

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки

«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
_____ д.т.н., проф. С. І. Шматков
«___» _____ 2021 р

Пояснювальна записка

до кваліфікаційної роботи
магістра

на тему: «Програмна модель розпізнавання образів для мобільного робота»

Захищено на засіданні
Атестаційної комісії № 39
протокол № __ від __.12.2021 р.
Оцінка _____ / _____
Голова Атестаційної комісії
_____ д.т.н., професор
_____ **МІНУХІН**
Сергій Володимирович

Виконав:

студент 6 курсу, групи КУ– 61
Галузь знань: 15 – Автоматизація та
приладобудування
за спеціальністю 151 – Автоматизація та
комп'ютерно-інтегровані технології.
Капуста Максим Анатолійович _____

Керівник:

_____ д.т.н., доцент кафедри
теоретичної та прикладної
системотехніки
СТРІЛЕЦЬ Вікторія Євгенівна _____

Рецензент:

_____ д.т.н., доцент електроніки та
управляючих систем
СТЕРВОЄДОВ Микола
Григорович _____

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи «Програмна модель розпізнавання образів мобільним роботом» складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи складає 86 сторінок із яких 65 сторінок основної частини, 2 таблиці, 41 рисунок, 24 найменувань списку використаних джерел на 2 сторінках і 4 додатків на 11 сторінках.

Дане дослідження присвячене можливості використання згорткових нейронних мереж в мобільних роботах для розпізнавання об'єктів в реальному часі. Його метою є підвищення точності розпізнавання образів у мобільному роботі в реальному часі за допомогою згорткових нейронних мереж. Під час виконання даної роботи були розглянуті класичні методи розпізнавання об'єктів на основі машинного навчання. Для досягнення поставленої мети був знайдений відповідний набір даних, обрано пакет програмного забезпечення.

Результатом даного дослідження є натренована згорткова нейронна мереж ResNet-18 яка розпізнає об'єкти. Областю використання запропонованого робота є місця підвищеної небезпеки, завали тощо.

Ключові слова: згорткові нейронні мережі, розпізнавання образів, мобільний робот, PYTHON, OPENCV.

ABSTRACT

The explanatory note to the qualification work "Software model of pattern recognition by a mobile robot" consists of an introduction, three sections, conclusions, a list of used sources and appendices. The total volume of the work is 86 pages, of which 65 pages of the main part, 2 tables, 41 figures, 24 titles of the list of used sources on 2 pages and 4 appendices on 11 pages.

This study focuses on the possibility of using convolutional neural networks in mobile robots to recognize objects in real time. Its goal is to increase the accuracy of image recognition in mobile work in real time using convolutional neural networks. During this work, the classical methods of object recognition based on machine learning were considered. To achieve this goal, an appropriate data set was found, a software package was selected.

The result of this study is a trained convolutional neural network ResNet-18 that recognizes objects. The scope of the proposed robot is a place of increased danger, blockage, etc.

Keywords: convolutional neural networks, pattern recognition, mobile robot, PYTHON, OPENCV.

ЗМІСТ

Вступ.....	6
Розділ 1. Аналіз технологій і методів розпізнавання образів	8
1.1 Технології розпізнавання зображень.....	8
1.2 Сучасні методи розпізнавання зображень	9
1.3 Штучні нейронні мережі	12
1.3.1 Персептрон	15
1.3.2 Згорткові нейронні мережі.....	18
1.3.3. Рекурентна нейронна мережа	21
1.4 Базові алгоритми навчання нейронних мереж	22
1.4.1 Градієнтний спуск.....	22
1.4.2 Алгоритм зворотного розповсюдження помилки.....	25
Висновок за розділом 1.....	27
Розділ 2. Розробка мобільного робота для спостереження	29
2.1 Загальні відомості про мобільних роботів.....	29
2.2 Структура мобільного робота	34
2.3 Мобільні роботи спеціального призначення	37
2.4 Бойові роботи	46
2.5 Побутові роботи	50
2.6 Обґрунтування та вибір основних компонентів мобільного робота.....	52
2.7 Проектування мобільного робота.....	56
Висновок за розділом 2.....	59
Розділ 3. Розробка програмного забезпечення для мобільного робота	60
3.1 Розробка програмного забезпечення для управління роботом.....	60
3.2 Структура згорткової нейронної мережі.....	60
3.3 Розробка згорткової нейронної мережі.....	66

Висновок за розділом 3	71
Висновки	72
Список літератури	73
Додатки	75
Додаток А	75
Додаток Б	78
Додаток В	82
Додаток Г	86

ВСТУП

Сучасні методи розпізнавання образів мають досить великий спектр використання. Щодня великі компанії та невеликі команди ентузіастів створюють нові підходи та алгоритми обробки кадрів аби розвивати технології комп'ютерного зору.

Але деякі проблеми, такі як розпізнавання об'єктів в зашумлених, пошкоджених фото, не можуть вирішити класичні алгоритми та багатосарові нейронні мережі. Перспективу в подоланні цих труднощів бачать у новій нейромережевій парадигмі – згорткові нейронні мережі. Вони мають велику кількість переваг в порівнянні з класичною схемою нейронної мережі на базі перцептрона.

Робототехніка за останній час отримує все більше і більше сфер де використовують роботів, оскільки це стає вигідніше. Все більшу популярність набирають воєнні та рятувальні роботи, які можуть знешкоджувати міни, патрулювати вулиці тощо. Але майже всі такі роботи призначені для роботи в місцях на відкритому повітрі, і є досить великими та важкими.

Таким чином **актуальність** роботи полягає у використанні згорткових нейронних мереж для вирішення завдання пошуку об'єктів за допомогою мобільного робота. У даній роботі розглянуто технології навчання згорткових мереж для розпізнавання об'єктів в реальному часі, та віддалене управління роботом. **Метою** роботи є підвищення точності розпізнавання образів у мобільному роботі в реальному часі за допомогою згорткових нейронних мереж.

Таким чином **об'єктом дослідження** є процес розпізнавання образів мобільним роботом. **Задачу** розпізнавання образів можна сформулювати як аналіз даних з камери, та виділення основного об'єкту в кадрі для розпізнавання.

Тому предметом дослідження є моделі і методи розпізнавання образів, які використовуються у мобільних роботах.

Зараз процес розпізнавання є досить неточним, та навчання мережі потребує багато часу тому метою дослідження є підвищення точності розпізнавання образів за допомогою згорткових нейронних мереж.

РОЗДІЛ 1. АНАЛІЗ ТЕХНОЛОГІЙ І МЕТОДІВ РОЗПІЗНАВАННЯ ОБРАЗІВ

1.1 Технології розпізнавання зображень

Розпізнавання зображень, підкатегорія комп'ютерного зору та штучного інтелекту, являє собою набір методів виявлення та аналізу зображень для автоматизації конкретного завдання. Це технологія, яка здатна ідентифікувати місця, людей, предмети та багато інших типів елементів у зображенні та робити з них висновки, аналізуючи їх [1].

Класифікація зображень досягається шляхом диференціації зображення на встановлену категорію на основі змісту бачення. Класифікація зображень є одним із напрямів машинного навчання. Машинне навчання складається з модуля вилучення особливостей, який вилучає такі важливі ознаки, як ребра, текстури тощо, та модуля класифікації, який класифікує на основі вилучених особливостей. Основне обмеження машинного навчання полягає в тому, що, відокремлюючи, воно може витягувати лише певний набір ознак на зображеннях і не в змозі виділити диференційовані ознаки з навчального набору даних. Цей недолік виправляється за допомогою глибокого навчання [2].

Різні види класифікаторів мають свої переваги і недоліки. Так, класифікатори, в яких використовуються методи статистики мають хорошу математичну обґрунтованість, але при цьому складні у використанні і вимагають знання ймовірного розподілу вихідних даних і оцінки його параметрів (тому їх називають параметричними), а також мають фіксовану структуру моделі. Крім цього, статистичні методи оцінюють тільки ймовірність приналежності об'єкта класу, але не «пояснюють» чому.

Класифікатори, засновані на машинному навчанні не вимагають оцінки параметрів розподілу вихідних даних, а міра подібності в них формалізується за допомогою функції відстані. Такі класифікатори називаються метричними. Як правило, вони простіше в реалізації і використанні, ніж параметричні, а їх результати зручніше для інтерпретації і розуміння. Але при цьому метричні класифікатори є наближеними моделями – забезпечують рішення тільки в обмеженому числі практично значущих випадків, можуть дати неточне або не єдине рішення. Тому використовувати їх результати потрібно з певною часткою обережності.

Певним компромісом між параметричних і метричними методами є використання для вирішення завдань класифікації штучних нейронних мереж (НМ). Дійсно, НМ є непараметричними моделями, що не вимагають припущень про імовірнісний розподіл даних, але при цьому і не використовують заходи відстаней. Це робить їх універсальними класифікаторами, дозволяючи отримувати результати [3] навіть у випадках, коли параметричні і метричні класифікатори не забезпечує прийняттого рішення.

1.2 Сучасні методи розпізнавання зображень

Розглянемо основні методи класифікації, які відносяться до методів машинного навчання. Спочатку розглянемо статистичні підходи для класифікації, а саме [4]:

- логістична регресія;
- наївний байєсівський класифікатор;
- метод k -найближчих сусідів;
- дерева прийняття рішень;
- випадковий ліс;
- метод опорних векторів.

Логістична регресія. Логістична регресія це підвид різновид множинної регресії, загальне призначення якої полягає в аналізі зв'язку між декількома незалежними змінними (званими також регресорів або предикторами) і залежною змінною. Регресія в загальному вигляді застосовується, коли вхідні і вихідна змінні безперервні .

Наївний байєсівський класифікатор. Наївні методи Байеса – це набір алгоритмів навчання з учителем, заснованих на застосуванні теореми Байеса з «наївним» припущенням про умовну незалежності між кожною парою характеристик з урахуванням значення змінної класу.

Метод k -найближчих сусідів. Метод k -найближчих сусідів використовується для вирішення задачі класифікації. Він відносить об'єкти до класу, якому належить більшість з k його найближчих сусідів в багатовимірному просторі ознак. Це один з найпростіших алгоритмів навчання класифікаційних моделей. Число k – це кількість сусідніх об'єктів в просторі ознак, які порівнюються з класифікованих об'єктом.

Іншими словами, якщо $k = 10$, то кожен об'єкт порівнюється з 10-ю сусідами.

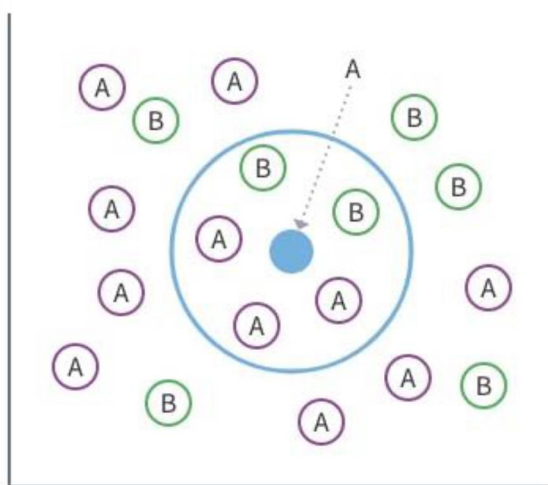


Рисунок 1.1 – Метод k -найближчих сусідів

Дерева прийняття рішень. Дерева рішень (DT) – це непараметричний контрольований метод навчання, що використовується для класифікації та регресії. Мета полягає в тому, щоб створити модель, яка передбачає значення цільової змінної шляхом вивчення простих правил прийняття рішень, виведених із особливостей даних. Дерево можна розглядати як кусочно постійне наближення.

Найпопулярнішим алгоритмом тренування дерева прийняття рішень є алгоритм ID3, який визначає черговість змінної і її атрибутів через їх інформаційну значимість (інформаційну ентропію). Для цього слід знайти ентропію всіх невикористаних ознак і їх атрибутів щодо тестових екземплярів і вибрати той, для якого ентропія мінімальна (а інформативність – максимальна).

На рисунку 1.2 зображений приклад дерев рішень.

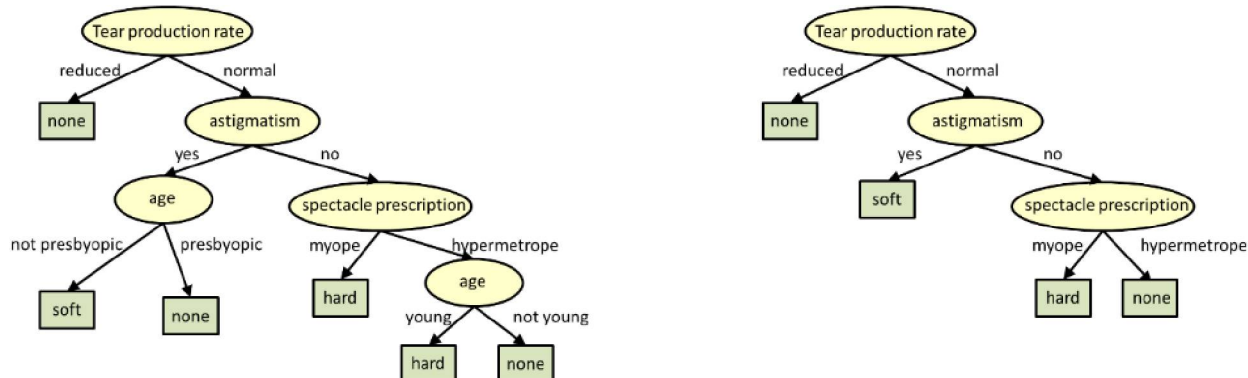


Рисунок 1.2 – Дерева рішень

Випадковий ліс. Випадковий ліс – це контрольований алгоритм навчання, який використовується як для класифікації, так і для регресії. Але, тим не менш, він в основному використовується для задач класифікації. Оскільки знаємо, що ліс складений з дерев, і більше дерев означає більш стійкий ліс. Точно так же алгоритм випадкового лісу створює дерева рішень для вибірок даних, а потім

отримує прогноз по кожній з них і, нарешті, вибирає краще рішення за допомогою голосування

Метод опорних векторів. Метод опорних векторів – один з найбільш популярних методів навчання, який застосовується для розв'язання задач класифікації і регресії. Основна ідея методу полягає в побудові гіперплощини, що розділяє об'єкти вибірки оптимальним способом. Алгоритм працює в припущенні, що чим більше відстань (зазор) між роздільною гіперплощиною і об'єктами поділених класів, тим менше буде середня помилка класифікатора.

Основна ідея класифікатора на опорних векторах полягає в тому, щоб будувати роздільну поверхню з використанням тільки невеликої підмножини точок, що лежать в зоні, критичної для поділу, тоді як інші, вірно класифіковані спостереження навчальної вибірки поза цією зоною, ігноруються (точніше, є "резервуаром" для оптимізаційного алгоритму).

1.3 Штучні нейронні мережі

Штучні нейронні мережі були розроблені для моделювання за структурою мозку. Вперше вони були розроблені дослідниками Уорреном Мак-Каллоком та Уолтером Піттом. Тоді модель спочатку називалась пороговою логікою, яка розгалужувалася на два різні підходи: один, натхненний більше біологічними процесами, а другий, орієнтований на застосування штучного інтелекту.

Основні етапи в історії дослідження та застосування штучних нейронних мереж:

- 1943 р. – У. Маккалок і У. Піттс формалізують поняття нейронної мережі у фундаментальній статті про логічне обчислення ідей та нервової активності.

– 1948 р. – Н. Вінер разом із соратниками публікує роботу про кібернетику. Основною ідеєю є представлення складних біологічних процесів математичними моделями.

– 1949 р. – Д. Хебб пропонує перший алгоритм навчання.

