

Невлюдов І.Ш., Євсєєв В.В., Максимова С.С.

**INDUSTRY 5.0
ТА
КОЛАБОРАТИВНА РОБОТОТЕХНІКА:
ДИНАМІЧНИЙ ОПИС
НАВКОЛИШНЬОГО СЕРЕДОВИЩА
РОБОТІВ-МАНІПУЛЯТОРІВ З
ВИКОРИСТАННЯМ МОВИ PYTHON**



Харківський національний університет радіоелектроніки
Кафедра комп'ютерно-інтегрованих технологій, автоматизації,
робототехніки та безпекової інженерії (КІТАРБІ)

Невлюдов І.Ш., Євсєєв В.В., Максимова С.С.

**INDUSTRY 5.0 ТА КОЛАБОРАТИВНА РОБОТОТЕХНІКА:
ДИНАМІЧНИЙ ОПИС НАВКОЛИШНЬОГО СЕРЕДОВИЩА
РОБОТІВ-МАНІПУЛЯТОРІВ З ВИКОРИСТАННЯМ МОВИ PYTHON**

[Монографія]

Харків

2026

УДК 681.51:[004.896:004.438]

H40

Рекомендовано Науково-технічною радою Харківського національного університету радіоелектроніки (протокол № 1 від 30.01.2026 року)

Рецензенти:

доктор технічних наук, професор, завідувач кафедри «Біомедичної інженерії», Харківського національного університету радіоелектроніки

Аврунін Олег Григорович

кандидат технічних наук, доцент, директор Державного підприємства «Південний державний проектно-конструкторський інститут авіаційної промисловості»

Артюх Роман Володимирович

доктор технічних наук, професор, професор кафедри «Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки», Харківського національного університету радіоелектроніки

Цимбал Олександр Михайлович

H40 **Industry 5.0 та колаборативна робототехніка: динамічний опис навколишнього середовища роботів-маніпуляторів з використанням мови Python: монографія / І.Ш. Невлюдов, В.В. Євсєєв, С.С. Максимова. – Харків : Видавництво Іванченка І. С., 2026. – 279 с.**

Монографія "Industry 5.0 та колаборативна робототехніка: динамічний опис навколишнього середовища роботів-маніпуляторів з використанням мови Python" розглядає сучасні тенденції впровадження колаборативних роботів у виробничі процеси Industry 5.0, фокусуючись на їхній взаємодії з людиною, адаптації до динамічних умов середовища та використанні передових технологій програмування для їх контролю.

У вступі окреслюються основні проблеми й завдання, пов'язані з переходом до Industry 5.0, а також роль колаборативних роботів як ключового елементу нової виробничої парадигми. Перший розділ досліджує концепцію колаборативних роботів-маніпуляторів в Industry 5.0, висвітлюючи відмінності від Industry 4.0, нову екосистемну модель та їхні завдання у промислових умовах.

Другий розділ фокусується на динамічному описі навколишнього середовища колаборативного робота, зокрема на методах і алгоритмах, які дозволяють роботу адаптуватися до змін у середовищі в реальному часі. Важливу роль відіграє комп'ютерний зір і розпізнавання об'єктів за допомогою адаптивних алгоритмів.

Третій розділ присвячений використанню мови програмування Python для розробки робототехнічних систем. У ньому розглядаються основні бібліотеки для машинного навчання та комп'ютерного зору, такі як OpenCV, TensorFlow і MediaPipe, а також середовище розробки PyCharm.

Четвертий розділ демонструє практичні приклади реалізації програм для колаборативних роботів, зокрема використання нейронних мереж для ідентифікації об'єктів у робочій зоні робота, реалізація алгоритмів трекінгу, а також методів прогнозування траєкторії руху людини за допомогою LSTM і фільтра Калмана.

П'ятий розділ аналізує перспективи розвитку колаборативної робототехніки в умовах Industry 5.0, описуючи ключові виклики та тенденції, а також потенціал розширення використання таких роботів у різних галузях промисловості.

Монографія може бути корисна науковцям, аспірантам, здобувачам, які займаються питаннями розробки інтелектуальних робототехнічних систем та фахівцям в галузі знань 17 - Електроніка, автоматизація та електронні комунікації за спеціальністю 174 - Автоматизація, комп'ютерно-інтегровані технології та робототехніка та освітньо-професійних програм: «Робототехніка та кіберфізичні системи», «Автоматизація та комп'ютерно-інтегровані технології» та G7 Автоматизація, комп'ютерно-інтегровані технології та робототехніка.

УДК 681.51:[004.896:004.438]

ISBN:978-617-8332-95-2.

DOI: 10.30837/978-617-8332-95-2.

© Невлюдов І.Ш.,

Євсєєв В.В.,

Максимова С.С., 2026

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ	9
1. Колаборативні роботи-маніпулятори в Industry 5.0.....	11
1.1 Основні принципи та відмінності від Industry 4.0	11
1.2 Екосистемна модель концепції Industry 5.0	28
1.3 Вплив Industry 5.0 на розвиток робототехніки	34
1.4 Визначення та класифікація колаборативних роботів	43
1.5 Особливості взаємодії людини та колаборативного робота	54
1.6 Завдання колаборативних роботів в промислових умовах	58
2. Динамічний опис навколишнього середовища колаборативного робота.....	61
2.1 Методи динамічного опису середовища	61
2.2 Використання комп'ютерного зору для спостереження за середовищем .	66
2.3 Адаптивні алгоритми для аналізу та розпізнавання об'єктів	72
3. Розробка програмного забезпечення для колаборативних роботів маніпуляторів з використанням мови Python	76
3.1 Вступ до Python як основної мови програмування для робототехніки	76
3.2 Використання середовища PyCharm для розробки та налагодження програм	78
3.3 Огляд основних бібліотек для комп'ютерного зору та машинного навчання (cv2, numpy, tensorflow, mediapipe)	81
4. Приклади розробки програм для динамічного опису середовища	91
4.1 Програмна реалізація моделі нейронної мережі MobileNetV2 для розпізнавання та ідентифікації людини в робочій зоні колаборативного робота.....	91
4.2 Дослідження впливу частоти обробки кадрів при ідентифікації об'єктів на базі моделі нейронної мережі MobileNetV2 для мобільного робота	97

4.3 Програмна реалізація алгоритму CAMShift для трекінгу людини в робочій зоні колаборативного робота.....	105
4.4 Використання фільтра Калмана для представлення ймовірнісних моделей визначення місцезнаходження людини в робочій зоні колаборативного робота.....	113
4.5 Використання Convolutional Neural Networks (CNNs) для аналізу та виявлення ключових точок об'єктів на зображеннях.....	122
4.6 Використання рекурентних нейронних мереж LSTM для прогнозування траєкторії переміщення рук людини в робочій зоні колаборативного робота маніпулятора.....	132
4.7 Порівняльний аналіз методів прогнозування траєкторії переміщення об'єктів в робочій зоні колаборативного робота маніпулятора.....	144
4.8 Розпізнавання людини в робочій зоні колаборативного робота маніпулятора на базі глибокої нейронної мережі MobileNetV2 в режимі реального часу.....	154
4.9 Дослідження існуючих методів представлення навколишнього середовища колаборативного робота маніпулятора в рамках кіберфізичних виробничих систем.....	166
4.10 Ідентифікація людини оператора в робочій зоні колаборативного робота для отримання команд в рамках концепції Industry 5.0.....	174
5. Перспективи розвитку робототехніки в умовах Industry 5.0.....	183
5.1 Тенденції та виклики робототехніки в майбутньому.....	183
5.2 Потенціал розширення колаборативних роботів.....	188
5.3 Вплив нових технологій на робочі процеси в Industry 5.0.....	190
Висновки.....	196
Перелік посилань.....	198
Додаток А. Основні функції бібліотеки cv2.....	243
Додаток Б. Основні функції бібліотеки NumPy.....	246
Додаток В. Основні функції бібліотеки TensorFlow.....	250
Додаток Г. Основні функції бібліотеки MediaPipe.....	254

Додаток Е. «Код програми розпізнавання людини в робочій зоні колаборативного робота маніпулятора на базі глибокої нейронної мережі MobileNetV2 в режимі реального часу».....	257
Додаток К. «Код дослідження впливу частоти обробки кадрів при ідентифікації об'єктів на базі моделі нейронної мережі MobileNetV2 для мобільного робота».....	259
Додаток Л. «Код програми реалізації алгоритму CAMShift для трекінгу людини в робочій зоні колаборативного робота»	262
Додаток М. «Код програми використання фільтра Калмана для представлення ймовірнісних моделей визначення місцезнаходження людини в робочій зоні колаборативного робота»	265
Додаток Н. «Код програми використання Convolutional Neural Networks (CNNs) для аналізу та виявлення ключових точок об'єктів на зображеннях»	267
Додаток П. «Код програми використання рекурентних нейронних мереж LSTM для прогнозування траєкторії переміщення рук людини в робочій зоні колаборативного робота маніпулятора»	269
Додаток Р. «Код програми ідентифікації людини оператора в робочій зоні колаборативного робота для отримання команд в рамках концепції Industry 5.0».....	272

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- AI (Artificial Intelligence) - штучний інтелект.
- API (Application Programming Interface) - програмний інтерфейс додатків.
- AR (Augmented Reality) - доповнена реальність.
- BCI (Brain-Computer Interface) - інтерфейс мозок-комп'ютер.
- CAD (Computer-Aided Design) - автоматизоване проєктування.
- CAM (Computer-Aided Manufacturing) - автоматизоване виробництво.
- CAMShift (Continuously Adaptive Mean Shift) - алгоритм безперервного адаптивного середнього зсуву.
- CNC (Computer Numerical Control) - числове програмне керування.
- CNN (Convolutional Neural Network) - згортова нейронна мережа.
- CV (Computer Vision) - комп'ютерний зір.
- DNN (Deep Neural Network) - глибока нейронна мережа.
- ERP (Enterprise Resource Planning) - планування ресурсів підприємства.
- GPU (Graphics Processing Unit) - графічний процесор.
- HMI (Human-Machine Interface) - інтерфейс людина-машина.
- IoT (Internet of Things) - інтернет речей.
- IIoT (Industrial Internet of Things) - промисловий інтернет речей.
- LIDAR (Light Detection and Ranging) - вимірювання дальності за допомогою лазера.
- LSTM (Long Short-Term Memory) - довготривала короткочасна пам'ять.
- ML (Machine Learning) - машинне навчання.
- MLP (Multilayer Perceptron) – багатошаровий перцептрон.
- MQTT (Message Queuing Telemetry Transport) - протокол передачі телеметрії через чергу повідомлень.
- NC (Numerical Control) - числове керування.
- NLP (Natural Language Processing) - обробка природної мови.
- NN (Neural Network) - нейронна мережа.

