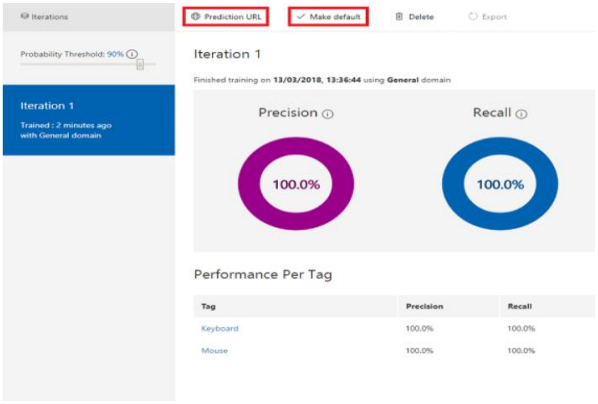

Рисунок 3.11 – Кнопки які описано вище, та 100% обробка обох об’єктів

3.4 Програмне забезпечення для обробки зображення за допомогою C#

Для обробки зображення було написано код на C#. Для початку розглянемо простір імен які будуть використані в проєкті (рис 3.12).

```
1  using System;
2  using System.IO;
3  using System.IO.Ports;
4  using System.Net.Http;
5  using System.Net.Http.Headers;
6  using System.Threading.Tasks;
```

Рисунок 3.12 – Простори імен для проєкта

В цьому проєкті прийдеться працювати с файловою системою, тож розглянемо простір імен, який допоможе з цією роботою, а саме System.IO.

Фреймворк .NET надає великі можливості з управління та маніпуляції файлами та каталогами, які здебільшого зосереджені у просторі імен System.IO. Класи, розташовані в цьому просторі імен (такі як Stream, StreamWriter, FileStream та ін), дозволяють керувати файловим введенням-виводом.

System.IO.Ports – содержит классы для управления послідовними портами. Найбільш важливий з них, клас SerialPort, забезпечує засоби для синхронного та керованого подіями вводу-виводу, для доступу до стану підключення-відключення пристрою, а також для доступу до властивостей драйвера послідовного порту. Він може використовуватися для упаковки об'єктів Stream, що відкриває доступ до послідовного порту з класів, що використовують потоки.

System.Net.Http – надає інтерфейс програмування для сучасних додатків HTTP.

System.Net.Http.Headers – Забезпечує підтримку колекцій заголовків HTTP, використовуваних простором з іменем System.Net.Http.

System.Threading.Tasks – надає типи, які спрощують роботу по написанню паралельного та асинхронного коду. Основні типи: Завдання, що представляє

асинхронну операцію, яку можна очікувати та змінити, і Завдання<TResult>, що представляє собою задачу, яка може повернути значення. Клас TaskFactory надає статичні методи для створення завдань, а клас TaskScheduler надає інфраструктуру планування потоків за замовчуванням.

Далі вказуємо шлях до зображень (рис 3.13).

```
string imageFilePath = "C:\\Users\\virag\\Desktop\\HackGSU\\frame0.jpg";
```

Рисунок 3.13 – Шлях до зображення

Далі використовуємо асинхронний метод «public static async Task MakePredictionRequest» та передаємо параметром шлях до зображення «(string imageFilePath)» (рис 3.14).

```
public static async Task MakePredictionRequest(string imageFilePath)
{
```

Рисунок 3.14 – Асинхронний метод з параметром

Після цього оголосимо та ініціалізуємо потрібні нам змінні (рис 3.15).

```
var client = new HttpClient();
int recycle = 1;
SerialPort port;
```

Рисунок 3.15 – Оголошення та ініціалізація

Далі напишемо request headers та додаємо параметром prediction key (рис 3.16).

```
client.DefaultRequestHeaders.Add("Prediction-Key", "0f555d505ca743328e7a976a5e4683af");
```

Рисунок 3.16 – Prediction key

Також в асинхронному методі відбувається заповнення тіла запиту (рис 3.17)

```
content.Headers.ContentType = new MediaTypeHeaderValue("application/octet-stream");
response = await client.PostAsync(url, content);
string longass = await response.Content.ReadAsStringAsync();
string subOne = longass.Substring(longass.IndexOf("tagName"));
string subTwo = subOne.Substring(subOne.IndexOf(":"));
string subThree = subTwo.Substring(subTwo.IndexOf(":") + 1,subTwo.IndexOf("}") - 1);
Console.WriteLine(subThree);
```

Рисунок 3.17 – Заповнення тіла запиту

Після цього написано метод який получає зображення побайтово (рис 3.18)

```
private static byte[] GetImageAsByteArray(string imageFilePath)
{
    FileStream fileStream = new FileStream(imageFilePath, FileMode.Open, FileAccess.Read);
    BinaryReader binaryReader = new BinaryReader(fileStream);
    return binaryReader.ReadBytes((int)fileStream.Length);
}
```

Рисунок 3.18 – Метод для отримання зображення по байтам

3.5 Висновки до розділу 3

В третьому розділі завдяки проведеної роботи в першому та другому, а саме завдяки отриманої інформації було розроблена схему підключення потрібних нас компонентів до обраного контролера, а саме Arduino Uno. Також після цього було написано скрип на язику програмування Python для отримання зображення з камери встановленої на контейнері в якому відбувається сортування. Коли вже можливо отримувати зображення з камер було почато навчання проєкта для розпізнавання об’єктів, для прикладу було взято комп’ютерну миш та клавіатуру , і в кінці кінців було написано код для відправки на сервер та обробки зображень за допомогою язика C#

ВИСНОВКИ

У вступі були сформульовані завдання та цілі та у кратці описано актуальність роботи. У першому розділі розглядаються підходи та технології для сортування деталей на підприємствах. Також методи розпізнання об’єктів на фото та відео, було приведено декілька прикладів для спільного розуміння які можливості відкриваються з правильним використанням цих технологій.