– 1958 р. – Ф. Розенблатт винаходить одношаровий перцептрон і демонструє його здатність розв'язувати задачі класифікації. Перцептрон набув популярності – його використовують для розпізнавання образів, прогнозування погоди тощо.

– 1960 р. – Уїдроу спільно зі своїм студентом Хоффом на основі дельта-правила (формули Уїдроу) розробили Адалін, який одразу почав використовуватися для завдань передбачення та адаптивного управління. Наразі Адалін (адаптивний суматор) є стандартним елементом багатьох систем обробки сигналів.

– 1963 р. в Інституті проблем передачі інформації АН СРСР А.П. Петровим проводиться докладне дослідження завдань «важких» перцептронів.

– 1969 р. М. Мінський публікує формальний доказ обмеженості перцептронів і показує, що він нездатний вирішувати деякі завдання (проблема «парності» та «один у блоці»), пов'язані з інваріантністю уявлень. Інтерес до нейронних мереж різко спадає.

– 1972 р. Т. Кохонен та Дж. Андерсон незалежно пропонують новий тип нейронних мереж, здатних функціонувати як пам'ять.

– 1973 р. – Б. В. Хакімов пропонує нелінійну модель із синапсами на основі сплайнів і впроваджує її для вирішення завдань у медицині, геології, екології.

– 1974 р. – Пол Дж. Вербос та А. І. Галушкін одночасно винаходять алгоритм зворотного розповсюдження помилки для навчання багат шарових перцептронів

– 1975 р. – Фукусіма представляє когнітрон – мережу, що самоорганізується, призначену для інваріантного розпізнавання образів, але це досягається лише за допомогою запам'ятовування практично всіх станів образу.

– 1982 р. – після періоду забуття, інтерес до нейромереж знову зростає. Дж. Хопфілд показав, що нейронна мережа із зворотними зв'язками може являти собою систему, що мінімізує енергію (так звана мережа Хопфілда). Кохоненом представлені моделі мережі, що навчаються без вчителя (нейронна мережа Кохонена), що вирішує завдання кластеризації, візуалізації даних (карта Кохонена, що самоорганізується) та інші задачі попереднього аналізу даних.

– 1986 р. – Девідом І. Румельхартом, Дж. Є. Хінтоном і Рональдом Дж. Вільямсом і одночасно з С. І. Барцевим та В. А. Охоніним (Красноярська група) перевідкритий та суттєво розвинений метод зворотного розповсюдження помилки. Почався вибух інтересу до нейронним мережам, що навчаються.

– 2007 р. Джеффри Хінтоном в університеті Торонто створено алгоритми глибокого навчання багат шарових нейронних мереж Успіх обумовлений тим, що Хінтон під час навчання нижніх шарів мережі використовував обмежену машину Больцмана (RBM – Restricted Boltzmann Machine).

Хоча спочатку в штучних нейронних мережах було багато досліджень та розробок, їхня популярність незабаром знизилася, а дослідження сповільнилися через технічні обмеження. Інтенсивність обчислень штучних нейронних мереж була надто складною для комп'ютерів у той час. Тоді комп'ютери не мали достатньої обчислювальної потужності, і підготовка нейронних мереж зайняла б занадто багато часу. В результаті інші методи машинного навчання стали більш популярними, і штучними нейронними мережами здебільшого знехтували.

Однак один важливий алгоритм, пов'язаний зі штучними нейронними мережами, був розроблений у цей час – зворотне розповсюдження, відкрите Полом Вербосом. Зворотне розповсюдження – це спосіб навчання штучних нейронних мереж, намагаючись мінімізувати помилки. Цей алгоритм дозволив вченим набагато швидше тренувати штучні мережі.

Штучні нейронні мережі знову стали популярними в кінці 2000-х років, коли такі компанії, як Google і Facebook, показали переваги використання методів машинного навчання на великих наборах даних, зібраних від повсякденних користувачів. У наш час ці алгоритми в основному використовуються для глибокого навчання, що є областю машинного навчання, яка намагається моделювати відносини, які є більш складними, наприклад, нелінійні відносини [5].

1.3.1 Перцептрон

Перцептрон – це математична модель сприйняття інформації мозком, розроблена Френком Розенблаттом у 1957 році. Перцептрон став однією з перших моделей нейромереж [8].

Перцептрон приймає на вхід кілька бінарних значень та виробляє єдиний бінарний вихід.

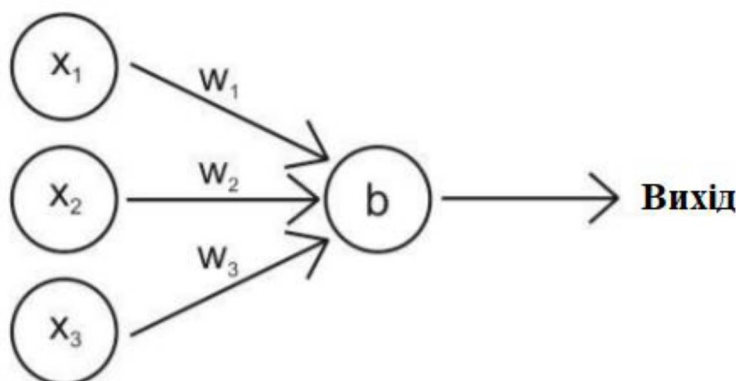


Рисунок 1.3 – Одношаровий перцептрон

На рис. 1.3 перцептрон отримує на вхід три елементи x_1, x_2, x_3 . Результат обчислюється за формулою (1.1):

$$g(z) = g(\sum_{i=1}^3 w_i x_i + b) = f(x) = \begin{cases} 0, & \sum_{i=1}^3 w_i x_i + b \leq 0 \\ 1, & \sum_{i=1}^3 w_i x_i + b > 0 \end{cases} \quad (1.1)$$

де w_1, w_2, w_3 – ваги, дійсні числа, що виражають важливість кожного елемента входу, його вплив на вихід, b – зсув, граничне значенням залежно від якого змінюється вихід.

Перцептрон може використовувати ступінчасту функцію активації (рис. 1.4).

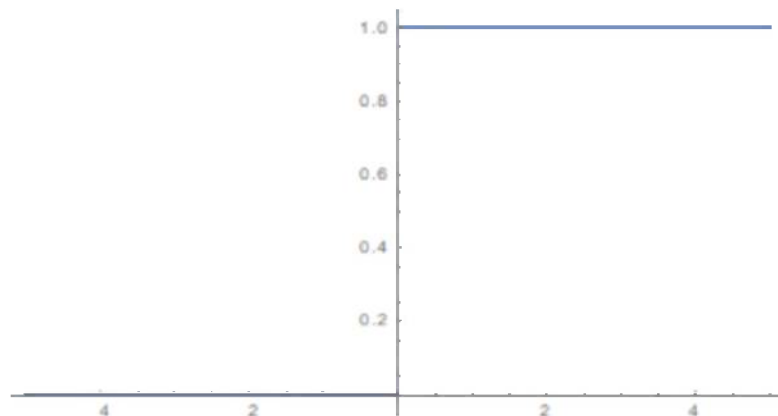


Рисунок 1.4 – Ступінчата функція активації

Вище (рис. 1.3) показаний лише один нейрон. Саму ж нейронну мережу можна подати об'єднання безлічі таких нейронів (рис. 1.5), де вихід одного нейрона є входом наступного, і нейрон на кожному

Наступний рівень залежить від нейронів на попередніх.

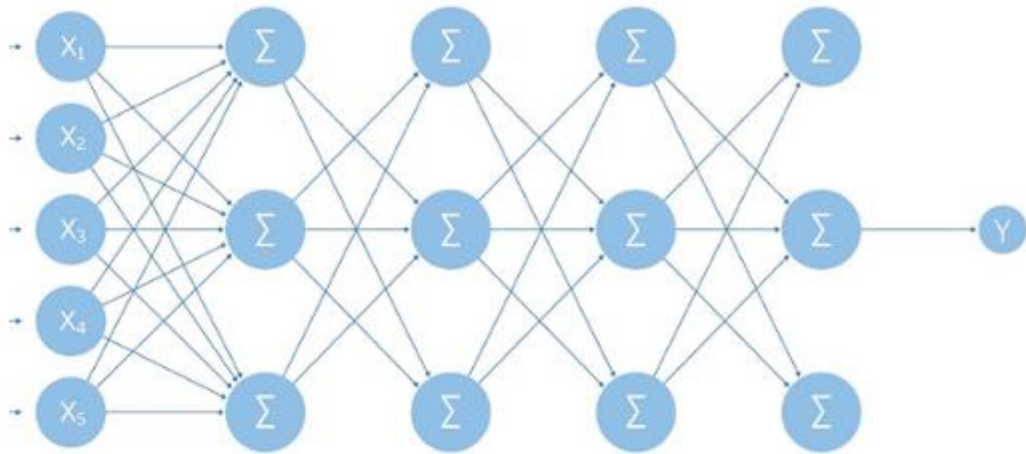


Рисунок 1.5 – Багатошаровий перцептрон

Приховані шари є структурою навчального алгоритму. Ціль нейронної мережі – наблизити деяку функцію f . Підібрати у процесі навчання такі ваги та зміщення, щоб максимально точно наближати необхідний результат. Введемо позначення, які використовуватимуться далі. Розглянемо нейронні мережі де вихід одного шару є входом наступного. Тобто в мережі немає циклів і інформація поширюється тільки в одному напрямку. Такі нейронні мережі називаються платою попереду. Також кожен нейрон деякого шару пов'язаний з усіма нейронами з попереднього шару. Кількість шарів нейронної мережі – L (перший шар входу не враховуємо).

Вхід – вектор стовпець $x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$, де n – кількість елементів входу. Ваги

та сполучні шар l і $l+1$ задаються матрицею $w_l = \begin{pmatrix} w_{11}^l & \dots & w_{1n}^l \\ \vdots & \ddots & \vdots \\ w_{m1}^l & \dots & w_{mn}^l \end{pmatrix}$, де n –

кількість нейронів шару $l+1$, m – кількість нейронів шару l . Тобто w_{ij}^l – вага між нейронами i та j шарів $l+1$ та l відповідно. Зміщення шару l задає вектор стовпець

$b^l = \begin{pmatrix} b_1^l \\ b_1^l \\ \dots \\ b_1^l \end{pmatrix}$, де n – кількість нейронів у шарі 1. Введемо позначення

$z^1 = w^1 * x + b^1$ щоб отримати значення першого прихованого шару, необхідно скористатися (1.1) або іншою функцією (наприклад, сигмоїдальною). Значення нейронів першого прихованого шару позначимо через $a^1 = g(z^1)$. Тоді обчислювати значення кожного наступного шару можна за формулою $a^1 = g(z^1) = (w^1 * a^{l-1} + b^1)$. Підсумковий вихід із нейронної мережі буде: $a^L = g(z^L)$

Загальний процес обчислення значень нейронної мережі можна представити так:

$$\begin{aligned} a^1 &= g(w^1 x + b^1), \\ a^2 &= g(w^2 a^1 + b^2), \\ &\dots \\ a^L &= g(w^L a^{L-1} + b^L), \end{aligned} \tag{1.2}$$

де g – деяка функція, що перетворює один шар на інший. Це може бути як перцептрон, так і інша функція. Перцептрон – найпростіша модель нейрона, до його недоліків відносяться розривність та бінарність. Так невелика зміна ваги може позначитися на зміні виходу в протилежний бік. Перцептрон було наведено як історичну довідку першого штучного нейрона.

1.3.2 Згорткові нейронні мережі

Згорткові нейронні мережі є одним з найбільш важливих класів теорії глибокого машинного навчання для вирішення задач комп'ютерного зору. Згорткові нейронні мережі складаються з декількох шарів обробки, які як правило чергуються, кожен з шарів містить, як лінійні, так і нелінійні оператори.

Сьогодні використання згорткових мереж є одним з основних методів для вилучення ознак з аудіо, відео і текстових даних.

Згорткова нейронна мережа (ЗНМ) складається з трьох основних видів шарів: згортковий шар, субдискредитуєчий шар і вихідний шар (найчастіше повнозв'язний) [6].

Шари ЗНМ розташовані один за одним: спочатку згортковий шар, а потім субдискредитуєчий, за останніми згортковим шаром слідує вихідний шар. Згорткові і субдискредитуєчі шари вважаються шарами двовимірної розмірності, а вихідний шар, як правило, являє собою вектор з простору $\mathbb{R}$. У ЗНМ кожен двовимірний шар має декілька рівнів. Кожен рівень являє собою двовимірний масив. Вихід кожного рівня називають картою ознак.

Приклад архітектури згорткової нейронної мережі представлений на рис. 1.6.

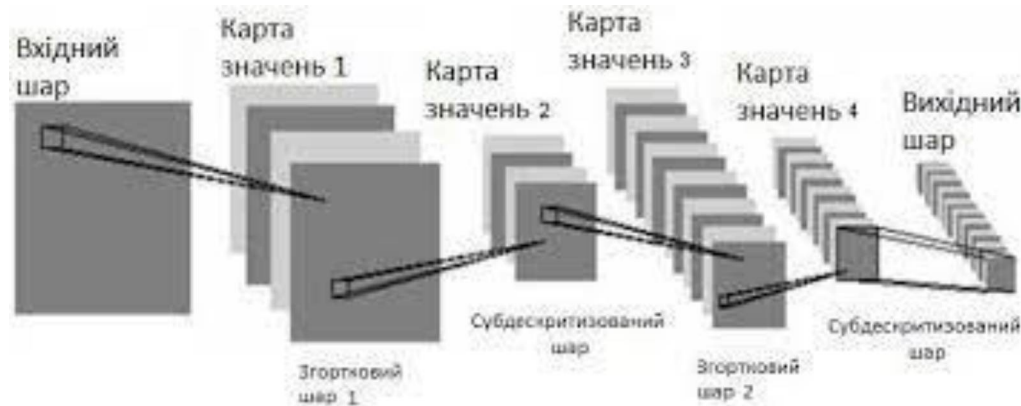


Рисунок 1.6 – Архітектура згорткової нейронної мережі

Основною ідеєю використання згорткового шару є застосування математичної операції згортки (фільтра) до зображення [7]. Згортка – це двовимірна матриця коефіцієнтів. Вхід такого фільтра – це фрагмент двовимірного зображення, а вихід – деяке число (рис. 1.7). Перевага використання подібного роду фільтрів полягає в наступному: число на виході тим

більше, чим більше елемент зображення схожий на застосований до нього фільтр. Отже, використання операції згортки допомагає отримати на виході зображення, кожен піксель якого буде відповідати ступеню подібності шматочка зображення на фільтр. Іншими словами, ми отримаємо карту ознак.

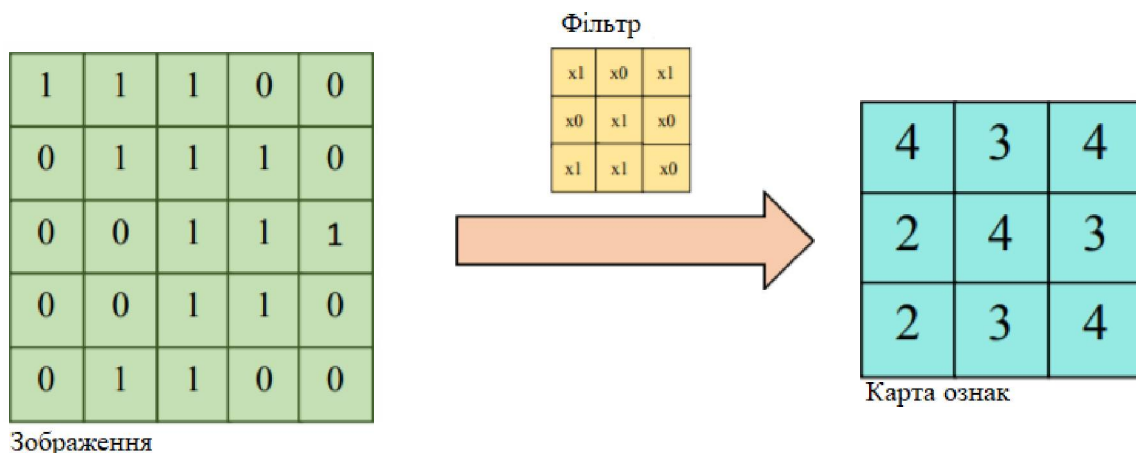


Рисунок 1.7 – Приклад використання двовимірної операції згортки

Субдискредитизація застосовується для зменшення загального розміру зображення і збільшення ступеня інваріантності застосовуваних до нього фільтрів. При розгляді архітектури ЗНМ будемо спиратися на факт, що наявність якої-небудь ознаки на зображенні набагато важливіше, ніж точне знання її координат. Таким чином, суттю субдискредитизації є вибір максимального нейрона з декількох сусідніх. Потім, цей нейрон приймаємо за елемент наступної, але вже зменшеної карти ознак. Дана операція допомагає збільшити інваріантність до масштабу вхідного зображення.