OPC UA (Open Platform Communications Unified Architecture) - уніфікована архітектура відкритих платформ комунікації.

PaaS (Platform as a Service) - платформа як сервіс.

PLC (Programmable Logic Controller) - програмований логічний контролер.

PRM (Probabilistic Roadmap) - ймовірнісна дорожня карта.

PyPI (Python Package Index) - індекс пакетів Python.

RNN (Recurrent Neural Network) - рекурентна нейронна мережа.

ROS (Robot Operating System) - операційна система для роботів.

RPA (Robotic Process Automation) - роботизована автоматизація процесів.

RTOS (Real-Time Operating System) - операційна система реального часу.

SCM (Supply Chain Management) - управління ланцюгами постачання.

SDK (Software Development Kit) - комплект розробника програмного забезпечення.

SLAM (Simultaneous Localization and Mapping) - одночасна локалізація та побудова карти.

TCP/IP (Transmission Control Protocol/Internet Protocol) - протокол управління передачею/інтернет-протокол.

TTS (Text-to-Speech) - перетворення тексту в мовлення.

UAV (Unmanned Aerial Vehicle) - безпілотний літальний апарат.

UGV (Unmanned Ground Vehicle) - безпілотний наземний транспортний засіб.

UI (User Interface) - інтерфейс користувача.

URDF (Unified Robot Description Format) - уніфікований формат опису роботів.

UX (User Experience) - користувацький досвід.

VFD (Variable Frequency Drive) - перетворювач частоти.

VPS (Visual Positioning System) - візуальна система позиціонування.

VR (Virtual Reality) - віртуальна реальність.

VSLAM (Visual Simultaneous Localization and Mapping) - візуальна одночасна локалізація та побудова карти.

XR (Extended Reality) - розширена реальність.

YOLO (You Only Look Once) - алгоритм розпізнавання об'єктів у реальному часі.

ЄС – Європейський союз.

ШІ – штучний інтелект.

ВСТУП

В умовах стрімкого розвитку технологій, що спостерігається в останні десятиліття, промисловість переживає новий етап еволюції - перехід від концепції Industry 4.0 до Industry 5.0. Ця трансформація передбачає не лише інтеграцію розумних машин та систем, але й більш глибоку синергію між людиною та технологіями, що сприяє підвищенню ефективності та гнучкості виробничих процесів. У центрі цієї парадигми знаходяться колаборативні роботи-маніпулятори, здатні взаємодіяти з людьми у спільному робочому просторі, забезпечуючи при цьому високий рівень безпеки та продуктивності.

Актуальність досліджень, представлених у даній монографії, зумовлена необхідністю розробки нових підходів до динамічного опису навколишнього середовища роботів-маніпуляторів, які функціонують в умовах Industry 5.0. Важливим аспектом є забезпечення точної ідентифікації об'єктів у робочій зоні, що вимагає використання сучасних методів комп'ютерного зору та машинного навчання. В цьому контексті використання мови програмування Python, яка набула широкого поширення у сфері робототехніки, та середовища розробки PyCharm дозволяє створювати ефективні алгоритми для обробки даних та управління роботами.

Монографія охоплює питання інтеграції інтелектуальних систем в робототехніку, акцентуючи увагу на використанні бібліотек, таких як OpenCV, TensorFlow та Mediapipe, для розпізнавання та аналізу об'єктів у режимі реального часу. Дослідження спрямовані на розробку адаптивних алгоритмів, які забезпечують безперервний моніторинг та аналіз робочого середовища, що є критично важливим для безпечної взаємодії між людиною та колаборативним роботом. Особливу увагу приділено розробці програмного забезпечення для моделювання ідентифікації та трекінгу людини у робочій зоні, використовуючи сучасні методи нейронних мереж.

Дана монографія є не лише науково-дослідним матеріалом, але й практичним керівництвом для розробників, що працюють у сфері робототехніки, та здобувачів, які готуються до роботи у цій галузі. Викладені в ній методи і технології сприяють підвищенню якості проектування робототехнічних систем, орієнтованих на потреби Industry 5.0, що підвищує доцільність і значущість цього дослідження для майбутнього розвитку автоматизації та комп'ютерно-інтегрованих технологій.

Монографія може бути корисна науковцям, аспірантам, здобувачам, які займаються питаннями розробки інтелектуальних робототехнічних систем та фахівцям в галузі знань 17 - Електроніка, автоматизація та електронні комунікації за спеціальністю 174 - Автоматизація, комп'ютерно-інтегровані технології та робототехніка та освітньо-професійних програм: «Робототехніка та кіберфізичні системи», «Автоматизація та комп'ютерно-інтегровані технології» та G7 Автоматизація, комп'ютерно-інтегровані технології та робототехніка.

1. КОЛАБОРАТИВНІ РОБОТИ-МАНІПУЛЯТОРИ В INDUSTRY 5.0

1.1 Основні принципи та відмінності від Industry 4.0

Із прискоренням процесів глобалізації та ростом динамічності світового ринку компанії стикаються із необхідністю прийняття цілого ряду змін, викликаних як зовнішніми, так і внутрішніми факторами, з метою довгострокового функціонування в умовах зростаючої конкуренції [1]. Особливої уваги у даному питанні заслуговує проблема управління міжнародним бізнесом. Враховуючи, що невід'ємними факторами, характерними для сучасної реальності, є нестабільність, невизначеність, складність і неоднозначність, питання управління міжнародним бізнесом стає все більш актуальним [2]. Таким чином, необхідно розуміти основні властивості та специфіку функціонування підприємств на міжнародному ринку, усвідомлювати ключові відмінності у процесах управління такими підприємствами, а також оцінювати основні питання та проблеми даного напрямку. Глобальна економіка дозволяє компаніям у країнах, що розвиваються, отримати більш легкий доступ до нових технологій, навичок, умінь, ринків та джерел фінансування; а також більш далекосяжних перспектив росту, ніж будь-коли [3]. В той же час, це зіштовхує їх із інтенсивною конкуренцією імпортерів дешевих товарів та іноземних фірм, розташованих на місцевому рівні. З падінням торговельних бар'єрів немає такої речі, як тільки внутрішній ринок. Будь-який товар або послуга, що пропонує компанія, що розвивається, має дедалі більше відповідати стандартам ціни, якості та доставки на міжнародних ринках. Наразі людство переживає четверту промислову революцію і є свідком технологічних та економічних змін, пов'язаних з інтеграцією у діяльність таких інновацій як штучний інтелект (AI) [4], блокчейн [5], інтернет речей (IoT) [6], криптовалюти [7], засобів автоматизації [8], тощо. Природно, початкове

бачення Industry 4.0 було розроблено насамперед для промислової автоматизації та інтелектуального виробництва, проте незабаром Industry 4.0 стала глобальним трендом, деякий час вона сприймалася як насамперед актуальна для європейського бізнес-середовища та політики [9]. Однак швидкий розвиток високотехнологічних та інноваційних підходів до виробництва спонукав світову спільноту приділити більше уваги до пошуку можливих шляхів для гармонійного існування суспільства та спровокував формацію концепції Industry 5.0 як еволюцію оригінальної концепції з посиленою роллю людей, що забезпечує бізнес-цінність поряд з роботами [10-12]. Дана промислова парадигма ставить перед дослідниками, у першу чергу, завдання виявлення та систематизації конкретних рекомендацій для управління міжнародним бізнесом. Вперше про програму «Industry 4.0» мова зайшла у 2011 році на промисловій виставці в Ганновері, де уряд Німеччини поставив задачу розширити застосування інформаційних технологій у виробництві [13]. Над створенням програми модернізації промислових підприємств країни в цьому напрямку працювала високопрофесійна команда, до якої увійшли представники бізнесу і держави. Мета програми – збереження і збільшення конкурентних переваг підприємств країни. Характерні риси Industry 4.0 - це повністю автоматизовані виробництва, на яких керівництво всіма процесами здійснюється в режимі реального часу і з урахуванням мінливих зовнішніх умов [14]. Кіберфізичні системи створюють віртуальні копії об'єктів фізичного світу, контролюють фізичні процеси і приймають децентралізовані рішення. Вони здатні об'єднуватися в одну мережу, взаємодіяти в режимі реального часу, самоналагоджуватися і самонавчатися [15-17]. Важливу роль відіграють інтернет-технології, що забезпечують комунікації між персоналом та машинами. Підприємства виготовляють продукцію відповідно до вимог індивідуального замовника, оптимізуючи собівартість виробництва. Експерти виділяють чотири базових технології, в результаті впровадження яких очікуються революційні зміни.

Інтернет речей (Internet of Things, IoT), у цій технології Інтернет використовується для обміну інформацією не тільки між людьми, але і між різними «речами», тобто машинами, пристроями, датчиками, тощо [18-20]. З одного боку, «речі», забезпечені датчиками, можуть, обмінюватися даними і обробляти їх без участі людини. З іншого боку, людина може активно брати участь в цьому процесі, наприклад, коли мова йде про «розумний будинок».

Різновидом IoT є промисловий (індустріальний) інтернет речей (Industrial Internet of Things, IIoT) [21-23]. Саме він відкриває пряму дорогу до створення повністю автоматизованих виробництв. Починається все з того, що ключові компоненти обладнання забезпечуються різними датчиками, виконавчими механізмами і контролерами; зібрані дані обробляються і надсилаються до відповідних служб підприємства, що дозволяє персоналу оперативно приймати обґрунтовані і виважені рішення. Але завдання-максимум полягає в досягненні такого рівня автоматизації підприємства, при якому на всіх ділянках, де це можливо, машини працюють без участі людей. Роль персоналу при цьому зводиться до контролю роботи машин і реагування лише на екстрені ситуації [24,25].