Впровадження датчиків нового покоління та швидкодіючої обчислювальної техніки призвело до створення автоматизованої системи машинного зору. З його допомогою вдалося вирішити цілу низку завдань – підвищити якість продукції, мінімізувати вплив людського чинника, відстежувати переміщення продукції та багато іншого. Модулі візуального контролю широко потрібні в різних галузях, зокрема в промисловості, фармацевтиці, науці та техніці.

За допомогою технології машинного зору виконується комплекс різних завдань від зчитування та контролю графічних кодів до контролю якості друку та відповідності етикетки встановленим стандартам.

Основними завданнями, які були вирішенні під час виконання кваліфікаційної роботи:

- ознайомлення з технологіями машинного зору;
- ознайомлення з сенсорами для реалізації зору;
- проаналізовано технічну літературу по темі проєкта;
- проаналізовано систем технічного зору в робототехніці;
- розглянуто варіанти технічної модернізації платформи;
- розроблено програмне забезпечення для мобільної платформи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. – Харків: ХНУРЕ, 2021. – 55 с.

2. ДСТУ 3008-15. Документація. звіти у сфері науки і техніки. структура і правила оформлення [Текст] – Введ. 2015-06-22. – К. Держстандарт України, 2017. – 29 с.

3. Освітньо – професійна програма «Автоматизоване управління технологічними процесами» другого (магістерського) рівня вищої освіти за спеціальністю 151 – «Автоматизація та комп'ютерно – інтегровані технології» галузь знань 15 «Автоматизація та приладобудування» (Вчена рада ХНУРЕ протокол № 1 від 28.01.2021 р.) [Електронний ресурс]. – Режим доступу: https://tapr.nure.ua/wp-content/uploads/2021/03/opp_autp_2021_compressed.pdf.

4. Положення про кваліфікаційну роботу здобувача вищої освіти на другому (магістерському) рівні: Наказ ХНУРЕ від 06 травня 2021 р. № 143 [Електронний ресурс]. – Режим доступу: https://nure.ua/wp-content/uploads/Main_Docs_NURE/143-vid-06.05.2021-pro-vvedennja-v-dijurishennjavchenoiradi - universitetu.pdf.

5. Невлюдов І.Ш. Основи наукових досліджень: навч. посібник / І. Ш. Невлюдов, Ю. М. Олександров, А. О. Андрусевич, О. О. Чала. – Кривий Ріг: Криворізький коледж НАУ, 2019. – 396 с.

6. Освітньо-професійна програма «Комп'ютерно-інтегровані технологічні процеси і виробництва». – Режим доступу :

https://nure.ua/wpcontent/uploads/Education_programs/2021/2021_mag_151_opp_kitpv.pdf.

7. Методы детекции объектов. [Електронний ресурс]. URL: https://www.zebra.com/ru/ru/solutions/intelligent-edge-solutions/rtls/what-are-location-solutions.html.

8. Технології детекції об’єктів. [Електронний ресурс]. URL: https://habr.com/ru/post/397757/.

9. Системи сортування. [Електронний ресуср] Url: https://www.zebra.com/ru/ru/solutions/intelligent-edge-solutions/rtls.

10. Методы и технологии сортировки на промышленных объектах. [Елктронний ресур] Url: https://habr.com/ru/post/280848/.

11. Невлюдов І.Ш. Виробничі процеси та обладнання об’єктів автоматизації. Збірник задач: Навчальний посібник / І.Ш. Невлюдов, А.О. Андрусевич, Г.В. Пономарьова, А.О. Функендорф. Кривий Ріг: КК НАУ. 2018. 332 с.

12. YOLOv4 – самая точная real-time нейронная сеть на датасете Microsoft COCO [Електронний ресурс] Url: https://habr.com/ru/post/503200/.

13. YOLOv4 – нейронная сеть на датасете [Електронний ресурс] Url: https://habr.com/ru/post/529410/.

14. Що таке Yolo та як використовувати. [Електронний ресурс]. URL: https://secretmag.ru/enciklopediya/chto-takoe-yolo-obyasnyaem-prostymi-slovami.htm.

15. Which systems are used for goods sorting in the warehouse and how do you find the best solution? [Електронний ресурс]. URL: https://kapelou.com/en/blog/distributor/systemy-sortuvannia-tovariv/.

16. Как создать робота: обзор современных технологий. [Електронний ресурс]. – URL: https://evergreens.com.ua/ru/articles/how-to-create-robots.html.

17. Задача нахождения объектов на изображении. [Електронний ресурс]. URL: https://neerc.ifmo.ru/wiki/index.php.

18. Плата Arduino Uno R3: схема, описание, подключение устройств. [Електронний ресурс]. – URL: https://arduinomaster.ru/platy-arduino/plata- arduino-

uno/.

19. Ардуино. [Електронний ресурс] Url: https://robocraft.ru/arduino/14.

20. Arduino vs Raspberry PI. [Електронний ресурс] Url: https://www.interviewbit.com/blog/arduino-vs-raspberry-pi/#:~:text=Arduino%20boards%20are%20micro%2Dcontrollers,IDE%20(Integrated%20Development%20Environment).

21. Розробка і автоматизація. [Електронний ресурс]. Url: https://electronoff.ua/arduino.

22. Raspberry Pi: описание, подключение, схема [Електронний ресурс] Url: https://3d-diy.ru/wiki/arduino-platy/obzor-plat-raspberry-pi/.

23. Распознавание объектов на изображении. [] Url: https://learn.microsoft.com/ru-ru/training/modules/detect-objects-images/.

24. Проект Splash. [Електронний ресурс]. – URL: https://robocraft.ru/projects/4162.