Таким чином, ЗНМ будується за допомогою чергування шарів згортки і субдискредитизації. Подібна архітектура нейронної мережі дозволяє будувати карти ознак з карт ознак, а це означає, що ми отримали можливість розпізнавати ієрархії ознак. Найчастіше, на практиці, в кінці роботи ЗНМ карти ознак

перетворюються в вектор або скаляр. Звідси виникла потреба в додаванні повнозв'язного шару на виході.

Повнозв'язний шар є найпростішим і широко застосовуваним шаром нейронної мережі. Кожен нейрон в цьому шарі – перцептор з нелінійною функцією активації. Як функції активації прийнято використовувати або логістичну функцію виду: $f(x) = \frac{1}{1+e^{-x}}$, або гіперболічний тангенс: $f(x) = A * \tanh(Bx)$, де $A, B = const$. Кожен нейрон повнозв'язного шару з'єднується з виходами нейронів попереднього шару, підсумовує свої входи, примножує на ваги, додає поріг і результат подає на вхід функції активації. Отримане значення є виходом нейрона.

1.3.3. Рекурентна нейронна мережа

Рекурентна нейронна мережа (RNN) – це вид нейронної мережі прямого поширення, яка має внутрішню пам'ять. RNN є рекурентною по своїй природі, оскільки виконує одну і ту ж функцію для кожного вводу даних, у той час, як виведення поточного вводу залежить від останнього результату. Після створення виводу він копіюється та відправляється назад у поточну мережу. Для прийняття рішень мережа виконує поточний ввід і вивід, який дізнається з попереднього вводу.

Приклад архітектури рекурентної нейронної мережі наведений на рис. 1.8.

На відмінну від нейронних мереж прямого поширення, RNN може використовувати свій внутрішній стан (пам'ять) для обробки послідовностей вхідних даних. Це робить їх незмінними для таких завдань, як несегментоване пов'язане розподілення рукописного вводу або розподілу речей. У інших нейронних мережах всі входи незалежних друг від одного, тоді як у RNN всі входи пов'язані з іншим.

Спочатку мережа бере $X(0)$ від послідовності введення, а потім виводить $h(0)$, який разом з $X(1)$ є входом для наступного кроку. $h(0)$ і $X(1)$ – це входні дані для наступного кроку. Точно так же $h(1)$ з наступного є входом з $X(2)$ для наступного кроку і так далі. Таким чином, він постійно реєструє контекст часу навчання [9].

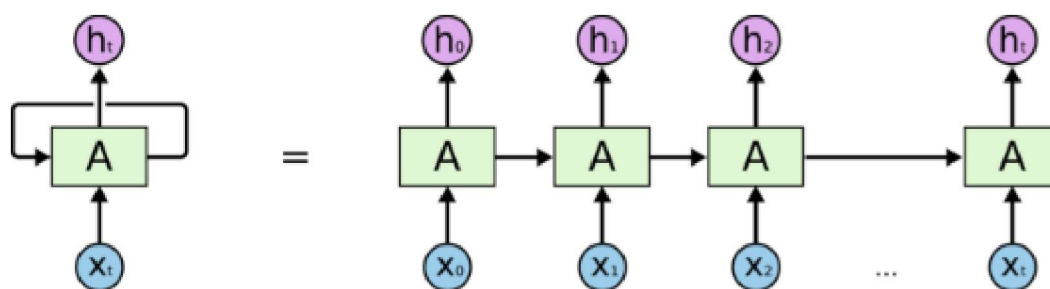


Рисунок 1.8 – Архітектура рекурентної нейронної мережі з одним входним, прихованим і вихідним нейроном

1.4 Базові алгоритми навчання нейронних мереж

1.4.1 Градієнтний спуск

Завдання навчання нейронної мережі – мінімізація функції вартості, тобто. мінімізація функції багатьох змінних. Оскільки функція, яку необхідно мінімізувати досить складна і зазвичай має дуже велику кількість параметрів, звичайний спосіб пошуку екстремуму функції не підійде. Спочатку потрібно підрахувати похідні першого порядку, вирішити систему рівнянь і знайти точки підозрілі на екстремум. Потім для того щоб перевірити, яка з точок є локальним мінімум, потрібно обчислити похідні другого порядку. Такий спосіб дуже затратний за обчисленням і заскладний. Тому алгоритми мінімізації функції, що застосовуються у машинному навчанні, в основному базуються на градієнтному спуску.

Розглянемо алгоритм градієнтного спуску. Для простоти обчислень оберемо функцію вартості. Наше завдання полягає в мінімізації функції багатьох змінних, ваги і зсуви.

Визначення 1. Глобальний мінімум функції $f: X \rightarrow \mathbb{R}, x_0 \in \min f \Leftrightarrow$

$$\forall x \in X \text{ виконується } f(x) \geq f(x_0)$$

Визначення 2. Локальний мінімум функції $f: X \rightarrow \mathbb{R}, x_0 \in \text{loc min } f \Leftrightarrow$

$$\exists O \in (x_0), \text{ що } \forall x \in O \cap (x_0) \text{ виконується } f(x) \geq f(x_0)$$

Визначення 3. Градієнтом функції $f: \mathbb{R}^n \rightarrow \mathbb{R}$ називається вектор приватних похідних.

Визначення 4. Похідна функції $f: \mathbb{R}^n \rightarrow \mathbb{R}$ у напрямку e називається проекцією градієнта цей напрям.

Нехай завдання полягає у мінімізації функції $f: \mathbb{R}^n \rightarrow \mathbb{R}$.

Для мінімізації функції f необхідно знайти напрямок, у якому f зменшується найшвидше. Тобто необхідно знайти $\min_e \nabla f = \min \|e\| \nabla f \cos \theta$, вважаючи довжину вектору постійною, вираз зведеться до $\min \cos \theta$, $\theta = \pi$ тобто. значення похідної у напрямку мінімально якщо її напрямок протилежний напрямку вектору градієнта.

Таким чином можна зменшувати функцію у напрямку зворотного градієнту за таким правилом:

$$x'_j = x_j - \eta \frac{\delta f}{\delta x_j}, \quad (1.3)$$

де η – learning rate, швидкість навчання, позитивне число, що задає розмір кроку, з яким задана функція зменшується. Зазвичай за η вибирається невелике позитивне число. Так можна зменшувати значення функції, застосовуючи це правило кілька разів поспіль. Перенесемо (1.3) на функцію вартості, отримаємо:

$$\begin{aligned}
 w'_i &= w_i - \eta \frac{\partial C}{\partial w_i} \\
 b'_i &= b_i - \eta \frac{\partial C}{\partial b_i}
 \end{aligned}
 \tag{1.4}$$

Так як функція вартості усереднена по всіх входах, то для обчислення нових значень доведеться обчислювати градієнт ∇C для кожного входу, а потім усереднювати його. Іноді кількість таких входів може бути більшою. І таке обчислення займе багато часу. Тому зручніше проводити обчислення градієнта на випадкових невеликих наборах даних (нам не потрібно знати точне зменшення функції, нам головне рухатися в потрібному напрямку). Такі входи називають mini batch. При цьому вважають, що значення градієнта на таких партіях приблизно таке саме, як і на всіх даних. Такий алгоритм навчання називається стохастичним градієнтним спуском.

Наприклад, вся множина вхідних даних X випадково розбивається на кілька підмножин: $X_1, X_2, \dots, X_k$, нехай при цьому розмір кожного підмножини m . Далі алгоритм тренується на кожній з цих підмножин, тобто.

$$\begin{aligned}
 w'_i &= w_i - \eta \frac{\partial C_{X_j}}{\partial w_i} \\
 b'_i &= b_i - \eta \frac{\partial C_{X_j}}{\partial b_i}
 \end{aligned}
 \tag{1.5}$$

де C_{X_j} – функція усереднена за набором X_j . Значення вагів та зсувів будуть змінюватися після кожного тренування на наборах X_j . Набір тренувань з усіх партій даних X_j називається епохою. Коли одна епоха закінчилася дані знову поділяються на підмножини і відбувається навчання кожному з підмножин, тобто починається наступна епоха.

Звернімо увагу на можливі проблеми градієнтного спуску. На рис. 1.9 представлена функція однієї змінної з кількома мінімумами.

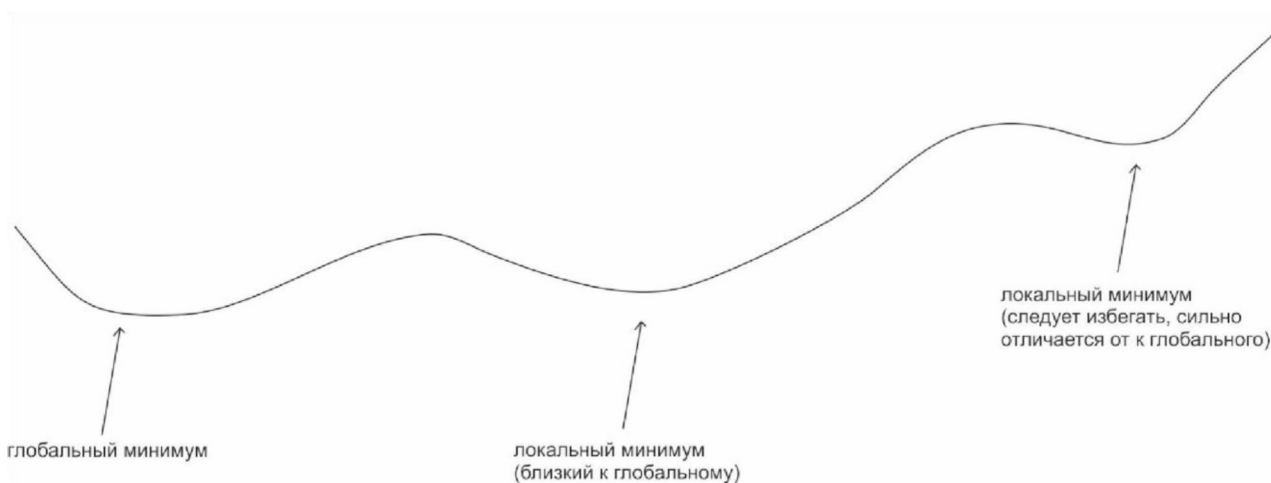


Рисунок 1.9 – Локальний мінімум

Якщо під час роботи алгоритму градієнтного спуску початкове положення значення ваги і зсуву було обрано поруч із однією з локальних мінімумів, існує ймовірність того, що можна вибратися з цього локального мінімуму. І якщо значення цього локального мінімуму сильно відрізняється від глобального, то результат роботи нейронної мережі буде далекий від бажаного.

При глибокому навчанні мінімізується функція багатьох змінних, яка має безліч локальних мінімумів, сідлових точок, оточених плоскою областю. Це робить оптимізацію досить складною. Тому зазвичай погоджуються на пошук значення функції, що є дуже низьким, але не обов'язково є глобальним мінімумом.

1.4.2 Алгоритм зворотного розповсюдження помилки

Якщо алгоритм градієнтного спуску служить для мінімізації функції, то алгоритм зворотного поширення помилки призначений для швидкого обчислення градієнта. Для обчислення градієнта необхідно обчислювати окремі похідні: $\frac{\partial C}{\partial w_j^i}, \frac{\partial C}{\partial b_j^i}$. Обчислювати їх щоразу вручну важке завдання, оскільки

нейронна мережа може бути глибокою і мати безліч параметрів. При цьому необхідно робити це швидко. Для швидкого обчислення градієнта функції вартості придумали алгоритм зворотного поширення помилки. Метод був вперше придуманий у 1970-і роки, далі суттєво розвинений у 1986 р. Опис присутній у [10, 11]. Розглянемо далі.

Символом $\odot$ позначатимемо операцію Адамара.

Визначення 5. Операція Адамара – операція над двома матрицями однакової розмірності, результатом якої є матриця такої ж розмірності. Значення елемента з індексом ij виходить, як добуток елементів з таким самим індексом вихідних матриць.

Як функцію вартості використовуємо середню квадратичну помилку, як функцію активації сигмоїд. Введемо помилку нейрона j шару l :

$$\delta_j^l = \frac{\delta C}{\delta z_j^l}. \quad (1.6)$$

Тоді загальний алгоритм зворотного розповсюдження помилки буде таким:

1. Розрахуємо активацію для кожного з шарів нейронної мережі:

$$a^l = \sigma(w^l a^{l-1} + b^l). \quad (1.7)$$

2. Обчислюємо вектор помилки останнього шару:

$$\delta^L = \nabla C \odot \sigma'(w^L a^{L-1} + b^L). \quad (1.8)$$

3. Розповсюджуємо помилку назад, для кожного шару $l=L, L-1, \dots, 2$

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(w^l a^{l-1} + b^l) \quad (1.9)$$

4. Обчислюємо градієнт функції вартості за формулами:

$$\frac{\delta C}{\delta b_j^l} = \delta_j^l, \frac{\delta C}{\delta w_{ij}^l} = a_j^l \delta_j^l. \quad (1.10)$$

Алгоритм називається зворотним поширенням помилки, оскільки помилка поширюється тому, перші верстви мережі. Обчислюють помилку з останнього шару і закінчуючи першим. І обчислюють всі необхідні параметри нейронної мережі за два проходи.

Висновок за розділом 1

Для розпізнавання об'єктів на зображеннях існує багато видів класифікаторів.

Різні види класифікаторів мають свої переваги і недоліки. Так, класифікатори, в яких використовуються методи статистики мають хорошу математичну обґрунтованість, але при цьому складні у використанні і вимагають знання ймовірного розподілу вихідних даних і оцінки його параметрів. Крім цього, статистичні методи оцінюють тільки ймовірність приналежності об'єкта класу, але не «пояснюють» чому.

Класифікатори, засновані на машинному навчанні не вимагають оцінки параметрів розподілу вихідних даних, а міра подібності в них формалізується за допомогою функції відстані. Такі класифікатори називаються метричними. Як правило, вони простіше в реалізації і використанні, ніж параметричні, а їх результати зручніше для інтерпретації і розуміння. Але при цьому метричні класифікатори є наближеними моделями — забезпечують рішення тільки в

обмеженому числі практично значущих випадків, можуть дати неточне або не єдине рішення.

Певним компромісом між параметричних і метричними методами є використання для вирішення завдань класифікації нейронних мереж. Дійсно, НМ є непараметричних моделями, що не вимагають припущень про імовірнісний розподіл даних, але при цьому і не використовують заходи відстаней.

РОЗДІЛ 2. РОЗРОБКА МОБІЛЬНОГО РОБОТА ДЛЯ СПОСТЕРЕЖЕННЯ

Для тестування та оцінки роботи системи в реальних умовах потрібно побудувати її прототип. У цьому розділі було проаналізовано існуючі мобільні роботи, та детально розглянуті компоненти прототипу мобільного робота для спостереження та реалізація програмного забезпечення.