Також важливою технологією є цифрові екосистеми. Це системи, що складаються з різних фізичних об'єктів, програмних систем і керуючих контролерів, що дозволяють уявити таке утворення як єдине ціле. Фізичні та обчислювальні ресурси в такій екосистемі тісно пов'язані, моніторинг і управління фізичними процесами здійснюється з використанням технологій IIoT [26-28]. Традиційні інженерні моделі гармонійно співіснують з комп'ютерними.

Аналітика великих даних (Data Driven Decision) або просто Великі дані (Big Data) [29,30]. Величезні обсяги інформації, що накопичуються в результаті «оцифрування» фізичного світу, можуть бути ефективно оброблені тільки комп'ютерами (в майбутньому, можливо, квантовими), із застосуванням хмарних обчислень і технологій штучного інтелекту (Artificial Intelligence) [31,32]. В результаті людина, яка контролює той чи інший процес,

ситуацію, обстановку має отримувати оброблені дані, максимально зручні для сприйняття, аналізу й ухвалення рішення.

Нова промислова парадигма, Industry 5.0, з'явилася дуже скоро після Industry 4.0 і викликала дискусію про роль і причини застосування нової парадигми. Industry 4.0 базується на концепції «розумної фабрики», де розумні продукти, машини, системи зберігання та дані об'єднуються у формі кіберфізичних виробничих систем[33-36]. У технічному аспекті Industry 4.0 покращила взаємодію між людиною та машиною, але в соціально стійкому аспекті технологічні перетворення Industry 4.0 повинні ретельно враховувати центральну роль людини. Під час пандемії COVID-19 було підкреслено роль і важливість співробітників, а сама пандемія спонукала до переосмислення парадигми Industry 4.0 [37,38]. Таким чином, ідея Industry 5.0 з'явилася як розширення Industry 4.0 із соціальним та екологічним вимірами.

З одного боку, Industry 5.0 зосереджена на навичках, знаннях і вміннях працівників співпрацювати з машинами та роботами, а з іншого - гнучкості виробничих процесів і впливі на навколишнє середовище [39,40]. Оптимізація процесів за допомогою цих досягнень суттєво вплинула на значущість людини у виконанні багатьох завдань у різних галузях, що викликало значне занепокоєння та протест серед працівників. Таким чином, п'ята промислова революція має зробити галузі більш орієнтованими на майбутнє, гармонійний розвиток та людиноцентризм. Industry 5.0 не слід розглядати як хронологічне продовження чи альтернативу існуючій парадигмі Industry 4.0. Industry 5.0 доповнює та розширює відмінні риси Industry 4.0. Вона наголошує на аспектах, які будуть вирішальними факторами при розміщенні промисловості в майбутньому європейському суспільстві. Ці фактори мають не тільки економічний чи технологічний характер, але також містять важливі екологічні та соціальні аспекти (концепція сталого розвитку). Парадигма гармонійного розвитку суспільства в Industry 5.0 представлена на рисунку 1.1 .



Рисунок 1.1 - Парадигма гармонійного розвитку суспільства в Industry 5.0 [40]

Концепція активно обговорюється з 2020 року на низці міжнародних самітів та конференцій. Основна увага приділялася технологіям, які підтримують Industry 5.0. Було досягнуто консенсусу щодо необхідності більш повної інтеграції соціальних та екологічних європейських пріоритетів у технологічні інновації та зміщення акценту з індивідуальних технологій на системний підхід. Було визначено шість категорій, кожна з яких, як вважається, розкриває свій потенціал у поєднанні з іншими як частина технологічних структур [40, 41]:

- 1) індивідуалізована взаємодія людини та машини;
- 2) біоінформаційні технології та інтелектуальні матеріали;
- 3) цифрові двійники та моделювання;
- 4) технології передачі, зберігання та аналізу даних;
- 5) штучний інтелект (AI);
- 6) технології енергоефективності, відновлюваних джерел енергії, зберігання та автономії [41].

Нова роль галузевого працівника в Industry 5.0 істотно змінюється. Працівник розглядається не як «вартість», а як «інвестиційна» позиція для компанії, що дозволяє компанії та працівникові розвиватися. Це означає, що роботодавець зацікавлений в інвестуванні у навички, здібності та благополуччя своїх співробітників для досягнення їх цілей [42]. Такий підхід дуже відрізняється від простого врівноваження витрат на робочу силу з фінансовими доходами: людський капітал більш шанується і цінується. Важлива передумова Industry 5.0 у тому, що технологія служить людям, а не навпаки [43]. Принципи постійного саморозвитку та самонавчання виходитимуть на перший план для працівників виробництва. Корпоративні культури будуть будуватися на підтримці освіти, розвитку творчого та нешаблонного мислення персоналу, який працює з роботами. Роботи підштовхуватимуть і доповнюватимуть цей нескінченний процес розвитку.

Одним з найважливіших компонентів Industry 5.0 стане інтерфейс людина-машина. Роботи навчатимуться у людей, а люди отримають вигоду від робіт, які виконують завдання, які люди не можуть або не хочуть виконувати в рамках виробничих операцій [44-46]. Поєднання людського інтелекту з когнітивними здібностями технологічно просунутого робітника – потужна комбінація, призначена для досягнення високих результатів. Отже, п'ята промислова революція станеться, коли три її основні елементи – інтелектуальні пристрої, інтелектуальні системи та інтелектуальна автоматизація – повністю зіллються з фізичним світом у співпраці з людським інтелектом. Термін «автоматизація» визначає автономних роботів як інтелектуальних агентів, що працюють одночасно з людьми в одному робочому просторі [47].

Проте наразі існують практичні питання щодо ведення бізнесу в умовах Industry 5.0. Як наголошувалося раніше, дана концепція досить чітко ґрунтується на засадах сталого розвитку (гармонійного розвитку) [47,48]. Попри той факт, що існує не лише Глобальний Договір ООН, покликаний на дотримання Цілей сталого розвитку, а й постійно відбуваються

саміти/конференції, де наголошують на необхідності імплементації ідей сталого розвитку на підприємствах, а механізм оцінки рівня сталого розвитку відсутній [49]. Особливої уваги потребує розбір технологій, що використовуються у контексті Industry 5.0.

Низка перспективних технологічних тенденцій, таких як Edge Computing (граничні обчислення) [50], Digital Twins (цифрові близнюки) [51], ІоЕ (Інтернет усього) [52], аналітика великих даних [53], коботи (cobots) [54], 6G [55] і блокчейн [56], інтегровані з когнітивними навичками та інноваціями, які можуть допомогти галузям збільшити виробництво та швидше постачати індивідуальні продукти. Ці передові технології роблять Industry 5.0 вдосконаленою моделлю виробництва з акцентом на взаємодії між машинами та людьми. Розумні машини розроблені для спільної роботи з людьми, і ця спільна робота робить людські здібності більш продуктивними, надзвичайно простими в автоматизації для окремих осіб і малих підприємств, ніж будь-коли раніше.

Важливою концепцією п'ятої промислової революції став ІоЕ. ІоЕ є поняттям, яке поєднує різні аспекти технологій та підходів, щоб створити зв'язану інтернет-екосистему [57]. У контексті Industry 5.0 ІоЕ означає розширення концепції ІоТ та включення в неї людей, процесів та інтелектуальних систем. В Industry 5.0 з'єднані пристрої ІоТ, люди, машини та системи, щоб створити спільну мережу, в якій вони можуть спілкуватися, обмінюватися даними та співпрацювати для досягнення спільних цілей [58]. Основна ідея полягає в тому, що спільна робота розумних систем, людей та машин призводить до більш ефективних та інноваційних рішень. ІоЕ в Industry 5.0 включає в себе використання передових технологій, таких як [59]: штучний інтелект (AI), аналітика даних, блокчейн, розширена реальність (AR) та віртуальна реальність (VR). Ці технології сприяють збору, обробці та аналізу великих обсягів даних, розумінню контексту та прийняттю інформованих рішень. Наприклад, в розумному місті системи управління рухом можуть взаємодіяти з автомобілями та пішоходами, аналізувати дані з

сенсорів та забезпечувати оптимальну організацію руху [60]. Таким чином, з вищесказаного можна зрозуміти, що на відміну від Industry 4.0, яка фокусується на автоматизації та зменшенні людської участі, Industry 5.0 прагне поєднати переваги як людини, так і машин [61].

Загальні відмінності Industry 4.0 та Industry 5.0 представлені у вигляді таблиці порівняння 1.1 [40].

Таблиця 1.1 - Загальні відмінності Industry 4.0 та Industry 5.0 [40, 62-66]

Industry 4.0	Industry 5.0	Специфікація відмінностей
IoT (Internet of Things)	IoE (Internet of Everything)	Industry 4.0 базується на «Інтернеті речей» (IoT) та хмарних обчисленнях, а Industry 5.0 базується на «Інтернеті усього» (IoE)
Роботи	Коботи	Industry 4.0 орієнтована на оптимізацію виробничих процесів, а Industry 5.0 - на оптимізацію всього ланцюжка створення вартості.
Людино-орієнтовна	Людино-центрична	
Оптимізація виробничих процесів	Оптимізація всього ланцюжка створення вартості	
Кібер-фізичні системи	Оптимізаційні системи	Industry 4.0 зосереджена на використанні кіберфізичних систем, а Industry 5.0 - на використанні автономних систем
Big Data та аналітика	Прогностична аналітика та машинне навчання	Industry 4.0 зосереджена на використанні великих даних і аналітики, тоді як Industry 5.0 зосереджена на використанні прогностичної аналітики та машинного навчання.

Як зазначалося раніше, відповідно до Європейської комісії, прийдешня п'ята промислова революція можлива лише при збалансованій успішній імплементації трьох категорій: гнучкості, людино-центричності та сталості. Відтак, відповідно до цих категорій, технології, що використовуються в контексті Industry 4.0 та Industry 5.0, мають наступні фокуси, які представлено у таблиці 1.2 [40, 67-71].