2.1 Загальні відомості про мобільних роботів

Оскільки в різних джерелах інформації немає однозначних визначень та термінів щодо мобільних роботів і загальноприйнятої класифікації, спочатку необхідно запровадити кілька визначень.

Насамперед, загальне поняття "робот" трактується окремими авторами по-різному. Найбільш відповідним є таке визначення: робот – машина з автоматичним чи інтерактивним керуванням, що перепрограмується або самонавчається; маніпуляційна чи мобільна; має кілька ступенів рухливості; використовується як у детермінованих, так і в недетермінованих середовищах та процесах.

У запропонованому визначенні можна назвати такі особливості:

- до роботів віднесені системи з інтерактивним управлінням (тобто керовані змінно – то оператором, то автоматично), це дозволяє вважати роботами не тільки системи, оснащені маніпуляторами, але і великий клас пристроїв, що з'явилися останнім часом, званих мобільними роботами, які в більшості випадків мають саме інтерактивне управління (автономні мобільні роботи, тобто керовані повністю автоматично, існують поки що тільки в лабораторних розробках);

– до роботів не відносять пристрої з ручним керуванням, це, звичайно, не дозволяє віднести до роботів частину мобільних систем з дистанційним ручним керуванням (керованих протягом усього часу тільки оператором і не мають автоматичного керування), але якщо не прийняти це обмеження, то доведеться відносити до роботів усі маніпулятори з ручним управлінням (а ними можна підрахувати навіть порталні крани, екскаватори тощо), що абсолютно неприйнятно, необхідно, однак, враховувати, що в джерелах інформації мобільні системи з ручним керуванням все ж таки часто називаються мобільними роботами.

Під мобільними роботами пропонується розуміти лише пристрої, переміщення яких визначаються невизначеною (заздалегідь не прокладеною), необмеженою траєкторією.

До роботів пропонується відносити тільки машини, що володіють декількома ступенями рухливості. З тим, що історично склалося вважати машини з одним ступенем рухливості просто транспортними засобами.

Зазначаючи, що мобільні роботи можуть застосовуватися як у детермінованих, так і в недетермінованих середовищах (процесах), підкреслюється, що сфера використання роботів необмежена (на відміну, наприклад, від промислових роботів).

Термін "мобільний робот" має у різних джерелах інформації такі визначення:

- рухомий робот [12];
- робот, який встановлений на рухомому візку [13].

Такі визначення є визначеннями з неоднозначними ознаками (критеріями). Справа в тому, що в новому класі машин, що з'явився останнім часом і названий мобільними роботами, є не просто переміщення робота на певну відстань, а переміщення робота на колесах, гусеницях або ногах на будь-яку задану людиною або програмою відстань. Це передбачає відсутність однозначного,

наперед прокладеного маршруту переміщення (траєкторії). Такі роботи зазвичай мають розвинену сенсорну систему, елементи штучного інтелекту. Ознака як більшої чи меншої незалежності від людини-оператора, а й незалежності від однозначної, попередньо визначеної траєкторії міститься у самій назві таких роботів. Тобто автономні мобільні роботи – це роботи, керовані без участі людини.

В результаті сказаного вище пропонується таке визначення: мобільний робот (англ. Mobile robot; locomotive robot) – робот, відмінною рисою якого є наявність локомоційної здатності (тобто здатності до переміщувальних рухів системи в просторі).

Існують різні підходи до класифікації мобільних роботів.

На рис. 2.1 наведено класифікацію мобільних роботів за сферами їх застосування.

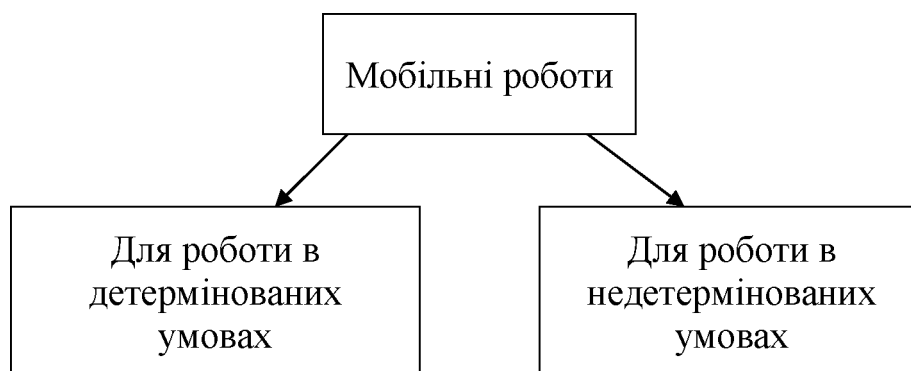


Рисунок 2.1 – Класифікація мобільних роботів по сферах застосування

Необхідно відзначити, що для роботи в недетермінованих умовах можуть в переважній більшості випадків використовуватись тільки мобільні роботи. Недетерміновані умови – це недетерміновані середовища або недетерміновані процеси.

До недетермінованих середовищ пропонується відносити природні середовища та середовища, створювані аварійними ситуаціями, як у природних умовах, так і при руйнуваннях будівель та споруд, спроектованих та створених людиною. До дій робота в природних середовищах відносяться дії при гірничих розробках, польових сільськогосподарських роботах, лісопосадках, розвідці на місцевості, підводних і підземних роботах, визначення радіоактивної зараженості місцевості, військових діях при розмінуванні та патрулюванні, обслуговуванні сліпих як поводитирів на вулиці тощо.

До дій робота при руйнуванні створених людиною середовищ відносяться дії з розчищення завалів, рятувальних роботах у зруйнованих спорудах тощо.

До недетермінованих процесів належатиме кожен процес, перебіг і результат якого не залежить повністю від цілеспрямованої діяльності людини. Недетермінованими процесами є всі природні процеси (землетруси, виверження вулканів тощо), пожежі, вибухи, збирання сільськогосподарських продуктів тощо.

Слід зазначити, що у детермінованих умовах мобільні роботи використовуються рідко. Детерміновані умови – це або детерміновані середовища та (або) детерміновані процеси.

До детермінованих середовищ відносять середовища, що проектується та створюються людиною (виробничі, побутові приміщення та будівлі, будівельні майданчики та інші споруди). До детермінованих процесів належати кожен процес, перебіг і результат якого повністю залежить від цілеспрямованої діяльності людини (діяльності з безпосереднього здійснення процесу, управління процесом тощо)

Наприклад використання мобільних роботів в детермінованих умовах відноситься, наприклад, обслуговування атомних електростанцій та інших споруд, перебування в яких небезпечно для людини; обслуговування людей у побуті як роботи-прибиральниці, роботи-доглядальниці тощо.

За рівнем незалежності від участі людини-оператора в контурі управління всі мобільні роботи можуть бути класифіковані на великі групи [14]:

- з автоматичним керуванням (автономні роботи);
- з інтерактивним керуванням (телеоператори, телероботи).

Мобільні роботи з автоматичним керуванням – роботи, керування якими здійснюється системою керування без участі людини.

Для того щоб здійснювати дії без участі людини (зазвичай у недетермінованому, змінному середовищі) роботи повинні мати штучний інтелект. У зв'язку з цим, на відміну інших роботів, мобільні роботи з автоматичним управлінням у джерелах інформації зазвичай називаються автономними. Автономні мобільні роботи – це роботи, здатні самі (залежно від сигналів від датчиків) приймати рішення в обстановці, що змінюється. Іноді вони називаються також мобільними роботами, що самонавчаються.

Роботи з інтерактивним управлінням слід визнати системами вищого інтелектуального рівня, ніж автономні роботи, оскільки можливості людського розуму лише на рівні прийняття рішення у критичних ситуаціях перевершують нині найрозвиненіші системи штучного інтелекту.

Мобільні роботи з інтерактивним управлінням – це роботи, які можуть працювати, і в автоматичному режимі, і керуватися людиною-оператором. На відміну від біотехнічних систем (систем з ручним управлінням), інтерактивні системи мають пристрої пам'яті автоматичного виконання певних дій [15].

За функціональним призначенням мобільні роботи класифікуються на такі групи:

- спеціального призначення;
- для військових та воєнізованих застосувань;
- для екстремальних ситуацій, наукових досліджень;
- для спортивних, промислових та побутових застосувань.

2.2 Структура мобільного робота

Будь-який мобільний робот (МР) може бути представлений у вигляді сукупності трьох великих систем – транспортної, спеціальної та керування [16].

Транспортна система являє собою транспортний засіб, призначений для доставки спеціального та технологічного обладнання до місця виконання поставленого завдання.

Транспортний засіб складається з ходової частини, корпусу та енергетичної установки. Як правило, система керування встановлюється всередині корпусу. Залежно від типу середовища експлуатації ходова частина може бути гусенична, колісна, колісно-гусенична, напівгусенична, крокуюча, колісно-крокова, роторна.

Зовнішність наземного мобільного робота в першу чергу визначається типом і конструкцією рушія, що служить для перетворення в процесі взаємодії із зовнішнім середовищем зусилля, одержуваного від двигуна, в тягове зусилля, рушійний транспортний засіб.

Вибір типу рушія та його розмірів є дуже складним завданням. Практично неможливо створити універсальну конструкцію рушія, що дає можливість однаково впевнено пересуватися у різноманітних умовах довкілля. Безліч видів та властивостей основ, складні перетину рельєфу місцевості, необхідність переміщення по елементах споруд і всередині будівель є причиною створення великої кількості схем компонування роботів з різними типами рушіїв.

Основна увага розробників приділяється різним варіантам колісного та гусеничного рушіїв. Дещо менша увага приділена рушійному рушію. І суттєво менше – іншим типам (наприклад, роторно-гвинтовому, апаратам на повітряній подушці та ін.).

Для кожного типу рушія існує своя сфера застосування. Так, як рушій багатофункціонального мобільного робота, призначеного для використання на

важкопрохідній місцевості, вибирають гусеничний рушій як найбільш універсальний. При переважному використанні робота на дорогах кращим є колісний варіант транспортного засобу. Застосування крокуючих машин перспективне лише в середовищі, де швидкість колісного або гусеничного рушія поступається швидкості рушійного рушія (наприклад, в гірській місцевості, в осередках руйнувань і т.п.). При конструюванні звичайних транспортних засобів параметри рушія оптимізуються найбільш характерних умов застосування і поверхонь руху. Однак, для мобільного робота така оптимізація неможлива через невизначеність умов руху. Тому в даний час рушії роботів конструюються з можливістю адаптації до поверхні руху.

Спеціальні системи служать безпосереднього виконання поставлених завдань. Спеціальна система складається з необхідного набору технологічного обладнання, склад якого визначається видом задачі, що вирішується, і призначенням МР.

Система управління забезпечує управління рухом та роботою технологічного обладнання, а також адаптивне керування ходовою частиною та енергетичною установкою з урахуванням взаємодії транспортної системи з навколишнім середовищем.

Система управління рухом повинна також забезпечувати планування руху в недетермінованих умовах на основі картографічної бази, з урахуванням безперервно надходить інформації в систему управління від технічних органів чуття та навігаційної системи.

Складність системи управління визначається складністю розв'язуваної задачі, ступенем невизначеності довкілля і необхідною ступенем автономності робота.

Саме розвиток систем управління визначає розвиток робототехнічних комплексів загалом, і, зокрема, є основою класифікації мобільних роботів по поколінням. Система управління містить три рівня управління: верхній

(стратегічний), середній (тактичний) і нижній (виконавчий), які мають вбудовані механізми адаптації, що працюють на основі оцінки якості реалізації планів різного рівня в реальному фізичному світі. Організація взаємодії рівнів управління повинна дозволяти приймати рішення на тому рівні, який на даний момент має найдостовірнішу інформацію, без передачі управління на більш високий рівень.

Людина (оператор) нині є невід'ємною частиною системи управління. Функції людини у системі управління визначають її складність.

У роботах першого покоління оператор бере активну участь в управлінні мобільним роботом на всіх трьох рівнях, аж до безперервного ручного управління виконавчими механізмами. Це полегшує конструкцію системи управління, але ускладнює роботу оператора. Основні недоліки дистанційного керування обумовлені наявністю телевізійного та радіоканалів зв'язку, їх невисокою схибленістю, неможливістю зберігати режим радіомовчання, небезпекою несподіваного припинення зв'язку в зонах радіотіні.

У роботах другого покоління управління нижнього рівня покладено на бортову систему управління роботом. Спільним для роботів другого покоління є використання зворотного зв'язку як відповідно до поточного стану робота, так і відповідно до стану довкілля.

Третє покоління роботів залишає людині лише стратегічний рівень: система спілкування з оператором зводиться до видачі завдання та прийняття звіту про його виконання. Платити за полегшення життя оператора доводиться дуже дорого: автоматична система повинна мати універсальність, гнучкість і широту можливостей природного інтелекту. При цьому будь-яка розв'язувана системою штучного інтелекту додаткова задача, або тим паче клас завдань чи ситуацій, вимагає як розробки спеціальних алгоритмів рішення, так і спеціалізованих технічних засобів – нових технічних органів чуття,

спецобчислювачів і виконавчих органів. Тобто кожне таке завдання є складною науково-технологічною проблемою.

2.3 Мобільні роботи спеціального призначення

Мобільні роботи спеціального призначення застосовуються під час проведення вибухотехнічних робіт та антитерористичних операцій, а також під час охорони важливих об'єктів.

При цьому застосування роботів можливе для вирішення наступних тактичних завдань [17]:

– під час проведення вибухотехнічних робіт:

- 1) пошук та діагностика вибухових пристроїв;
- 2) знищення чи евакуація вибухових пристроїв;
- 3) розпорядження чи знешкодження вибухових пристроїв;
- 4) проведення хімічної та радіаційної розвідки об'єктів та територій;

– під час проведення антитерористичних операцій:

- 1) постановка радіоелектронних перешкод, димових та спеціальних завіс;
- 2) доставка та застосування спецзасобів нелетальної дії;
- 3) приховане проникнення на захоплені та охоронювані об'єкти;
- 4) ведення радіоелектронної аудіо- та відеорозвідки об'єктів та територій;
- 5) руйнування перешкод (двері, стіни);
- 6) ведення відволікаючого вогню, виявлення вогневих точок супротивника;

– при охороні об'єктів:

- 1) патрулювання території чи периметра об'єкта;
- 2) припинення спроб проникнення на об'єкт;

3) нейтралізація порушників.

Зазначені операції проводяться на різних об'єктах та в різноманітних умовах:

- на об'єктах громадського транспорту (міський транспорт, залізничний, авіаційний, морський, автомобільний);
- у місцях проживання та життєдіяльності людей (квартири, будинки, офіси та ін.);
- на промислових об'єктах (об'єкти хімічної промисловості, ядерного технологічного циклу та ін.);
- на об'єктах міської інфраструктури (каналізація, теплостанції, водопровід тощо);
- на відкритій місцевості, сильно пересіченій місцевості, у лісах тощо.

Специфіка операцій, умови експлуатації та функціональне призначення мобільного робота визначають його конструктивні особливості, ступінь складності системи керування, масогабаритні характеристики та склад спеціального обладнання.

До мобільного робота пред'являються такі загальні вимоги:

- робот повинен мати високі рухливість і прохідність у міських умовах, усередині будівель та споруд, у зонах руйнувань, на пересіченій місцевості, як на твердих гладких покриттях, так і на ґрунтових основах, що деформуються;
- робот повинен надійно діяти як у непередбачених природних умовах, так і в середовищі, спеціально пристосованому для проживання людини (всередині будинків, у транспортних комунікаціях), вписуватися в міські транспортні потоки або рухатися у складі транспортних колон;
- конструкція робота повинна забезпечувати його високу мобільність та швидке розгортання при виконанні спецоперацій.

Для виконання вищевказаних завдань спецпідрозділи мають такі основні групи мобільних роботів:

- мобільний робототехнічний комплекс (МРК) – універсальні наземні роботи, призначені для дій на об'єктах транспорту, промисловості, міської інфраструктури і т.д., на відкритій слабопересіченій місцевості;
- малогабаритний дистанційно-пілотований літальний апарат (МДПЛА)
- повітряний робот для проведення розвідки на відкритій місцевості, сильно пересіченій місцевості, в горах, у місті;
- спеціальні робототехнічні комплекси – роботи, здатні переміщатися вертикальними та похилими поверхнями промислових об'єктів та транспортних засобів, а також у трубопроводах та вузьких місцях.

Мобільні робототехнічні комплекси застосовуються при:

- бойових забезпеченнях спецоперацій (загороджувальний вогонь, розвідка боєм, руйнування загороджень тощо);
- проведення розвідки;
- проведення вибухотехнічних робіт (пошук, вилучення, транспортування та знешкодження або знищення вибухонебезпечних предметів та боєприпасів, що не розірвалися);
- забезпечення безпеки важливих об'єктів.

За масою (і, отже, мобільності) та основним призначенням МРК можна розділити на 4 групи:

- надлегкі, масою до 35 кг (рис. 1.2);
- легкі, масою до 150 кг;
- середні, масою до 800 кг;
- тяжкі, масою понад 800 кг.

Мобільний робот МРК-01 призначений для проведення інспекційних перевірок, пошуку та знищення вибухонебезпечних предметів, має масу 20 кг,

тому належить до класу надлегких МРК. Він має наступні технічні характеристики: габаритні розміри – $0,57 \times 0,48 \times 0,21$ м; швидкість пересування – 0-2,5 км/год; трансмісія – електромеханічна; рушій – колісний.

У світовій практиці найбільшого розвитку отримали надлегкі, легкі та робототехнічні комплекси перших трьох груп. Це зумовлено їх маневреністю, можливою швидкою технічною адаптацією до конкретного виду проведеної операції або виконуваних робіт, а також відносно невеликими матеріальними та економічними витратами на їх виробництво та експлуатацію. Основне призначення цих роботів – охорона приміщень важливих об'єктів, боротьба з терористичними акціями, пошук та знешкодження вибухонебезпечних предметів.

Спочатку закладений у конструкцію більшості роботів модульний принцип дозволяє створювати багатофункціональні комплекси, використовуючи єдину транспортну систему як базову та формуючи робочу систему при встановленні змінного озброєння або робочого обладнання та необхідної системи управління.

Для роботи масою до 800 кг розробляються оригінальні спеціалізовані транспортні модулі. Так робот МРК-25, показаний рисунку 2.2, має конвертовану ходову частину. Складання гусеничного обводу дає можливість роботу маневрувати у стиснених умовах (наприклад, розгортатися на сходових майданчиках) та забезпечує перевезення робота в джипі або мікроавтобусі. Цей робот має масу 186 кг; габаритні розміри – $0,95 \times 0,65 \times 0,9$ м; швидкість пересування – 0-1,1 км/год; радіус дії – 100 м; вантажопідйомність маніпулятора – 12 кг; зусилля захоплення захватного пристрою – 200 н.

Більш важкі робототехнічні системи використовують як базові шасі серійно випускаються зразки військової та цивільної транспортної техніки. Робот ETODS (США, BAT) виконано з урахуванням навантажувача типу “Bobcat” (рис. 2.3).



Рисунок 2.2 – Мобільний робототехнічний комплекс МРК-25
(МДТУ ім. М.Е. Баумана)



Рисунок 2.3 – Транспортний засіб роботизованої системи розмінування ETODS

Конструктивно універсальні мобільні роботи є малогабаритні самохідні засоби, що оснащуються розвідувальною апаратурою, набором змінного робочого обладнання та інструменту. Розраховані на дистанційне керування оператором, який веде спостереження безпосередньо або за допомогою

телевізійної камери. До складу встановлених на роботах комплексів приладів та обладнання входять:

- телевізійна апаратура (на сучасних зразках, як правило, кольорового зображення), що включає телевізійні камери (до чотирьох одиниць) та портативні монітори, за якими оператор веде спостереження за місцевістю та керує роботою машини;
- освітлювальні засоби (прожектори) для підсвічування при діях у темний час доби та низьких рівнях освітленості;
- маніпулятори для захоплення, переміщення та транспортування об'єктів;
- портативна рентгенівська апаратура для обстеження на місці об'єкта, що виявляється, і визначення ступеня його небезпеки;
- обладнання для знищення на місці вибухонебезпечних предметів (найбільшого поширення набули гідродинамічні руйнівники, що використовуються для знищення саморобних вибухових пристроїв у неметалевих оболонках, ацетиленові пальники для спалювання неметалічних мін та гладкоствольні рушниці для стрільби важкими кулями-болванками);
- набір інструменту для розбирання, відділення або виведення з ладу окремих компонентів виявленого боєприпасу з метою його нейтралізації;
- набір стетоскопів для прослуховування роботи годинникових механізмів підричників уповільненої дії, а також дзеркал для обстеження окремих компонентів підозрілого об'єкта, які знаходяться у важкодоступних місцях.

Самі машини виконуються на шасі з алюмінієвих сплавів та легованої сталі з колісною, гусеничною або змінною (швидко замінюваною з колісною на гусеничну та назад) ходовою частиною. На шасі змонтований повноповоротний (зазвичай) маніпулятор, пристосований для встановлення змінного робочого

обладнання, апаратури або інструменту. Як енергетична установка найчастіше служать електричні акумулятори, їх ємності зазвичай достатньо для роботи протягом декількох годин, проте можливе застосування двигуна внутрішнього згоряння або живлення від зовнішнього джерела електроенергії. При використанні акумуляторів привод ходової частини машини та робочого обладнання зазвичай електромеханічний, а двигуна внутрішнього згоряння – гідравлічний. Дистанційне керування роботою машин здійснюється по радіо (на відстані до 4000 м), по волоконно-оптичній лінії зв'язку (на відстані до 400 м), або по кабелю (на відстані до 100 м).

Малі маса і габарити дистанційно керованих машин допускають їх перевезення до місця виконання робіт легкими транспортними засобами, які вивантаження і навантаження виробляються легким апарелем своїм ходом. Низьке розташування центру ваги та наявність легких гусениць дозволяють машині долати круті підйоми та спуски, у тому числі сходові марші, проникати у невеликі приміщення та працювати на дуже обмеженій площі.

Малогабаритні дистанційно пілотовані літальні апарати (ДПЛА) з'явилися на початку 80-х років XX ст. [18]. Це був новий клас безпілотних розвідників. Піонерами в цій галузі стали ізраїльтяни, які першими створили і з великим успіхом застосували мініДПЛА в ході боїв із Сирією в долині річки Бекаа (Південний Ліван) у 1982 р. За Ізраїлем до робіт у цьому напрямі приступили СРСР, США, Великобританія, Франція, Італія, Канада, Китай, Ірак та інші країни, які мали розвинену авіапромисловість.

ДПЛА здатні:

- вести повітряну візуальну розвідку місцевості;
- вести радіаційну, хімічну та бактеріологічну розвідку;
- забезпечувати ретрансляцію радіосигналів;
- вражати цілі, доставляти до мети спеціальні технічні засоби.

В даний час відсутня єдина та чітка класифікація ДПЛА. Зокрема, прийнято підрозділяти безпілотні апарати за різними ознаками:

- залежно від застосовуваної системи управління – на політ за програмою або по радіокомандах (останні часто називаються дистанційно пілотованими, або телекерованими). Вважається, що дальність дії ДПЛА з радіокомандною системою наведення значно менше, ніж у апаратів, що здійснюють політ за програмою, оскільки управління ведеться, як правило, в УКХ діапазоні та визначається дальністю прямої видимості;

- за стартовою (злітною) вагою та розмірами – на малорозмірні (іноді їх ще називають мініатюрними), середньорозмірні та великорозмірні. Судячи з повідомлень іноземної преси, зараз основні зусилля західні фахівці спрямовують на створення малорозмірних ДПЛА, порівнянних за своїми ваговими та габаритними характеристиками з радіокерованими моделями літаків та вертольотів;

- за виконуваними завданнями – на розвідувальні, радіоелектронної боротьби та багатоцільові;

- за типом – ДПЛА літакового типу; вертолітного типу та автожири; підйомно-прив'язні системи; газонаповнені апарати (дирижаблі та аеростати).

До складу повітряних робототехнічних засобів входять дистанційно пілотовані літальні апарати (ДПЛА), кілька транспортних засобів, що забезпечують старт, обслуговування та керування.

Як силова установка на ДПЛА застосовуються авіаційні поршневі або турбореактивні двигуни, а для забезпечення їх старту (зльоту) з землі – порохові прискорювачі, що скидаються. Безпілотні апарати, що повертаються із завдання, спускаються на парашуті або підхоплюються в кінці глісади зниження спеціальною мережею, іноді вони роблять посадку за допомогою шасі.

Склад бортової апаратури визначається головним чином завданнями, що покладаються на ДПЛА: на розвідувальних використовуються аерофотоапарати

(АФА), телевізійні камери, інфрачервоні (ІЧ) станції, апаратура радіотехнічної розвідки. Апарати можуть нести на борту станції встановлення активних перешкод, пристрої викидання протирадіолокаційних відбивачів і т.д.

В останні роки в багатьох країнах підвищився інтерес до малогабаритних ДПЛА, що мають вертикальний зліт і посадку, виконаним за вертолітною схемою (рис. 2.4). Подібні МДПЛА характеризуються суттєво малою масою (до 50 кг) та габаритами, що дозволяє здійснювати їхнє перевезення неспеціалізованим транспортом та швидке розгортання на місці застосування.



Рисунок 2.4 – Вертоліт, що керується дистанційно

Коло завдань, розв'язуваних МДПЛА вертолітного типу, дуже широке:

- візуальний та технічний контроль великих важкодоступних територій;
- профілактичний контроль місцевості та об'єктів;
- пошук людей та об'єктів;
- доставка медикаментів та вантажів.

З середини 90-х у США ведуться розробки мініатюрних літальних апаратів – мікроДПЛА (рис. 2.5).

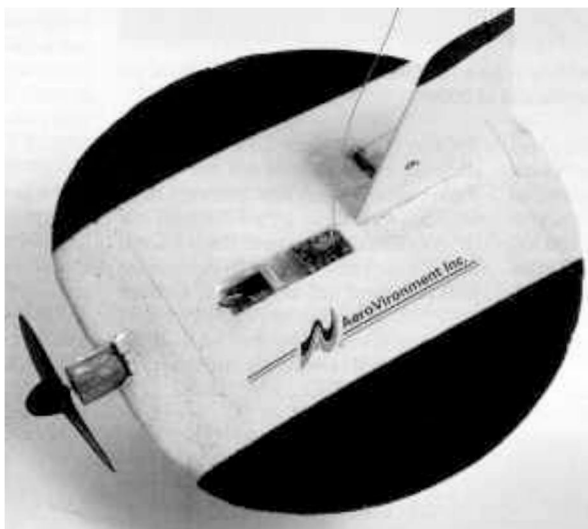


Рисунок 2.5 – МікроДПЛА (AeroVironment Inc., США)

МікроДПЛА являють собою дистанційно пілотовані пристрої з габаритами не більше 15 см, дальністю польоту близько 10 км, швидкістю 10-20 м/с і часом польоту до 1 год.

Як корисне навантаження використовуються мініатюрні цифрові камери, датчики, спеціальні технічні засоби.

2.4 Бойові роботи

Бойовий мобільний робот (або військовий робот) – автоматичний пристрій, що замінює людину у бойових ситуаціях для збереження людського життя або для роботи в умовах несумісних з можливостями людини у військових цілях: розвідка, бойові дії, розмінування тощо.

Бойовими роботами є не тільки автоматичні пристрої з антропоморфною дією, які частково або повністю замінюють людину, а й діючі в повітряному і водному середовищі, що не є місцем існування людини (авіаційні безпілотні з дистанційним керуванням, підводні апарати і надводні кораблі). Пристрій може бути електромеханічним, пневматичним, гідравлічним або комбінованим.

В даний час більшість бойових роботів є пристроями телеприсутності і лише деякі моделі мають можливість виконувати деякі завдання автономно, без втручання оператора [19].

Існують такі види бойових роботів:

- повітряні (БПЛА);
- мобільні робототехнічні комплекси;
- морські (надводні або підводні).

Загалом бойові БПЛА використовуються для ведення розвідки, завдання ударів по наземних цілях, коригування вогню тощо. Існують такі моделі БПЛА: Ка-37; Ка-137; ПС-01 "Комар"; Джміль-1 – прототип безпілотного літального апарату Бджола-1Т; Бджола-1Т; ВР-2; ВР-3; Ту-123 «Яструб» (ДБР-1) – надзвуковий дальній безпілотний розвідник; Ту-130; Ту-141 "Стриж"; Ту-143 "Рейс"; Ту-243 "Рейс-Д" – дозвуковий розвідник; Ту-300 "Коршун"; «Скат» – дозвуковий ударний; ZALA 421-08; Ельф-Д. Розглянемо докладніше деякі з бойових повітряних роботів.

БПЛА «Скат» (рис. 2.6) – розвідувальний та ударний безпілотний літальний апарат. Вперше був представлений на авіасалоні МАКС-2007 як повнорозмірний макет, призначений для відпрацювання конструкторсько-компонувальних рішень.

Цей літальний апарат виконує ведення розвідки, завдання ударів по наземних цілях авіабомбами та керованими ракетами Х-59, знищення радіолокаційних систем ракетами Х-31. ZALA 421-08 – надмалий безпілотний літальний апарат. Цей робот призначений для спостереження, цілевказівки, коригування вогню, оцінки збитків.

Весь комплекс складається із системи управління та двох апаратів. Польова комплектація передбачає компактний ручний електрогенератор для заряджання акумуляторів. Як змінне корисне навантаження в стандартному варіанті на борту

встановлені дві кольорові камери: одна дивиться вниз і вперед, друга поворотна по крену дивиться вниз-вбік.



Рисунок 2.6 – БПЛА «Скат»

Як додаткові корисні навантаження виступає тепловізор і фотокамера. Маршрут БПЛА можна змінювати у реальному часі. Якщо оператора цікавить якийсь об'єкт докладніше, він просто натискає пальцем на сенсорний екран ноутбука, і вибирає наприклад фігуру «коло вліво» та ZALA 421-08, закладаючи лівий віраж, починає робити «воронку», безперервно утримуючи мету в об'єктиві бортової камери. Запускається ZALA 421-08 із рук. Метод посадки – автоматично з парашутом.

Наступний вид бойових роботів – це мобільні робототехнічні комплекси (МРК). Ознайомимося докладніше з деякими з моделей сучасних бойових МРК.

МРК Guardian (рис. 2.7) – безпілотний військовий автомобіль. Створений ізраїльською фірмою G-NIUS, що належить ізраїльським оборонним компаніям «Ельбіт Мааракот» та «Таасія Авіріт». Призначений для патрулювання, супроводу автоколон, ведення розвідки та охорони. Guardian побудований на

базі чотириколісного баггі, що забезпечує підвищену прохідність на пересіченій місцевості. Вступив на озброєння Армії оборони Ізраїлю на початку 2009 року.

МРК Swords (скорочення від Special Weapons Observation Reconnaissance Detection Systems) – спеціальна бойова система спостереження та розвідки.

Створено компанією Фостер-Міллер TALON Робот. За твердженням виробника робот призначений для дій у місті, здатний долати пісок, воду та сніг (до 30,48 м глибини), а також здійснювати підйом сходами.