Таблиця 1.2 - Зміна фокусу технологій в контексті Industry 4.0 та 5.0

	Фокус	Industry 4.0	Industry 5.0
Адаптивність	Технології	Робототехніка, IoT, AI, Big Data	IoE, Advanced robotics, AI, людини та машини
	Ключові драйвери	Підвищення ефективності та зменшення витрат	Соціальна та екологічна сталість
	Ланцюги постачання	Цифровізовані та взаємопов'язані	Інтегровані та локалізовані
Людино-центричність	Роль людини	Зменшення людської участі (моніторинг, обслуговування, контроль)	Поліпшення можливостей людини та її участі (співтворці, розв'язувачі проблем та приймачі рішень)"
	Добробут працівників	Не є пріоритетом	Фокус на покращенні благополуччя та задоволеності працівників
Сталість	Продукти	Виготовлені в масових масштабах та стандартизовані	Індивідуалізовані та персоналізовані
	Сталий розвиток	Не є пріоритетом	Акцент на сталість та соціальну відповідальність

Перехід від Industry 4.0 до Industry 5.0 вимагає вирішення певної проблематики та викликів, пов'язаних зі змінами ланцюгів постачання. Деякі з цих проблем передбачають більш глибоку інтеграцію між різними елементами виробничих процесів та ланцюгів постачання, кібербезпеки, масштабуванні та пропускній здатності. Таким чином, управління проєктами з цифрової трансформації міжнародних підприємств може залежати від обраного підприємством фокусу: відмінність технологій, ключові драйвери, роль людини, добробут працівників, розвиток продуктів, сталий розвиток [40].

Відповідно до стратегії підприємства, проєкти з цифрової трансформації можуть бути спрямовані на такі напрями: соціальна та екологічна сталість, створення інтегрованих та локалізованих ланцюгів постачання, підвищення мотивації та задоволеності працівників, індивідуалізація та персоналізація готового продукту, тощо [72,73]. Вирішенням таких проблем слугують технології. Насамперед блокчейн, Digital Twins та Cloud Supply. Технологія блокчейн може позитивно вплинути на формування та зміну ланцюгів постачання та цінності при переході від Industry 4.0 до Industry 5.0. Вона

забезпечує безпеку, прозорість та недоступність до змін інформації, що передається між учасниками ланцюга постачання. Одним із ключових при використанні блокчейну в ланцюгах постачання є відстеження походження товарів. Так можна створити неперервний, незмінний журнал транзакцій, який дозволить точно відстежувати кожен етап виробництва та постачання товару. Це дозволить уникнути підробок, підвищить довіру між сторонами та забезпечить високу якість продукції. Блокчейн може покращити ефективність фінансових транзакцій у ланцюгах постачання. Використовуючи "смарт контракти" на базі блокчейну, можна автоматизувати процеси оплати, уникнути затримок та помилок, що часто виникають при традиційних фінансових операціях [74-76]. З метою підвищення ефективності імплементації технології блокчейн на підприємствах, було проведено аналіз використання різного програмного забезпечення провідними компаніями у різних галузях. Відповідно до мети впровадження та очікуваного функціоналу, пропонується використання відповідного програмного забезпечення, які представлені у таблиці 1.3 [40, 77-80].

Таблиця 1.3 - Кейси використання блокчейну у міжнародному бізнесі

Індустрія	Компанія	Мета впровадження	Функції	Програмне забезпечення
Технологічна	IBM	Управління ланцюгом поставок, торгове фінансування	Управління ланцюгами поставок та активами	IBM Blockchain Platform
Роздрібна торгівля	Walmart, Walmart China	Безпека харчових продуктів, управління ланцюгами поставок	Управління ланцюгами поставок, підвищення безпеки	IBM Food Trust, VeChain
Логістика	Maersk, FedEx	Управління ланцюгом поставок, відстеження вантажів	Управління ланцюгами поставок	IBM Blockchain Platform, Chainyard

Продовження таблиці 1.3

1	2	3	4	5
Фінанси	Barclays, Santander, Mastercard, JPMorgan Chase, China Construction Bank	Торгове фінансування, перевірка КУС, платежі, грошові перекази	Управління активами, підвищення рівня безпеки	Corda, Ripple, Mastercard Blockchain, Onyx, Hyperledger Fabric
Автомобільна	Porsche	Відстеження історії автомобіля, управління ланцюгом поставок	Управління ланцюгами поставок	XAIN
Аерокосмічна	Airbus	Управління ланцюгом поставок, відстеження технічного обслуговування	Управління ланцюгами поставок	Hyperledger Fabric

Це дозволить забезпечити швидку та безпечну оплату за товари та послуги у ланцюгах постачання. Європейська комісія визнала потенціал технології блокчейн у контексті Industry 5.0 та активно просуває її розвиток та впровадження. У зв'язку з цим, Європейська комісія висловила зацікавленість у потенціалі технології блокчейн у контексті Industry 5.0 у звіті під назвою «Блокчейн для промислових перетворень». Зокрема, Європейська комісія виділила наступні можливі застосування технології блокчейн у Industry 5.0 [40, 81,82]:

1. Управління ланцюгом поставок: створення безпечних та прозорих ланцюгів поставок, що забезпечить більшу видимість та можливість відстеження продуктів.

2. Управління інтелектуальною власністю: створення незмінних та недоступних для змін документів щодо прав на інтелектуальну власність, для більшої ефективності та прозорості в управлінні патентами, товарними знаками та авторськими правами.

3. Децентралізовані автономні організації (DAO): створення DAO, організацій, які керуються кодом, а не людьми. Це може забезпечити нові

форми співпраці та децентралізованого прийняття рішень в контексті Industry 5.0.

4. Управління цифровою ідентичністю: створення безпечних та децентралізованих систем цифрової ідентичності, для більшої приватності та контролю персональних даних. Загалом, Європейська комісія бачить блокчейн як ключовий інструмент концепції Industry 5.0, що дозволяє забезпечити більшу співпрацю, прозорість та ефективність в різноманітних застосуваннях. ЄС також виділяє кошти на блокчейн-проекти та дослідження через різні програми та ініціативи («Горизонт 2020» та Європейське блокчейн-партнерство).

Cloud Supply є ще однією технологією, яка може мати позитивні фактори впливу на ланцюги постачання та цінності [40, 83-85]. Її використання дозволяє підприємствам отримувати доступ до обчислювальних ресурсів, програмного забезпечення та даних через Інтернет. Це забезпечує більшу гнучкість та масштабованість управління ланцюгами постачання. Підприємства можуть легко змінювати масштаби своїх операцій, залежно від попиту та потреб ринку. Компанії можуть також отримувати в режимі реального часу інформацію про стан запасів, прогнозувати попит та координувати роботу з постачальниками. Cloud Supply сприяє покращенню співпраці між учасниками ланцюга постачання. Завдяки спільному доступу до хмарних платформ та даних, компанії можуть легко обмінюватись інформацією, спільно планувати ресурси та координувати свої дії, що дозволяє зменшити затримки і помилки, покращити взаєморозуміння та забезпечити більш ефективну роботу усього ланцюга постачання. Кейси використання Cloud supply у міжнародному бізнесі наведені в таблиці 1.4 [40, 86-90].

Таблиця 1.4 - Кейси використання Cloud supply у міжнародному бізнесі

[40]

Індустрія	Компанія	Мета впровадження	Функції	Програмне забезпечення
Їжа та напої	Coca-Cola, Nestle, McDonald's, PepsiCo	Управління ланцюгом поставок, оптимізація запасів та логістики	Управління ланцюгами поставок	SAP Ariba, Anaplan, JDA Software
Споживчі товари	L'Oreal, Procter & Gamble, Colgate-Palmolive	Управління закупівлями, співпраця з постачальниками, планування попиту	Управління ланцюгами поставок	SAP Ariba, Anaplan
Електронна комерція	Amazon	Управління ланцюгом поставок, оптимізація запасів	Управління ланцюгами поставок	Oracle Supply Chain Management
Меблі	ІКЕА	Управління ланцюгом поставок, оптимізація логістики	Управління ланцюгами поставок	Oracle SupplyChain Management

Технологія Digital Twins полягає у створенні віртуальних моделей фізичних об'єктів та процесів, що відбуваються у реальному світі [40, 91,92]. Застосування Digital Twins дозволяє учасникам ланцюга постачання моніторити, аналізувати та оптимізувати роботу системи в реальному часі. Віртуальна модель дозволяє прогнозувати різні сценарії та експериментувати з оптимальними рішеннями. Наприклад, Digital Twins може бути використано для прогнозування попиту на продукцію, виявлення несправностей у процесах виробництва та управління якістю продукції. Використання Digital Twins дозволяє покращити управління запасами, забезпечити більш точне прогнозування попиту та забезпечити високу якість продукції. Віртуальна модель також сприяє швидшій взаємодії між учасниками ланцюга постачання та дозволяє швидше реагувати на зміни ринкових умов. Кейси використання Digital Twins у міжнародному бізнесі наведені в таблиці 1.5 [40, 93-96]

Таблиця 1.5 - Кейси використання Digital Twins у міжнародному бізнесі

[40]

Індустрія	Компанія	Мета впровадження	Функції	Програмне забезпечення
1	2	3	4	5
Виробництво	Siemens, Caterpillar, General Electric	Прогнозне обслуговування, оптимізація виробництва Обслуговування та оптимізація обладнання Управління активами, прогнозне обслуговування	Підвищення ефективності виробництва Управління активами	Siemens Digital Twin ANSYS Twin Build Perredix Digital Twin
Автомобільна	Porsche, BMW	Дизайн та моделювання транспортних засобів Оптимізація виробництва, управління ланцюгами поставок	Управління ланцюгами поставок Підвищення ефективності виробництва	Siemens Digital Twin
Аерокосмічна	Boeing, Rolls-Royce, NASA, Airbus, Safran	Проектування та моделювання літаків або космічних апаратів. Проектування двигунів та оптимізація продуктивності	Підвищення ефективності виробництва	ANSYS Twin Builder Dassault Systèmes 3DEXPERIENCE
Енергетична	Schneider Electric, Total, EDF	Оптимізація електростанції, управління активами Оптимізація нафтової вишки, управління безпекою Оптимізація АЕС, управління безпекою	Підвищення ефективності виробництва Підвищення рівня безпеки Управління активами	AVEVA Digital Twin

Продовження таблиці 1.5

1	2	3	4	5
Теле-комунікації	Deutsche Telekom	Оптимізація мережі, управління активами	Підвищення ефективності виробництва Управління активами	SAP Digital Twin
Технологічна	Intel	Оптимізація дата-центрів, управління активами	Управління активами	Predix Digital Twin

З метою узагальнення, дану вище інформацію можна представити у вигляді рисунку 1.2 [40].