Рисунок 2.7 – Безпілотний військовий автомобіль Guardium

Він розрахований на 8,5 годин роботи від батареї у нормальному експлуатаційному режимі – очікування до 7 діб. Контролюється оператором на відстані до 1000 метрів. Він важить близько 45,36 кг або 27,22 кг у версії розвідки.

Є ряд різних видів зброї, які можуть бути розміщені на SWORDS: гвинтівки M16, 5,56-мм SAW M249, 7,62 мм кулемет M240, гвинтівки Барретт M82.50, шестиствольний 40-мм гранатомет або чотириствольний 61 мм.

2.5 Побутові роботи

Спочатку роботи з'явилися у наших будинках у вигляді роботів-іграшок. Одним із перших прикладів вдалої масової промислової реалізації роботів-іграшок став собачка AIBO корпорації Sony (рис. 2.8).



Рисунок 2.8 – Робот-собака AIBO

Вона має безліч модифікацій, перша модель була випущена у 1999 році. AIBO вміє ходити, «бачити» навколишні предмети за допомогою відеокамери та інфрачервоних датчиків відстані, розпізнавати команди та обличчя. Робот є повністю автономним: він може вчитися та розвиватися, ґрунтуючись на спонуканнях свого господаря, обстановки, чи іншого AIBO. Незважаючи на це, він піддається налаштуванням за допомогою спеціальних програм. Існує програмне забезпечення, що імітує «дорослого собаку», який відразу використовує всі свої функції і програмне забезпечення, що імітує «цуценя», яке розкриває свої можливості поступово.

Однак зараз найпоширенішими побутовими роботами є роботи-пилососи. Існують такі моделі цих пристроїв – Roomba (виробник – iRobot), Trilobite (виробник – Electrolux), Karcher RC3000 (виробник – Karcher) та ін.

Roomba – роботизований пилосос, розроблений та продається iRobot (рис. 2.9). Пристрій є диском 34 см в діаметрі і менше 9 см у висоту. Великий контактний сенсор встановлений у передній частині пристрою з інфрачервоним датчиком по центру у верхній передній частині. Для роботи Roomba використовує внутрішні акумулятори та потребує регулярної підзарядки від настінного модуля. Нові моделі третього покоління мають можливість пошуку зарядного пристрою, зв'язуючись із ним через інфрачервоний датчик. Roomba має невелику висоту. Він досить низький, щоб пройти під ліжком чи іншими меблями. Якщо він зрозуміє, що застряг, він перестане рухатися і почне подавати звукові сигнали, що допомагають власнику його виявити.



Рисунок 2.9 – Робот-пилосос Roomba

Інший клас пристроїв – сторожові чи охоронні роботи. iRobot Corporation представила універсального домашнього робота-візок iRobot-LE, керувати яким можна з будь-якої точки світу через будь-який web-браузер.

Для цього робот обладнаний усім необхідним: відеокамерою, мікрофонами, рушієм, що дозволяє навіть самостійно підніматися стандартними сходами, бортовим комп'ютером класу Pentium II, бездротовим доступом в Інтернет, датчиками і сенсорами, що запобігають зіткненням і навіть дозволяють відчувати запах гару. Як бортова операційна система використовується Linux.

2.6 Обґрунтування та вибір основних компонентів мобільного робота

У роботі розглядається створення мобільного робота для роботи в місцях підвищеної небезпеки, таких як завали, місця підвищеної радіації, тощо. Управління роботом можливо дистанційно, що анулює будь яку можливість травмування оператора. Робот буде зроблений з металу, що підвищить його та матиме гусеничне управління, що дає підвищену можливість маневрувати. У місцях де немає інтернету, можливо керувати роботом з підключеним кабелем.

Для його розробки було використано одноплатний комп'ютер від компанії Nvidia, а саме Jetson Nano 2GB (рис. 2.10). Цей комп'ютер було використано на заміну Raspberry PI 4b із-за наявності окремого графічного ядра, а також архітектури яка призначена для навчання нейронних мереж CUDA. Для даного комп'ютера підходять всі комплектуючі від Raspberry Pi.



Рисунок 2.10 – Jetson Nano 2GB

Характеристики Jetson Nano 2GB:

Процесор:

- Чотирьохядерний ARM Cortex-A57 64-розрядний на частоті 1,42 ГГц

Оперативна пам'ять:

- 4-ГБ LPDDR4

Графічний процесор:

- NVIDIA Maxwell з 128 ядрами NVIDIA CUDA

Для контролю двигунів використовуються GPIO порти які також є і в Raspberry Pi. Дані порти можуть бути тільки ввімкнуті, або вимкнуті, тобто сигнал 0 або 1. На рисунку 2.11 показано всі GPIO порти.

SoC GPIO	Linux GPIO #	Alternate Function	Default Function			Default Function	Alternate Function	Linux GPIO #	SoC GPIO
			3.3 VDC	①	②	5 VDC			
PJ.03	75	GPIO	I2C1_SDA	③	④	5 VDC			
PJ.02	74	GPIO	I2C1_SCL	⑤	⑥	GND			
PBB.00	216	AUD_CLK	GPIO	⑦	⑧	UART1_TXD	GPIO	48	PG.00
			GND	⑨	⑩	UART1_RXD	GPIO	49	PG.01
PG.02	50	UART1_RTS	GPIO	⑪	⑫	GPIO	I2S0_SCLK	79	PJ.07
PB.06	14	SPI1_SCK	GPIO	⑬	⑭	GND			
PY.02	194		GPIO	⑮	⑯	GPIO	SPI1_CS1	232	PDD.00
			3.3 VDC	⑰	⑱	GPIO	SPI1_CS0	15	PB.07
PC.00	16	SPI0_MOSI	GPIO	⑲	⑳	GND			
PC.01	17	SPI0_MISO	GPIO	㉑	㉒	GPIO	SPI1_MISO	13	PB.05
PC.02	18	SPI0_SCK	GPIO	㉓	㉔	GPIO	SPI0_CS0	19	PC.03
			GND	㉕	㉖	GPIO	SPI0_CS1	20	PC.04
PB.05	13	GPIO	I2C0_SDA	㉗	㉘	I2C0_CLK	GPIO	18	PC.02
PS.05	149	CAM_MCLK	GPIO	㉙	㉚	GND			
PZ.00	200	CAM_MCLK	GPIO	㉛	㉜	GPIO	PWM	168	PV.00
PE.06	38	PWM	GPIO	㉝	㉞	GND			
PJ.04	76	I2S0_FS	GPIO	㉟	㊱	GPIO	UART1_CTS	51	PG.03
PB.04	12	SPI1_MOSI	GPIO	㊲	㊳	GPIO	I2S0_DIN	77	PJ.05
			GND	㊴	㊵	GPIO	I2S0_DOUT	78	PJ.06

Рисунок 2.11 – GPIO порти Jetson Nano

Операційна система Jetson Nano це модифікована Ubuntu.

Для проекту буде використана 8-мегапіксельна камера Raspi CAM v2 бази сенсора Sony IMX219. Камера сумісна з усіма версіями Raspberry. Вона прикріплюється шлейфом, який іде в комплекті, до роз'єму CSI.

Камера невелика у розмірах: 25 x 23 x 9 мм. Розширення фото, отриманих з камера – 3280 x 2464 пікселів. Підтримувані формати відео – 1080p з швидкістю 30 кадрів за секунду(fps), 720p з 60 fps і 480p з 90 fps. Фокусна відстань дорівнює 3.33 мм. Камера зображена на рисунку 2.12.



Рисунок 2.12 – Raspi CAM v2

Робот рухається за допомогою двох двигунів з номінальною робочою напругою 6В, для контролю цих двигунів використовується драйвер LN298N .



Рисунок 2.13 – Двигун постійного струму

Оскільки швидкість обертання двигуна залежить від величини напруги, а Raspberry Pi не може контролювати подачу напруги на вхід, то буде використаний драйвер LN298N.

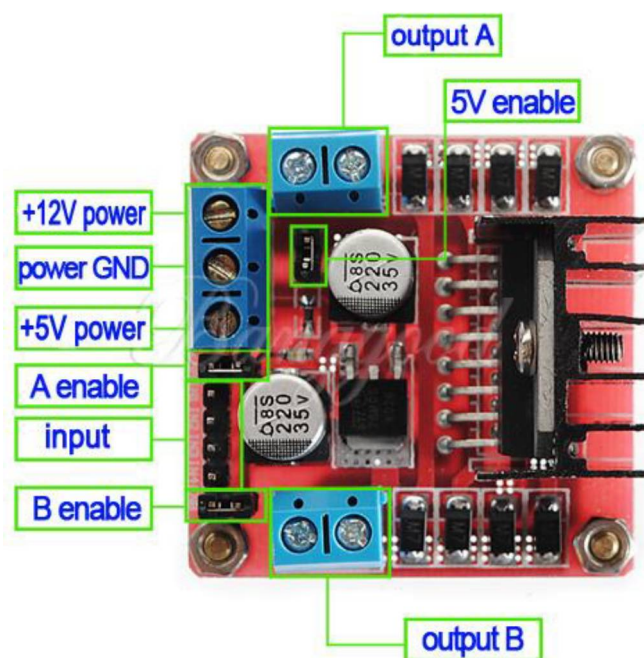


Рисунок 2.14 – Драйвер LN298N

Модуль дуже простий у використанні – він підтримує одночасно два двигуни для підключення і може керувати ними одночасно. На входи ми подаємо виходи двигунів, на виходи – 4 керуючі сигнали для двигунів, які підуть на I/O контакти GPIO, напруга на модуль подається зі стороннього джерела струму, так як потужності Raspberry Pi не вистачає для живлення двигунів.

Оскільки у нас всього два двигуна для повертання робота вправо чи вліво буде реалізовано за допомогою коду. Перевагою даного управління є те що робот може повернути на місті.

Шасі DFRobot Devastator v2 воно виконано із алюмінію. Всі деталі між собою з'єднуються за допомогою болтів і гайок. Однією з особливостей цього робота є амортизація підвіски, а також гусениці.

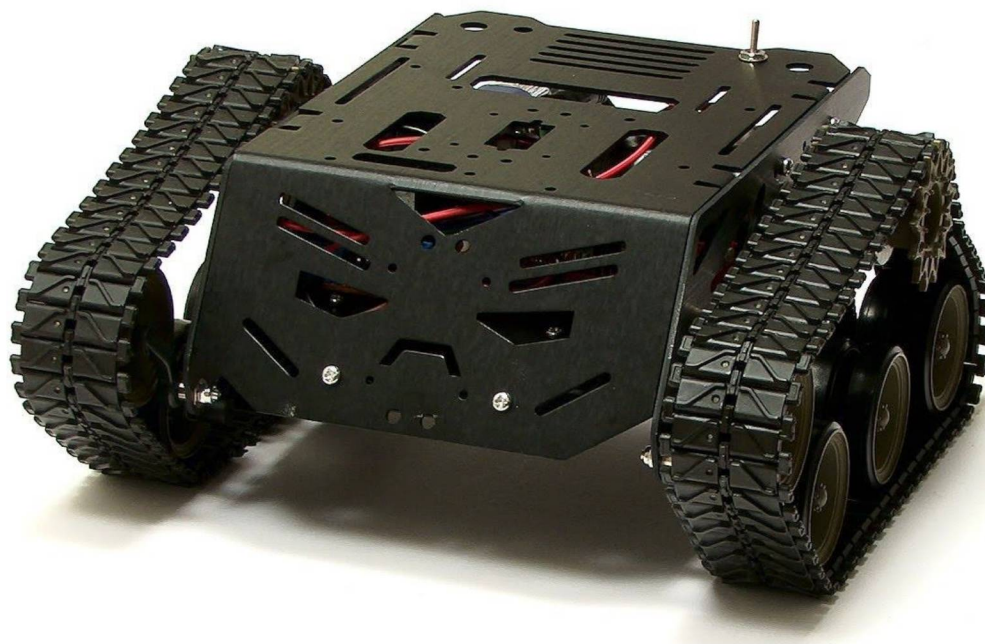


Рисунок 2.15 – Devastator v2

2.7 Проектування мобільного робота

У передній частині буде розміщений батарейний відсік для живлення драйвера двигунів, і сам драйвер, що буде регулювати подачу струму на двигуни.

У середині системи буде розміщено два двигуни, що під'єднані до гусениць – це буде приводом платформи. Всі частини з'єднані болтами та гайками.

Зверху робота буде знаходитись Jetson Nano в своєму окремому корпусі для мінімізації можливості його пошкодити.

На рис. 2.16 зображена електрична схема підключення драйверу та двигунів до Jetson Nano.

Готовий прототип робота зображений на рисунку 2.17.

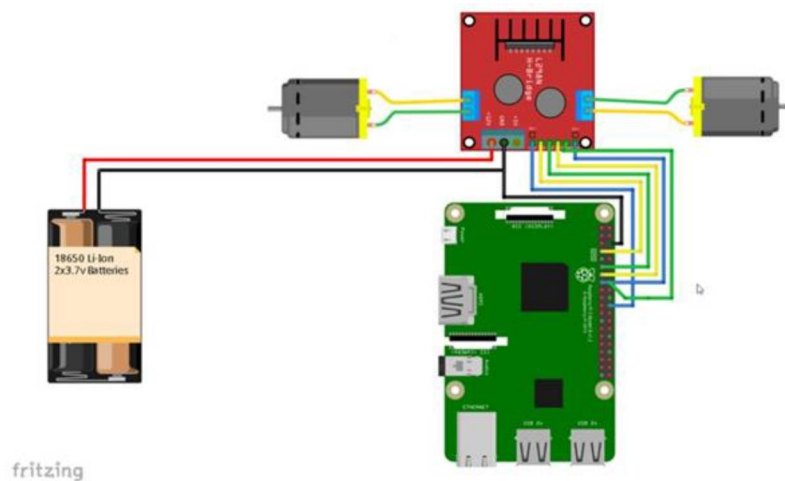


Рисунок 2.16 – Схема електричного кола робота



Рисунок 2.17 – Готовий прототип мобільного робота

Для керування двигунами використовується драйвер L298N. Драйвер здатен керувати двома двигунами постійного струму. Для кожного двигуна є два цифрові входи управління. Один вхід використовується для вибору напрямку

обертання, а інший для управління швидкості обертання двигуна. Швидкість регулюється за допомогою ШІМ (широтно імпульсну модуляцію). В таблиці 2.1 показаний результат подачі високого сигналу на входи драйвера двигуна.

Таблиця 2.1

Подача високого сигналу на входи драйвера двигуна

Вхід		Вихід		
1A	1B	2A	2B	Опис
0	0	0	0	Вимкнутий
0	1	0	1	Вперед
1	0	1	0	Назад
1	0	0	1	Вліво
0	1	1	0	Вправо
1	1	1	1	Вимкнутий

Висновок за розділом 2

В сучасному світі робототехніка використовується в багатьох сферах, навіть воєнній. Але в показаних прикладах, всі роботи були досить великими, тому не могли працювати у вузьких місцях. Тому розробка маленького робота який працює віддалено є актуальною.

Використання нейронних мереж дуже прискорить пошук потрібних об'єктів.

Плата Jetson Nano за свою ціну є ідеальним вибором якщо треба використовувати нейронні мережі, так як містить окреме графічне ядро, та архітектуру ядер MAXWELL. Також містить програмно-апаратну архітектуру паралельних обчислень CUDA, яка багатократно прискорює обробку даних для графічних ядер NVIDIA.

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОБІЛЬНОГО РОБОТА

3.1 Розробка програмного забезпечення для управління роботом

Для управління мобільним роботом буде використано локальний сервер. Сервер буде працювати на Jetson Nano, тому затримка між передачею даних з камери для обробки буде мінімальною. Програмне забезпечення на якому реалізований сервер – це python фреймворк Flask. Управління роботом було реалізоване за допомогою веб сайту. На сайті було реалізовано 4 кнопки управління і блок де буде показано відеотраснляцію.

Одночасно клієнт для перегляду потокового відео має інтерфейс керування роботом. Для керування системою можна використовувати кнопки «вперед», «назад», «вліво», «вправо».

Контроль реалізовано таким чином. Коли користувач натискає кнопку відправляється GET запит на сервер, де він оброблюється і на відповідний GPIO порт віддається високий сигнал. Коли користувач відпускає кнопку на сервер де GET запит, де він опрацьовується і зупиняє подачу високого сигналу.

3.2 Структура згорткової нейронної мережі

Для розпізнавання зображень, а саме для розпізнавання сцен була використана одна із згорткових мереж, а саме ResNet18. Використання ResNet18 обумовлене декількома причинами, а саме точністю класифікації в порівнянні з іншими згортковими мережами. Однією причиною використання саме ResNet мережі, є те що, при ускладненні задачі потрібно збільшувати кількість слоїв, але точність починає падати. На рис. 3.1 показано графік який описує відсоток

помилки при навчанні і тестуванні даних для 20-шарової мережі і 56-шарової мережі [20].

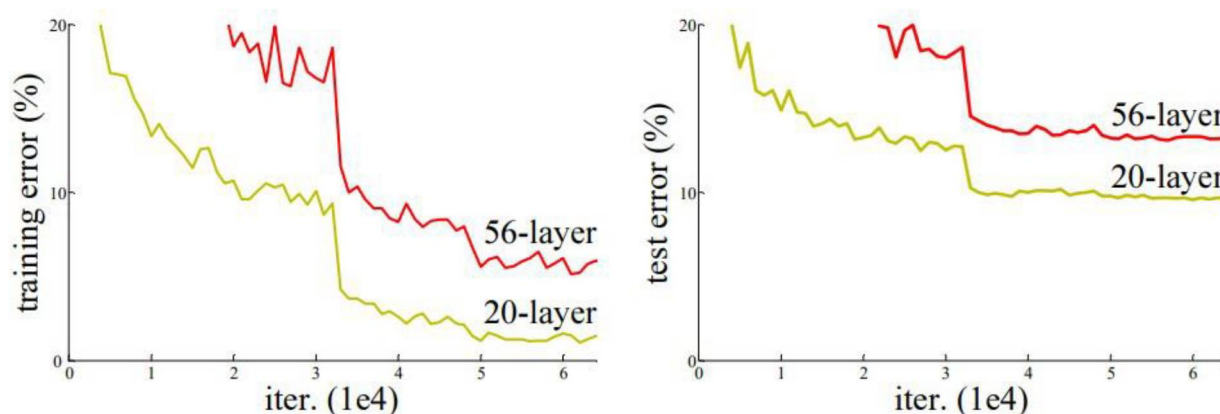


Рисунок 3.1 – Графік відсотку помилок при 20 і 56 шарах нейронної мережі

На рис. 3.1 видно, що відсоток помилки для 56-шарової мережі більше, ніж для 20-шарової мережі як в разі адаптації, так і даних тестування. Це говорить про те, що з додаванням додаткових шарів поверх мережі її продуктивність погіршується. Це може бути пов'язано з функцією оптимізації, ініціалізацією мережі і, що більш важливо, з проблемою зникаючого градієнта. Можна вважати, що це теж може бути результатом перенавчання, але тут відсоток помилки 56-шарової мережі найгірший як при навчанні, так і при тестуванні даних, чого не відбувається при перенавчанні моделі [21]. На рис. 3.2 зображена точність класифікації різних згорткових мереж.

Але як можна помітити, що мережі ResNet50, ResNet101, ResNet152 показують кращі результати, при великій кількості слоїв. А результатом такої точності є сама архітектура блоку ResNet.

model	ACC@1	ACC@5
AlexNet	56.522	79.066
VGG-11	69.020	88.628
VGG-13	69.928	89.246
VGG-16	71.592	90.382
VGG-19	72.376	90.876
VGG-11 with batch normalization	70.370	89.810
VGG-13 with batch normalization	71.586	90.374
VGG-16 with batch normalization	73.360	91.516
VGG-19 with batch normalization	74.218	91.842
ResNet-18	69.758	89.078
ResNet-34	73.314	91.420
ResNet-50	76.130	92.862
ResNet-101	77.374	93.546

Рисунок 3.2 – Точність класифікації згорткових мереж

Ще однією проблемою при великій кількості шарів є так названа «втрата градієнту». Але дана проблема навчання дуже глибоких мереж було полегшено з введенням ResNet або залишкових мереж, і Resnet складаються із залишкових блоків. На рис. 3.3 зображено структуру залишкового блоку ResNet.

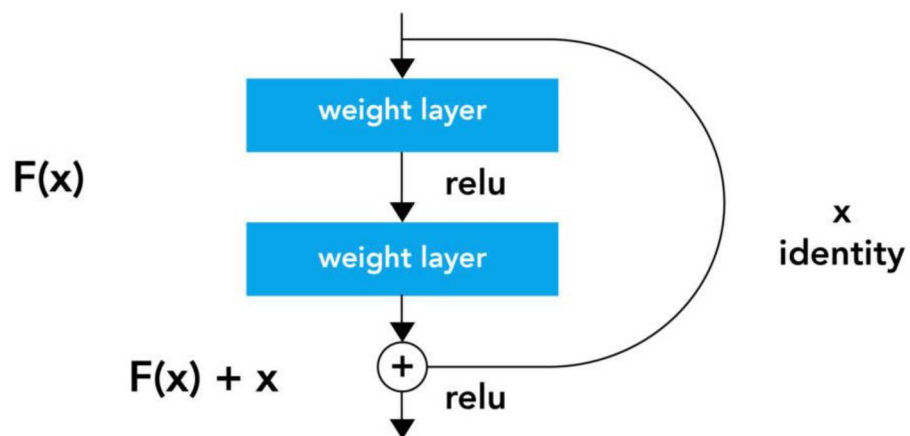


Рисунок 3.3 – Залишковий блок ResNet

Найперше, що помічаємо різним, це те, що існує прямий зв'язок, який пропускає деякі шари (можуть відрізнятися в різних моделях) між ними. Це з'єднання називається "пропуск з'єднання" і є ядром залишкових блоків. Через це пропускне з'єднання, вихід шару зараз не однаковий. Без використання цього з'єднання пропуску, вхід "x" помножується на ваги шару з подальшим додаванням зміщення.

Потім цей член проходить через функцію активації $f()$, і ми отримуємо результат як H . В стандартній мережі результат $H(x)$ був таким [14]:

$$H(x) = f(wx + b), \quad (3.1)$$

Тепер, з введенням пропускового з'єднання, висновок змінений на [14]:

$$H(x) = f(x) + x. \quad (3.2)$$

Але тепер є невелика проблема з цим підходом, коли розміри вхідних даних відрізняються від розмірів вихідних даних, що може статися з згортковими та об'єднаними шарами. У цьому випадку, коли розміри $f(x)$ відрізняються від x , ми можемо взяти два підходи:

- з'єднання для пропуску заповнено зайвими нульовими записами, щоб збільшити його розміри.
- метод проекції використовується для узгодження розмірності, що робиться додаванням 1×1 згорткових шарів до введення. У такому випадку вихідний результат є:

$$H(x) = f(x) + w1.x. \quad (3.3)$$

Тут додається додатковий параметр w_1 , тоді як додатковий параметр не додається при використанні першого підходу.

Розберемо саму структуру згорткової мережі ResNet18. 18 шарів ResNet18 представляють 18 шарів з вагами, включаючи згортковий шар і повністю пов'язаний рівень, за винятком рівня пулу і рівня BN. На рис. 3.4 показано структурну схему та різницю між мережами ResNet [20].

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Рисунок 3.4 – Структурна схема мереж ResNet

Розберемо з яких блоків складаються згорткові мережі ResNet. У ResNet використовуються два основні типи блоків, залежно від того, що відповідають вхідним та вихідним розмірам.

- блок ідентичності: коли розміри активації входу та виходу відповідають.
- блок згортки: коли розміри активації входу та виходу відрізняються один від одного.

Наприклад, щоб зменшити розмір активації ($H \times W$) у 2 рази, ви можете використовувати згортку 1×1 із кроком 2.

На рис. 3.5 нижче показано, як виглядають залишкові блоки та що знаходиться всередині цих блоків [21].

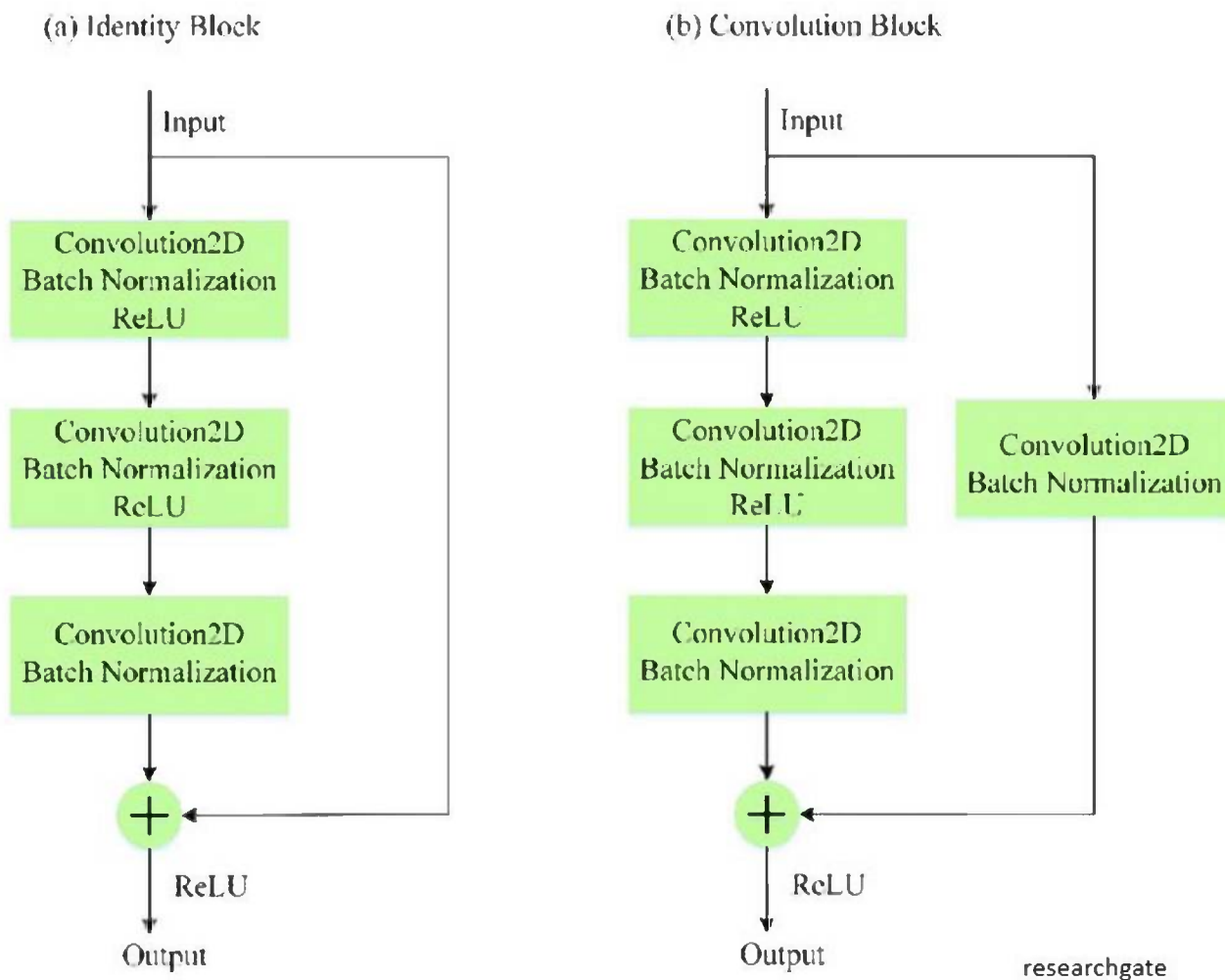


Рисунок 3.5 – Блок ідентичності та згортки

Як видно з рисунку вище всі блоки мережі ResNet складається з остаточних блоків. А сама архітектура ResNet18 показана рисунку 3.6.

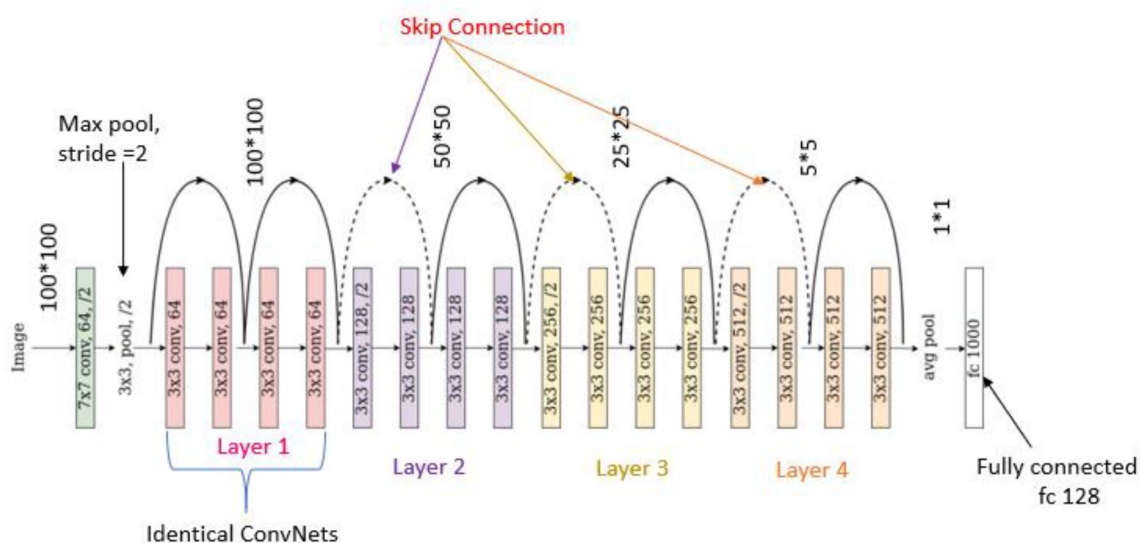


Рисунок 3.6 – Архітектура ResNet18

3.3 Розробка згорткової нейронної мережі

В якості програмного забезпечення для розробки згорткової мережі було обрано OpenCV (Tensorflow), так як вони офіційно підтримують архітектуру графічного ядра Jetson Nano. Існують бібліотеки для наукових обчислень. Наприклад, NumPy – призначена для роботи з N-мірними масивами, швидких обчислень у сфері лінійної алгебри та інших. Існує кілька фреймворків машинного навчання для python. Один з найпопулярніших – keras, є високорівневим та досить простим до освоєння. Він служить обгорткою над іншими фреймворками машинного навчання такими як TensorFlow, Theano, CNTK.

Бібліотека, яка відповідає за навчання нейронної мережі – це jetson-inference [22], та jetson utils. Бібліотеки написані на C++, що прискорює їх виконання. Бібліотека містить велику кількість алгоритмів та вибір згорткових мереж.

Для виклику алгоритму розпізнавання треба вказати об'єкт, згорткову нейронну мережу, камеру для розпізнавання, і монітор на який буде виводитись результат.

Об'єкт `detectNet` приймає зображення як вхідні дані та виводить список координат виявлених обмежувальних прямокутників разом із їхніми класами та значеннями довіри. `DetectNet` доступний для використання з Python і C++. Нижче описано різні попередньо навчені моделі виявлення, доступні для завантаження. За замовчуванням використовується модель `SSD-Mobilenet-v2` 91 класу, навчена на наборі даних MS COCO, яка досягає продуктивності висновку в реальному часі на Jetson з TensorRT[23].