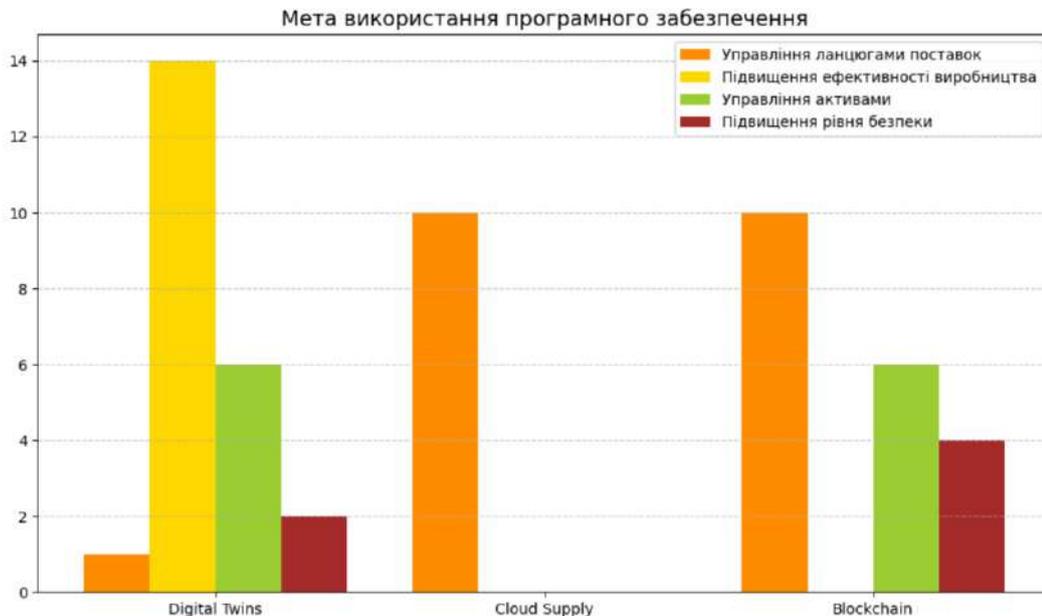


Рисунок 1.2 - Використання програмного забезпечення відповідно до функціонального навантаження [40]

Як показано на рисунку 1.2, основна роль використання різного програмного забезпечення Cloud Supply - управління ланцюгами поставок (оптимізація запасів, покращення процесу комунікації з постачальниками, скорочення строків поставок, підвищення ефективності використання складів). Програмне забезпечення Digital Twins, у свою чергу, більш націлено на підвищення ефективності виробництва (прогнозне обслуговування,

покращення процесів створення дизайну та моделювання транспортних засобів, проєктування, оптимізація обладнання, тощо). Використання програмного забезпечення блокчейн не лише впливає на управління ланцюгами поставок, а й допомагають в процесах управління активами та підвищення рівні безпеки (як людей, так і даних/інформації) [40].

Загалом, Industry 5.0 (у контексті продовження Industry 4.0) спонукає компанії використовувати нове програмне забезпечення з метою мінімізації логістичних витрат (строки поставок, ефективність використання складів, тощо), максимізації ефективності виробничих процесів (підвищення якості готового продукту, прискорення процесів розробки та моделювання продукту, зменшення часу виробництва, тощо), підвищення рівня безпеки (безпека працівників, безпека продукції для споживачів, безпека персональних даних, тощо) та підвищення ефективності управління активами. Таким чином, узагальнюючи вищезгадані порівняльні аналізи ключових характеристик, спільних та відмінних рис Industry 4.0 та Industry 5.0, а також відповідність використання певного програмного забезпечення згідно з метою впровадження інновацій та очікуваного функціонального навантаження, можна зробити висновки стосовно переходу від четвертої до п'ятої промислової революції. Як результат, можна стверджувати, що Industry 5.0 є надбудовою, що продовжує концепцію Industry 4.0, з урахуванням підвищення значущості ролі соціальної відповідальності і сталого розвитку, використання гнучких та адаптивних технологій для управління ланцюгами постачання у міжнародному бізнесі представлені в таблиці 1.6 [40, 97-102].

Таблиця 1.6 - Використання гнучких та адаптивних технологій для управління ланцюгами постачання у міжнародному бізнесі [40]

		Функції		Мета	
		Industry 4.0 (IoT)	Industry 5.0 (IoE)	Industry 4.0 (IoT)	Industry 5.0 (IoE)
Ланцюг поставок	Digital Twins	Створення віртуальних копій фізичних активів	Створення віртуальних копій фізичних активів і працівників	Оптимізація промислових процесів за допомогою передових технологій	Створення більш стійкої та соціально відповідальної галузі
		Оптимізація виробничих процесів і якості продукції	Оптимізація продуктивності людини та персоналізоване навчання		
		Створення розумних фабрик, прогнозне обслуговування та моделювання	Створення робочого середовища для співпраці, взаємодії людини і машини		
		Створення віртуальних копій машин, виробничих ліній, фабрик	Створення віртуальних копій робітників, моніторинг охорони праці		
	Cloud Supply	Інтеграція цифрових технологій і аналіз даних	Співпраця та людиноорієнтований підхід до управління ланцюгом поставок	Створення розумних, взаємопов'язаних і гнучких ланцюгів поставок	Створення стійких та соціально відповідальних ланцюгів поставок
		Покращена маневреність, чуйність та інноваційність	Екологічна та соціальна відповідальність		
		Прогнозне обслуговування, прогнозування попиту та 3D-друк	Циркуляційні ланцюжки поставок, відповідальне постачання та етичне виробництво		
	Блокчейн	Інтеграція блокчейну з IoT, III та ланцюгом поставок	Співпраця та людиноорієнтований підхід до управління системою	Безпечний і ефективний обмін даними та транзакції	Створення стійких та соціально відповідальних систем
		Покращена безпека даних, прозорість і ефективність	Екологічна та соціальна відповідальність		
		Управління ланцюгом поставок, відстеження активів і платежі	Сталі фінанси, циркулярна економіка та етичне управління		
		Прозорість ланцюжка поставок, відстеження активів і платежів	Відповідальне інвестування та зелені облигації		

Відтак, міжнародні компанії, що мають намір запускати проекти з цифрової трансформації, можуть одразу орієнтуватися на функції, які концептуально відносяться до Industry 5.0, адже як зазначалося раніше, Industry 5.0 є надбудовою та поліпшеною версією Industry 4.0. Наприклад, імплементація технології Digital Twins, що є властивою для обох Industry, надають змогу компаніям не лише створити віртуальні копії фізичних активів (Industry 4.0), а й додатково створити віртуальні копії працівників (Industry 5.0). Окремо слід зауважити, що контекст Industry 5.0 напряду вплине на вже існуючі бізнес-моделі міжнародних підприємств, а специфіка цифрової трансформації матиме особливо значний вплив на високотехнологічні підприємства.

1.2 Екосистемна модель концепції Industry 5.0

Таким чином, можна зробити висновок, що еволюція технологій Industry 5.0 відрізняється від цифрової революції Industry 4.0 тим, що трансформуює людську працю у спільну діяльність роботів і креативних особистостей з метою продукування кастомізованих продуктів і послуг, які відповідають смакам, побажанням та фінансовим можливостям споживачів[103–105]. Теоретичні та емпіричні дослідження вчених дають змогу стверджувати, що Industry 5.0 є всеосяжною й універсальною та володіє прогресуючим потенціалом розвитку, тобто екосистему модель концепції Industry 5.0 можна приставити на рисунку 1.3 [103, 106-110].

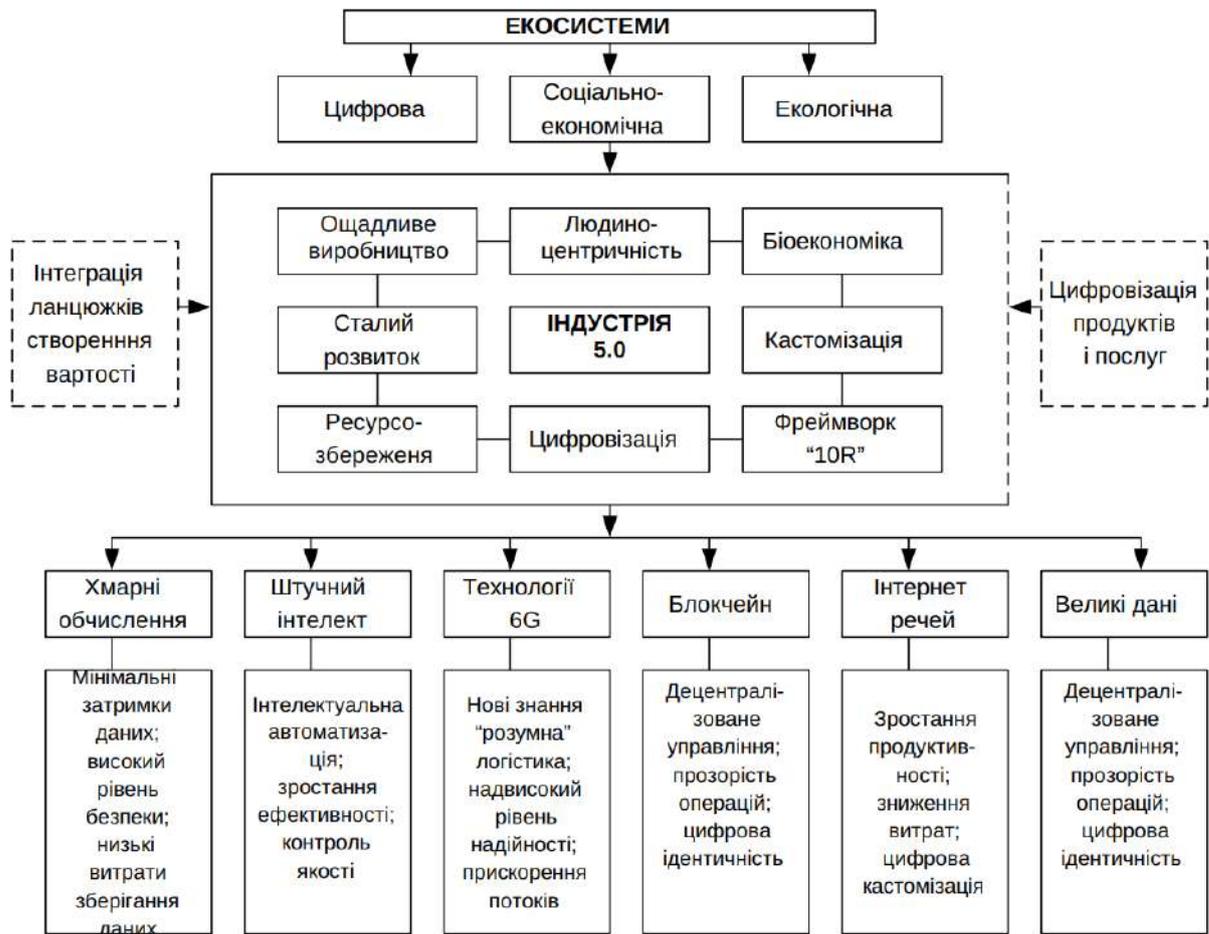


Рисунок 1.3 – Екосистема концепції Industry 5.0 [103]