Нижче показано виклик об'єкту `detectNet`, з мережею `SSD-Mobilenet-v2`, та захопленням відео з камери в CSI порті.

```
net = jetson.inference.detectNet("ssd-mobilenet-v2", threshold=0.5)
camera = jetson.utils.videoSource("csi://0") # '/dev/video0' for V4L2
display = jetson.utils.videoOutput("display://0") # 'my_video.mp4' for file
```

Але також є можливість встановлення своєї мережі, для цього треба запустити закинути вже навчену мережу, а потім запустити баш скрипт.

```
./download-models.sh
```

Даний файл робить перевірку на відповідність нейронної мережі,

COCO [24] розшифровується як Common Objects in Context, оскільки набір даних зображень був створений з метою покращення розпізнавання зображень. Набір даних COCO містить складні високоякісні візуальні набори даних для комп'ютерного зору, в основному сучасні нейронні мережі.

Особливості набору даних COCO:

- сегментація об'єктів з анотаціями примірників;
- призначення в контексті;
- сегментація суперпіксельного матеріалу;

- позначено більше 200 000 зображень із 330 000 зображень;
- 1,5 мільйона примірників об'єктів;
- 80 категорій об'єктів, «класи COCO», які включають «речі», окремі екземпляри, які можуть бути легко позначеними (людина, автомобіль, стілець тощо);
- 91 категорія матеріалів, де «матеріал COCO» включає матеріали та об'єкти без чітких кордонів (небо, вулиця, трава тощо), які надають важливу контекстну інформацію;
- 5 підписів на зображення;
- 250000 людей із 17 різними ключовими точками, широко використовуваними для оцінки позицій.

Класи набору даних COCO для виявлення та відстеження об'єктів включають наступні попередньо навчені 80 об'єктів.

Приклад зображення з датасету показано на рис. 3.7.

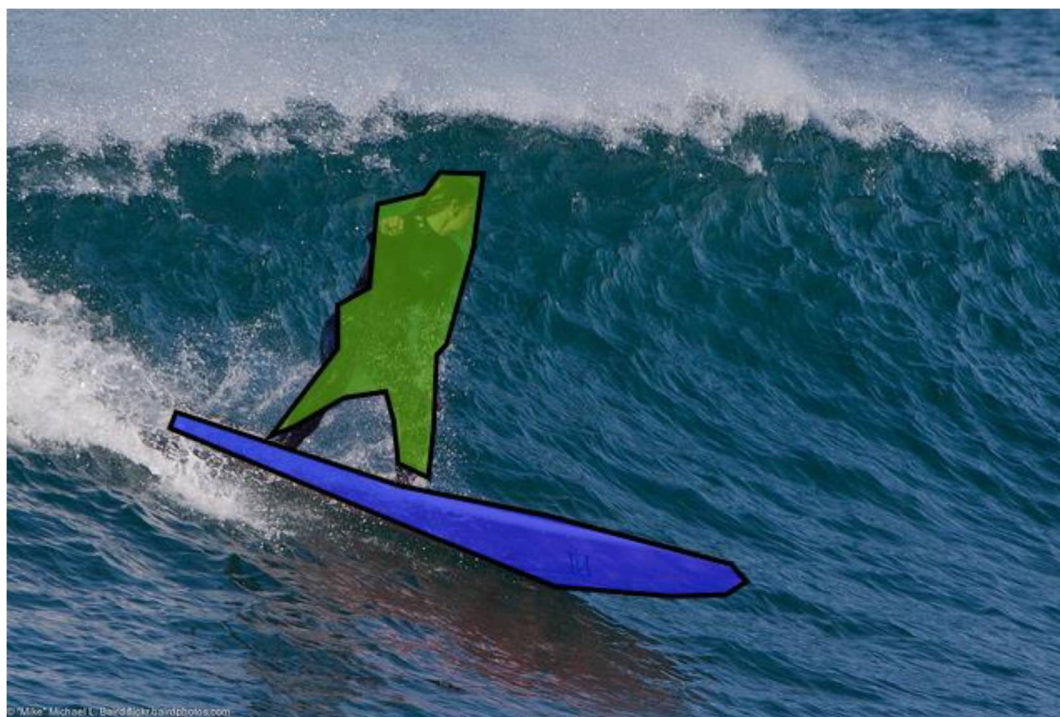


Рисунок 3.7 – Зображення з датасету

Після підключення бібліотеки треба передавати зображення з камери до циклу для обробки. Далі створюємо інтерфейс відеовиходу з об'єктом `videoOutput` і створимо головний цикл, який працюватиме до тих пір, поки користувач не вийде:

```
display = jetson.utils.videoOutput("display://0") while
display.IsStreaming():
```

Перше, що відбувається в головному циклі, це захоплення наступного відеокадра з камери. `camera.Capture()` буде чекати, поки наступний кадр не буде надіслано з камери та завантажено в пам'ять графічного процесора.

```
img = camera.Capture()
```

Повернене зображення буде об'єктом `jetson.utils.cudaImage`, який містить такі атрибути, як ширина, висота та формат пікселів:

```
<jetson.utils.cudaImage>
    .ptr # адреса пам'яті (зазвичай не використовується)
    .size # розмір у байтах
    .shape # (висота, ширина, канали) кортеж
    .width # ширина в пікселях
    .height # висота в пікселях
    .channels # кількість кольорових каналів
    .format # рядок форматування
    .mapped # true, якщо ZeroCopy
```

Далі мережа виявлення обробляє зображення за допомогою функції `net.Detect()`. Він отримує зображення з `camera.Capture()` і повертає список виявлення:

```
detections = net.Detect(img)
```

Ця функція також автоматично накладе результати виявлення на вхідне зображення.

Також можливо додати сюди заяву `print(detections)`, і координати, впевненість і інформація про клас будуть роздруковані на терміналі для кожного результату виявлення.

Результат виконання розпізнавання зображення на рис. 3.8.

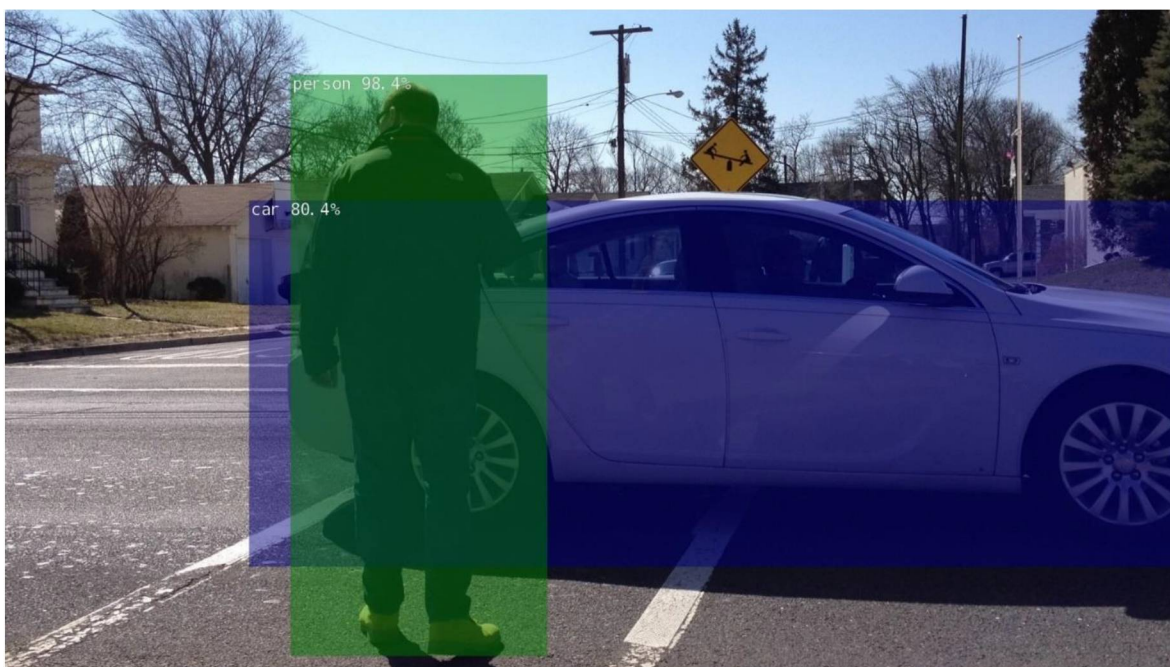


Рисунок 3.8 – Результат розпізнавання зображення

Точність розпізнавання при прямій трансляції 85%. Точність падає при зменшенні освітлення в кімнаті, так як збільшується зашумленість фото.

Середній FPS при трансляції дорівнює 32. FPS зменшується при збільшенні кількості об'єктів, що розпізнає мережа. Це трапляється із-за того що графічне ядро не встигає оброблювати велику кількість об'єктів в кадрі.

В таблиці 3.1 показані середні результати точності розпізнавання.

Спираючись на дані, отримані в таблиці 3.1, можна сказати, що ResNet-18 показала кращий результат, але найменшу кількість кадрів в секунду.

Таблиця 3.1

Результати розпізнавання

Назва мережі	Назва датасету	Точність при навчанні	Точність при тестуванні	Середній FPS
SSD-Mobilenet-v1	MS COCO	-	85,4%	32
SSD-Mobilenet-v2	MS COCO	-	87,2%	34
SSD-Inception-v2	MS COCO	-	80,4%	36
ResNet-18	MS COCO	92,3%	86,7%	29

Висновок за розділом 3

Розібрано архітектуру згорткової мережі ResNet18, та виділено її переваги.

Було реалізовано базову архітектуру рекурентної згорткової нейронної мережі ResNet-18, так як мережа показала досить хороші результати, то зміни в архітектурі не проводились.

Для навчання мережі було використано бібліотеку jetson-inference. Яка має велику кількість алгоритмів для навчання, та вбудованих навчених нейронних мереж.

Датасетом для навчання було обрано MS COCO, який спеціально створено для навчання мереж для розпізнавання об'єктів. Містить в собі 80 категорій зображень.

Для прискорення навчання було використано TensorRT. Це архітектурне рішення для прискорення паралельної обробки даних нейронної мережі.

ВИСНОВКИ

В результаті виконання магістерської дипломної роботи був розроблений прототип мобільного робота для розпізнавання об'єктів реальному часі.

Платою керування роботом виступила Nvidia Jetson Nano 2GB, як альтернатива Raspberry Pi 4b, із-за наявності окремого графічного ядра, та програмно-апаратного прискорювача паралельних обчислень CUDA.

Управління роботом здійснюється за допомогою веб додатку, віддалено.

Було досліджено можливість використання рекурентної згорткової мережі ResNet-18 для розпізнавання образів. Навчання нейронної мережі проводилось на датасеті MS COCO, що після навчання була отримана точність розпізнавання в 86.7 відсотка. Також було порівняно з іншими нейронним мережами що були навчені на цьому ж датасеті, і ResNet-18 показала одну з найбільших точностей з всіх нейронних мереж, що були перевірені.

Розроблена модель може бути використана для промислових проектів, але потребує деяких допрацювань:

1. Зменшення ваги рами(заміна її на пластикову).
2. Використання більш потужної плати JETSON TX2 NX з більш новою архітектурою Pascal.
3. Ізоляція корпусу для захисту від вологи.

Використання не інтернету для передачі інформації, а радіоканалу, що збільшить дальність управління.

СПИСОК ЛІТЕРАТУРИ

1. Основи теорії розпізнавання образів : навч. посіб. : у 2 ч. / А. С. Довбиш, І. В. Шелехов. – Суми : Сумський державний університет — 2015. — Ч. 1. — 109 с. (Last accessed: 17.05.2021)
2. Harshali M. Image classification using Deep learning — 2015 (Last accessed: 21.05.2021)
3. Hongkai W. Zongwei Z. Yingci L. Zhonghua C. Peiou L. Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non-small cell lung cancer from 18F-FDG PET/CT images — 2017 (Last accessed: 20.05.2021)
4. 7 Types of Classification Algorithms, URL: <https://analyticsindiamag.com/7-types-classification-algorithms/> (Last accessed: 18.05.2021)
5. Kang T. Using Neural Networks for Image Classification — 2015 (Last accessed: 21.05.2021)
6. Samer H. Kumar R. Rowen C. Using Convolutional Neural Networks for Image Recognition, 2015 (Last accessed: 21.05.2021)
7. K. O'Shea, Nash R. An Introduction to Convolutionan Neural Networks — 2015 (Last accessed: 21.05.2021)
8. Understanding Convolutions [Електронний ресурс]. URL: <http://colah.github.io/posts/2014-07-Understanding-Convolutions/> (Last accessed: 21.05.2021)
9. Lai S., Xu L., Liu K., Zhao J. Recurrent Convolutional Neural Networks for Text Classification — 2015. (Last accessed: 22.05.2021)
10. Nielsen, A. M. Neural Networks and Deep Learning / A. M. Nielsen – Determination Press, 2015

11. Goodfellow, I. Deep learning / I. Goodfellow, Y. Bengio, A. Courville. – MIT Press, 2016 – 785p.

12. Класифікація мобільних роботів. URL: <https://postnauka.ru/video/34424>

13. Про мобільні роботи: роль та перспективи промислового та побутового застосування, популярні моделі URL: <https://vektorus.ru/blog/mobilnye-roboty.html>

14. Роботи з галузей та призначення (області застосування роботів) URL: <http://robotrends.ru/robopedia/roboty-po-otraslyam-i-naznacheniyu>

15. Промислові роботи URL: https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BC%D1%8B%D1%88%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D1%80%D0%BE%D0%B1%D0%BE%D1%82

16. Батанов А. Ф., Грицынин С. Н., Муркин С. В. Робототехнические комплексы для обеспечения специальных операций // Специальная Техника. №6. – 1999. – С. 10–17.

17. Струтинський В.Б., Юрчишин О.Я., Кравець О.М. Розвиток основних положень проектування маніпуляторів мобільних роботів спеціального призначення адаптованих для роботи з небезпечними об'єктами // XXII МНТК “Прогресивна техніка, технологія та інженерна освіта” – 2021 – С. 129 - 131

18. Безпілотний літальний апарат URL: https://uk.wikipedia.org/wiki/%D0%91%D0%B5%D0%B7%D0%BF%D1%96%D0%BB%D0%BE%D1%82%D0%BD%D0%B8%D0%B9_%D0%BB%D1%96%D1%82%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BF%D0%B0%D1%80%D0%B0%D1%82

19. Типи бойових роботів URL: https://wiki.tntu.edu.ua/%D0%91%D0%BE%D0%B9%D0%BE%D0%B2%D0%B8%D0%B9_%D1%80%D0%BE%D0%B1%D0%BE%D1%82

20. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition — 2015 (Last accessed: 22.05.2021)

21. Maeda-Gutiérrez, V. Galván-Tejada, C.E.; Zanella-Calzada, L.A. Celaya-Padilla, J.M. Galván-Tejada, J.I. Gamboa-Rosales, H. Luna-García, H. Magallanes-Quintanar, R. Guerrero Méndez, C.A.; Olvera-Olvera, C.A. Comparison of Convolutional Neural Network Architectures for Classification of Tomato Plant Diseases — 2020 (Last accessed: 22.05.2021

22. NVIDIA Xavier - Deep Learning - Deep Learning Tutorials - Jetson Inference URL: https://developer.ridgerun.com/wiki/index.php?title=Xavier/Deep_Learning/Deep_Learning_Tutorials/Jetson_Inference

23. Understanding Nvidia TensorRT for deep learning model optimization URL: https://medium.com/@abhaychaturvedi_72055/understanding-nvidias-tensorrt-for-deep-learning-model-optimization-dad3eb6b26d9

24. MS COCO URL: <https://cocodataset.org/#home>