Методологія Industry 5.0 як відносно нового напрямку має міждисциплінарний характер та передбачає диференційоване за інструментарієм аналізу дослідження економічного, соціального, інформаційного та екологічного глобалізму. Особливо важливо при такому дослідженні враховувати масштаби та динаміку процесу становлення системи глобального управління ресурсами планети і перерозподілом світового доходу [103, 111]. Industry 5.0 стає, з одного боку, джерелом і стимулятором конкуренції, надаючи нові можливості розвитку, а з другого – породжує суперечності і виклики локального й загально цивілізаційного масштабів. На наш погляд, у теоретико-методологічному плані має місце системна асиметричність соціально-еколого-економічного розвитку, породжена зростаючою взаємозалежністю між суб'єктами господарювання макро- та

мікрорівнів. Системне дослідження витоків та проявів п'ятої промислової революції передбачає аналіз основних тенденцій, які характеризують Industry 5.0 [103].

Перша тенденція передбачає, що Industry 5.0 буде кількісно охоплювати всі країни, а якісно – всі сфери людської життєдіяльності. П'ята промислова революція передбачає співпрацю між людьми та інтелектуальними системами, такими як роботи. При цьому йдеться про роботів, які допомагають людям працювати краще і швидше, використовуючи передові технології: Інтернет речей, великі дані, технології віртуальної і доповненої реальності. Люди і системи діятимуть як партнери, а не як конкуренти. На цьому етапі розвитку промисловості автоматизовані механізми виконуватимуть монотонні, повторювані завдання, а людські ресурси нести будуть відповідальність за творчу складову, здійснюватимуть контроль за управлінськими системами і рівнем якості виробництва. Отже, метою Industry 5.0 є синтез когнітивних обчислювальних здібностей з людським розумом і винахідливістю у комплексних виробничих, управлінських і бізнес-процесах [103, 112].

Другою тенденцією є факт, що Industry 5.0 буде ґрунтуватися на біоекономіці. Оптимальне використання біологічних ресурсів у виробничих цілях допоможе досягти балансу між екологією, промисловістю, суспільством та економікою. Згідно з визначенням Європейської комісії, біоекономіка - це виробництво відновлюваних біологічних ресурсів, а також утилізація матеріалів і потоків відходів з метою виготовлення продуктів з доданою вартістю: продуктів харчування, кормів для тварин, речовин на біологічній основі та біоенергії [103, 113-115]. Без сумнівів, біоекономіка матиме вирішальне значення для сталого розвитку економіки та фундаментальних змін у різних галузях господарської діяльності.

Третя тенденція Industry 5.0 полягає в ефективному використанні людських ресурсів та автоматизованих механізмів у синергетичній єдності з навколишнім середовищем [103, 116]. Крім того Industry 5.0 фокусується на

мінімізації відходів і домінуючій ролі людського фактору у виробничому процесі. Масштаб економічних, соціальних та екологічних проблем, з якими зіштовхується людство останнім часом, робить дедалі необхіднішим звернення до поняття “циркулярна економіка”, що передбачає перехід від лінійних систем до циклічних шляхів ощадливого використання ресурсів та розвиток відновлювальної енергетики.

Четверта тенденція Industry 5.0 стосується людиноцентричної парадигми [103, 117-120]. Це передбачає крупномасштабні інвестиції в людський капітал, інтелектуальне «насичення» масових професій, створення сприятливих умов для всебічного, творчого розвитку людини. Наслідком цього буде зростання продуктивності праці в різних сферах господарської діяльності; нарощення інтелектуального потенціалу та стійкі теми соціально-еколого-економічного зростання; перманентна модернізація структури економіки відповідно до змін кон'юнктури на світових ринках.

П'ята тенденція Industry 5.0 відображає інклюзивний розвиток як скорочення дисбалансів і рівномірний розподіл соціально-економічних благ серед усіх верств населення при умові збереження навколишнього середовища [103, 121-123]. Основою інклюзивного розвитку є соціальна справедливість, захист прав і свобод людини та екологізація діяльності. На думку багатьох зарубіжних дослідників, розробка моделі інклюзивного розвитку сприятиме виведення світової економіки з регресу та стагнації, а також передбачатиме стимулювання екологічно узгодженого соціально-економічного зростання суб'єктів господарювання мікро- та макрорівнів.

Актуалізується потреба в наукових підходах до розробки збалансованих бізнес-моделей функціонування підприємств, які б враховували узгодженість соціальних, економічних та екологічних аспектів Industry 5.0. У контексті п'ятої промислової революції ціннісна складова є пріоритетним елементом феномену бізнес-моделі. При цьому консолідуючою основою стає концепція ланцюга спільного створення і розподілу цінності. Показово, що процес

створення цінності більше не ідентифікується в межах одного підприємства – навпаки, він здійснюється у корпоративній взаємодії постачальників, виробників, каналів розподілу, інших учасників господарської діяльності. Загалом у конструктивному компоненті Industry 5.0 очевидним є прагнення до економічної рівності та справедливості, формування конкурентної співпраці, а не конкурентної боротьби. Це, в свою чергу, дає змогу стверджувати про якісну трансформацію складових бізнес-моделі. До них належать три елементи [103, 124-126]:

- пропозиція цінності (“value proposition”);
- створення цінності (“value creation”) і донесення цінності до споживача (“value delivery”);
- привласнення цінності у вигляді доходу (“value capture”).

Стрімке поширення технологій Industry 5.0 (насамперед Інтернету речей, великих даних, адитивного виробництва, робототехніки) обумовлюють зміну стратегічного бачення підприємств, формування дієвих бізнес-моделей на засадах інтелектуалізації, соціалізації та екологізації. Відомий німецький економіст, президент Всесвітнього економічного форуму у Давосі Klaus Martin Schwab детермінує Industry 5.0 як «нові технології, що об’єднують фізичний, цифровий і біологічний світи, що впливають на всі дисципліни, економіки та галузі» [103, 127]. Хоча п’ята промислова революція не має всебічного обґрунтування та єдиного визначення, більшість наявних підходів вказують на те, що Industry 5.0 тісно пов’язана з використанням цифрових технологій з метою оптимізації та інтеграції виробничих процесів. Завдяки Industry 5.0 підприємства мають можливість суттєво диверсифікувати або навіть повністю змінити концепцію свого бізнесу. На думку Р. Стрендж і А. Зухелла, підприємства, що входять до ланцюгів створення цінності, мають переосмислити, скоригувати або повністю змінити свої бізнес-стратегії та бізнес-моделі, щоб підготуватися до масштабної трансформації підприємницького середовища, спричиненої Industry 5.0 [103, 128]. У таблиці

1.7 представлено основні характеристики Industry 5.0, які передбачають зміни в традиційних бізнес-моделях підприємств [103, 129-132].

Таблиця 1.7 - Детермінанти прогресу бізнес-моделей відповідно до Industry 5.0 [103]

Особливості Industry 5.0	Основні виклики, які впливають на традиційну бізнес-модель підприємства	Вимоги до трансформації підприємства
<p>Функціональна сумісність; Децентралізація процесу прийняття рішень; Можливість роботи в режимі реального часу; Клієнтоцентричний підхід; Віртуалізація; Модульність</p>	<p>Створення мереж і зниження бар'єрів; Автоматизація; Глобалізація і децентралізація виробництва; Впровадження інтегрованих виробничих систем; Забезпечення гнучкості та персоналізація; Зниження витрат; Поява "розумних" товарів і послуг; Фрагментація ланцюжків створення цінності; Інтелектуалізація виробництва; Розвиток компетенцій людських ресурсів</p>	<p>Стандартизація; Впровадження нових бізнес-моделей; Ефективна організація праці; Захист ноу-хау; Наявність висококваліфікованих працівників; Професійний розвиток працівників; Інвестиції в НДДКР; Удосконалення внутрішньої нормативної бази</p>

Для вивчення впливу Industry 5.0 (насамперед пов'язаною з нею цифровою трансформацією підприємства) на впровадження прогресивних бізнес-моделей може бути використано підхід, запропонований А. Остервальдером та І. Пінье, відповідно до якого «бізнес-модель деталізує як організація, створює, доносить до споживача і привласнює цінність» [103]. Таким чином, тенденція до динамічних змін бізнес-моделей відповідно до Industry 5.0 зумовлює поряд із селективними методами, спрямованими на визначення пріоритетів сталого розвитку, використовувати різновекторні заходи з метою результативного процесу формування цінності. Кожен етап демонструє зміни у ланцюгу створення цінності (враховуються ключові види діяльності, ресурси та партнерські відносини), донесення цінності до споживача (охоплює пропонований продукт і послуги, канали розподілу та продажу, комунікації та PR, споживчі сегменти і партнерські відносини), а також привласнення цінності (витрати і доходи, отримані підприємством). Варто наголосити, у контексті перспектив Industry 5.0 бізнес-модель є

прогресивною концепцією із превалюванням ціннісної складової, що здійснює позитивний вплив на споживачів, підприємство, соціальне та екологічне середовище. Ціннісно-орієнтований підхід до бізнес-моделювання передбачає прояв корпоративної соціальної відповідальності в умовах загальної цифровізації та глобальної конкуренції, коли економіка має стати не тільки циркулярною, а відновлювальною, ощадливою та зберігати біорізноманіття навколишнього середовища [103].

1.3 Вплив Industry 5.0 на розвиток робототехніки

Сучасна промисловість переживає прискорення змін у виробничих парадигмах, дедалі більше зростає попит на гнучкість і кастомізацію продукції. Щоб задовольнити ці потреби, важливо мати гнучку виробничу систему. Jain та ін. [133] зазначають, що не існує єдиного визначення гнучкої виробничої системи; зокрема, вони розрізняють адаптивний підхід, який реагує на обставини, та проактивний підхід, який планується заздалегідь. Незалежно від цих категорій, очевидно, що адаптивність виробничої системи має вирішальне значення, особливо для малих і середніх підприємств (МСП). Це породило потребу в більш гнучких виробничих системах і складних продуктах з високим ступенем варіативності. Gustavsson та ін. [134] додають, що часто необхідно заздалегідь обирати між продуктивністю та гнучкістю під час проєктування системи, пропонуючи методологію для цього визначення.

Автоматизація відіграла ключову роль у цій еволюції. Як зазначає Jovane [135], гнучкі виробничі системи уможливили масову кастомізацію. Однак традиційні виробничі рішення можуть бути недостатніми для задоволення нових вимог, що вимагає більшого акценту на гнучкості та реконфігурованості, як пропонує Urbani [136]. У цьому контексті промисловість повинна не лише адаптуватися до змін, але й передбачати їх,

інтегруючи системи, які є одночасно ефективними та здатними розвиватися відповідно до ринкових тенденцій.

Згодом, як підкреслює Mourtzis [137], еволюція виробничих систем, зумовлена зміною потреб клієнтів і технологічним прогресом, призвела до переходу від функціональних парадигм до парадигм, орієнтованих на клієнта (User-Centered Design, UCD). У [138] Chammas та ін. досліджують фундаментальні концепції UCD, підкреслюючи зростаючу важливість управління проєктами через належний баланс бюджету, часу та якості, необхідний для більш персоналізованого виробництва. Використання коботів (колаборативних роботів) у промисловості може сприяти покращенню цієї тріади елементів (якість, час і бюджет).

Колаборативні роботи, або коботи, стають вирішальними помічниками в цьому контексті, не лише допомагаючи зменшити фізичне навантаження на працівників, але й оптимізуючи ефективність процесу завдяки високій здатності до реконфігурації та створенню оптимізованих робочих траєкторій [139].

Як підкреслюють Fager та ін. у роботі [140], у випадку сортування об'єктів у системі комплектування коботи можуть зменшити витрати та покращити продуктивність (час та бюджет), коли потрібно виконати значний обсяг сортувальних робіт. Ель Makrini та ін. у роботі [141] демонструють, як інтеграція коботів і людей при складанні автомобілів призводить до поліпшення якості процесу, поєднуючи людську спритність і вміння вирішувати проблеми з точністю і силою коботів. Ще одне дослідження в цьому напрямку було проведено Safeea та ін. в [142], де стверджується, що кобот може виступати в ролі «третьої руки-помічника», яка піднімає і утримує деталі, поки людина виконує завдання зі складання. Таким чином, використання коботів може покращити умови праці та продуктивність, а також знизити ризики безпеки.

Хоча деякі дослідження підкреслюють переваги використання роботів (як колаборативних, так і промислових) у виробничих процесах порівняно з повністю ручними рішеннями, детального порівняння переваг інтеграції колаборативних роботів порівняно з промисловими все ще бракує. Спроба оцінити, яке рішення є кращим між колаборативними та промисловими роботами, була зроблена в статті Barravecchia та ін. [143]. У цьому дослідженні представлена методологія оцінки оптимальної компоновки, особливо для індивідуального виробництва, при використанні колаборативних роботів у складанні. У запропонованій Барравеккіа моделі також враховуються витрати, пов'язані з навчанням і часом складання. Процес навчання відбувається швидше в колаборативних рішеннях, ніж в промислових. Дослідження показує, що рішення з використанням коботів є ідеальним в умовах низьких обсягів виробництва, оскільки вони можуть знизити частоту дефектів і не вимагають перерозподілу або звільнення працівників. У дослідженні Heredia та ін. [144] проводиться порівняння між промисловими роботами (ПР) і роботами для спільної роботи (коботами), зосереджуючи увагу на поведінці енергоспоживання електронних компонентів (ЕК), і показує, що хоча промислові роботи споживають більше енергії, більша її частина використовується для обробки вантажів, на відміну від коботів, які споживають велику частку енергії для живлення електронних компонентів, хоча в абсолютному вираженні коботи загалом споживають менше енергії. Це спостереження може вказувати на потенційний недолік інтеграції коботів у промислове середовище. На відміну від промислових роботів, які переважно використовують енергію для безпосереднього виконання завдань, наприклад, переміщення важких вантажів, коботи витрачають значну частину своєї енергії на живлення електронних компонентів (ЕС). Наприклад, кобот може витрачати значну частину енергії лише на підтримку роботи своїх датчиків і систем управління, навіть якщо він не маніпулює об'єктами.

Якщо порівнювати промислових роботів з колаборативними з точки зору гнучкості виробництва, то колаборативні рішення мають численні переваги над промисловими роботами. Колаборативна робототехніка значно підвищує гнучкість виробництва, як показують деякі дослідження. В огляді літератури Keshvarparast та ін. [145] автори повідомляють, що на етапі проектування коботів гнучкість вважається ключовою характеристикою, так само як і безпека. Вони визначають два типи гнучкості: «Гнучкі коботи» (наскільки швидко робот може бути перепрограмований) і «Гнучка співпраця» (скільки завдань робот може виконати за певний час). Крім того, є кілька робіт, які говорять про важливість коботів для гнучкого виробництва, наприклад, Giberti та ін. у [146] визначають гнучкість як здатність системи швидко переналаштовуватися, щоб адаптуватися до нового продукту в межах одного сімейства продуктів. Автори пропонують підхід до спрощення програмування колаборативних роботів, який називається «Інтерактивне уточнююче програмування» (IRP - Interactive Refinement Programming). Цей підхід базується на примітивах і загальних навичках, розроблених інженерами-експертами, які потім можуть бути з'єднані в деревоподібну структуру для генерації конкретного завдання.

Lee та ін. у роботі [147] підкреслюють, що для більшої гнучкості виробництва необхідна тісна співпраця між людьми та роботами. Автори пропонують виробничу структуру, спеціально розроблену для такої співпраці, демонструючи її доцільність. Це особливо важливо в контексті Четвертої промислової революції, коли вимоги клієнтів різноманітні і швидко змінюються, про що також свідчить робота Sherwani та ін. [148]. Strassmaier та ін. [149] також підкреслюють важливість прийняття працівників, чому може сприяти надання більшої гнучкості та врахування просторових обмежень у співпраці. Ці висновки в сукупності підкреслюють роль колаборативної робототехніки в підвищенні гнучкості виробництва. Крім того, Отман та ін. [18] підкреслюють, що співпраця людина-робот (HRC) стала помітною рисою

інтелектуального виробничого середовища, і проводять систематичний огляд нових технологій, які можуть допомогти в системі HRC, таких як штучний інтелект, колаборативні роботи, доповнена реальність і цифровий двійник, надаючи уявлення про те, як слід вирішувати цю тему. Схожу роботу виконують Michalos та ін. [150], де автори мають на меті представити існуючі підходи до впровадження додатків для спільної роботи людей та висвітлити тенденції до досягнення безшовної інтеграції та роботів як колег на фабриках майбутнього.

Коботи ідеально підходять для цілої низки промислових застосувань. Скрізь, де б їх не використовували, їхня міцність, точність і повторюваність чудово доповнюють людські навички.

На підготовці та складальних лініях вони здатні замінити людей, які стоять на ногах і тримають важкі компоненти. Там же, як правило, є застосунки для обслуговування машин, які безумовно вимагають функціонера. При порівнянні принаймні вони не потребують охорони здоров'я.

Усуваючи потребу в додатковій робочій станції, колаборативні роботи також забезпечують ефективний спосіб проведення безперервних перевірок контролю якості як частини загального процесу обробки.

Вони також чудово справляються з виконанням трудомістких, повторюваних, брудних та складних обробних завдань на пакувальних та палетувальних лініях, що не подобається людям, але цілком можливо впоратися з цим у спеціальній роботизованій комірці.

Незалежно від того, чи використовуються вони для нанесення герметиків, адгезивів, фарб або інших рідин, колаборативні роботи також можуть безпечно працювати поряд із гуманоїдними в цілому ряді процесів дозування. Загальний вигляд колаборативних роботів серії CRX фірми FANUC (<https://crx.fanuc.eu/>), ABB, KUKA та Delta представлено на рисунку 1.4., а їх технічні характеристики наведені в таблиці порівняння 1.8.



a)



б)



в)



г)

a) Cobot Fanuc CRX-25iA [151]; б) Cobot ABB GoFa CRB 15000 [152];
в) Cobot KUKA LBR iisy 15 R930 [153]; г) Cobot Delta D-Bot DC20 [154]

Рисунок 1.4 - Загальний вигляд колаборативних роботів

Таблиця 1.8 - Порівняння технічних характеристик колаборативних роботів: Cobot Fanuc CRX-25iA; Cobot ABB GoFa CRB 15000; Cobot KUKA LBR iisy 15 R930; Cobot Delta D-Bot DC20[151-154]

Характеристика	Колаборативні роботи			
	Cobot Fanuc CRX-25iA	Cobot ABB GoFa CRB 15000	Cobot KUKA LBR iisy 15 R930	Cobot Delta D-Bot DC20
1	2	3	4	5
Вантажопідйомність	25 кг	5 кг	15 кг	20 кг
Радіус дії (робоча зона)	1 889 мм	950 мм	930 мм	1700 мм
Кількість осей	6	6	6	6
Повторюваність	±0,03 мм	±0,02 мм	±0,01 мм	±0,02 мм
Вага	64 кг	27 кг	35 кг	55 кг
Швидкість	До 1000 мм/с	До 2000 мм/с	До 1500 мм/с	До 2000 мм/с
Захист (ступінь захисту)	IP54	IP67	IP54	IP54

Продовження таблиці 1.8

1	2	3	4	5
Програмування	Інтуїтивне, через планшет	Інтуїтивне (Wizard Easy Programming)	KUKA Sunrise.OS	Інтуїтивне
Енергоспоживання	~ 300 Вт	~ 250 Вт	~ 350 Вт	~ 300 Вт
Сенсори безпеки	Датчики сили та моменту	Датчики сили та моменту	Інтегровані датчики	Інтегровані датчики
Особливості	Висока вантажо-підйомність	Легка вага, підвищена точність	Легке програмування, висока точність	Велика вантажо-підйомність
Ціна (приблизно)	Від \$35,000	Від \$25,000	Від \$40,000	Від \$30,000

Порівняння функціональних можливостей класичних роботів-маніпуляторів і колаборативних роботів (коботів) є важливим через різні сфери їхнього застосування та технологічні особливості. Класичні роботи-маніпулятори зазвичай використовуються в важких промислових умовах, де потрібно виконувати монотонні або високошвидкісні операції з великою точністю. Вони працюють в умовах ізоляції від людини через високий рівень ризику, але здатні виконувати завдання, що потребують великої вантажопідйомності. Водночас колаборативні роботи оптимізовані для безпечної взаємодії з людьми, що робить їх ідеальними для використання на виробничих лініях, де необхідна гнучкість і адаптивність до різних процесів. Коботи прості у налаштуванні та експлуатації, що дозволяє використовувати їх на малих і середніх підприємствах, а також у наукових дослідженнях. Тому порівняння цих типів роботів дозволяє краще зрозуміти їх сильні і слабкі сторони, що є ключовим для вибору оптимального рішення для конкретних виробничих завдань. Порівняння функціональних можливостей класичного робота-маніпулятора та колаборативних роботів (Cobot Fanuc CRX-25iA, KUKA LBR iisy 15 R930) наведено в таблиці 1.9.

З таблиці 1.9 порівняння функціональних можливостей класичних роботів-маніпуляторів та колаборативних роботів видно, що кожен тип має свої унікальні переваги та обмеження.

Таблиця 1.9 - Порівняння функціональних можливостей класичного робота-маніпулятора та колаборативних роботів

Функціональні можливості	Класичний робот-маніпулятор	Кобот (Cobot)
1	2	3
Призначення	Переважно для важких промислових завдань, повторюваних операцій у середовищі з високим ступенем автоматизації	Для співпраці з людьми, легка автоматизація завдань у різних середовищах
Безпека	Потребує огорожі, датчиків і захисних бар'єрів для запобігання нещасним випадкам	Інтегровані сенсори сили, моменту та безпеки, дозволяють працювати поряд з людьми без додаткових бар'єрів
Програмування	Складніші системи програмування, потребують знання мов програмування робота	Інтуїтивне програмування через планшети або НМІ, просте налаштування для непрофесійних користувачів
Повторюваність операцій	Висока точність та повторюваність при однотипних завданнях	Висока точність, але призначена для більш різноманітних та гнучких завдань
Вантажопідйомність	Вища (до 100 кг і більше, залежно від моделі)	Обмежена (до 25 кг у Fanuc CRX-25iA, до 15 кг у KUKA LBR iisy)
Гнучкість використання	Обмежена, призначена для вузьких завдань і програмованих операцій	Гнучка, легко інтегрується в різні виробничі процеси, можливість швидкого переналаштування
Швидкість роботи	Зазвичай швидший, для високошвидкісного виробництва	Менш швидкий, оптимізований для безпечної співпраці з людьми
Захист	Вимагає високого ступеня захисту для роботи в екстремальних умовах (IP67 та вище)	Менший захист (IP54), орієнтований на роботи у звичайних виробничих умовах
Сенсори та датчики	Обмежені або відсутні сенсори безпеки, можливе додаткове обладнання	Інтегровані сенсори для контролю сили і моменту, що забезпечують безпеку і адаптивність
Мобільність	Зазвичай стаціонарний, важкий монтаж	Компактний, легкий монтаж, можливість використання в різних місцях
Співпраця з людиною	Потребує повної ізоляції від людей через високий ризик нещасних випадків	Спеціально розроблений для безпечної співпраці з людиною (колаборації)
Вартість експлуатації	Висока вартість установки, обслуговування та безпеки	Низька вартість установки та експлуатації через простоту налаштування та безпеку
Застосування	Великі серійні виробництва, заводи, важка промисловість	Малі та середні підприємства, складальні лінії, наукові та дослідницькі роботи

Класичні роботи-маніпулятори більше підходять для великих виробництв, де важливі висока швидкість, точність і здатність працювати з важкими вантажами. Однак їх використання вимагає додаткових заходів безпеки, оскільки вони не можуть працювати поряд з людьми без ізоляції. Колаборативні роботи, з іншого боку, оптимізовані для безпечної взаємодії з людиною, що робить їх більш гнучкими та зручними для впровадження в середовищах, де потрібна швидка адаптація до змін у виробничих процесах. Вони мають меншу вантажопідйомність і швидкість, але забезпечують більшу гнучкість і легкість програмування, що робить їх ідеальними для малих і середніх підприємств.

1.4 Визначення та класифікація колаборативних роботів

Сучасні тренди розвитку робототехніки в цілому вказують на значний приріст випуску та застосування особливого виду промислових роботів (ПР) на сучасних технологічних виробництвах, а саме колаборативних промислових роботів (КПР) [155]. КПР – це вид ПР, розроблений для виконання різноманітних технологічних завдань у колаборації (співпраці) з працівниками (людьми) у виробничих секторах. КПР знайшли своє застосування не тільки у виробничих, а й у невиробничих сферах, наприклад, у медицині. Різноманітність сфер застосування КПР породжує очевидну, на думку авторів, проблематику термінології та змістовності загальних визначень, правил безпеки під час експлуатації та методів програмування, що забезпечують безпечну взаємодію людини та КПР, виконання безпечних (неколізійних) траєкторій тощо. Проведений аналіз досліджень наявних інформаційних джерел, наприклад, стандартів ISO, не формує єдиної термінології та пропонує різні технічні рішення, які є вузько націленими для конкретних технологічних процесів та/або завдань. Зараз термінологія

використання КПП та її змістовність не визначена і фактично вкрай рідко розглядається в Україні [155].

Застосування КПП на сучасних промислових виробництвах є однією із характерних ознак рівня Industry 4.0 та Industry 5.0. Застосування КПП на сучасному виробництві ставить ряд проблемних питань, пов'язаних зі змістовністю термінологічних визначень та їх коректного тлумачення. Нижче наведений стислий аналіз інформаційних джерел вказує на актуальність та релевантність змісту цього питання, а також на різноманітність підходів та методів досліджень у цьому напрямі.

Роботи [155, 156, 157] вказують на наявність проблематики застосування виробничих стандартів ISO на практиці у виробництві, у тому числі в сфері робототехніки. Через швидкий технологічний розвиток потрібні зрозумілі мости між технічною та юридичною сферами в колаборативних технологіях. Вказані роботи акцентують увагу на тому, що з появою відносно нового для виробництва виду промислових роботів (КПП) з'являється ряд питань щодо однозначності термінів у різних стандартах та як правильно юридично трактувати вказане вище. У [155,158] висвітлюється ідентифікація та дослідження категорій у стандарті ISO [159]. Зміст публікації спрямований на те, щоб виробники роботів та мехатронних пристроїв усвідомили важливість чітких визначень згідно зі стандартами для дотримання законодавства в країнах, де вони експлуатуються. Зазначені роботи частково розглядають проблему, яку висвітлюють автори цієї статті, проте пропонують рішення лише в загальному понятті та в юридичному аспекті. У [155,160,161] представлено новий підхід до контролю безпеки в колаборативній робототехніці для промислового застосування на основі вимог останніх стандартів безпеки. Вимоги безпеки перевіряються шляхом виконання тестів перетину між обмежувачами об'ємами, розміри яких обчислюються онлайн, на основі швидкості руху ланок маніпуляційної системи (МС) вздовж траєкторії до зупинки конструктивних елементів КПП. Цей підхід

спрямований на подолання обмежень, які погіршують продуктивність КІР через застосування надмірно консервативних обмежень безпеки. Траєкторії зупинки для МС обчислюються в режимі онлайн і виконуються під час встановлення потенційних зіткнень. Ці траєкторії оптимізовані для мінімізації часу зупинки із врахуванням динаміки КІР та обмежень максимального крутного моменту на з'єднаннях ланок МС [155].

Запропонований підхід дозволяє в режимі онлайн оцінити безпечні зони, яких вимагає технічна специфікація. Вказані роботи вирішують ряд проблемних питань колаборативної робототехніки, що передбачають дотримання вимог безпеки за стандартами ISO, але все ж в них повністю не дотримуються термінологічні визначення за відомими системами стандартів. У [155, 162] розглядаються вимоги щодо забезпечення безпечної роботи систем із використанням КІР, які згадуються в стандарті [163] для колаборативних роботів. Стандарт [155, 163] пропонує рекомендації, які розробники та користувачі мають враховувати при впровадженні засобів захисту в системи з колаборативними пристроями. Але в цьому нормативному документі не передбачено чіткої орієнтації для розробників чи користувачів у виборі запобіжних заходів з усього переліку. У цій публікації висвітлено описані вище прогалини та недоліки і запропоновано фреймворк (програмна платформа), заснований на відомих стандартах [155, 164], для орієнтування розробників та користувачів у переліку захисних заходів щодо застосування КІР на основі результатів аналізу небезпек та оцінки ризиків. Такі напрацювання сортують існуючі терміни та визначення за [155, 163], що є корисним на практиці та частково вирішує проблему однозначності термінів, проте не пропонує нічого нового для зменшення неоднозначності термінологічних та змістовних прогалин. Стаття [115, 165] вказує на зростання потреби ПР з високою точністю позиціонування відповідно до вимог Industry 4.0.

Стандарти [166] і [167] окреслюють вимоги до розрахунку точності ПР та визначають базовий елемент робота (robot base frame – RBF) як одну зі складових, яка впливає на кінцеву точність, що визначена геометричними характеристиками МС [155]. Так званий RBF фактично є точним положенням ланки всередині конструкції ПР та КПР. Однак специфічний підхід чи процес його визначення не передбачено жодним стандартом. У [155, 165] пропонується основа для універсального підходу щодо визначення RBF, який інтегрує метрологічне обладнання та проектування Design of Experiments (DOE). Підхід DOE досліджує, як різні фактори (використовувані осі робота, кількість використовуваних положень точок, їх положення тощо) впливають на повторюваність при встановленні RBF, вимірюючи положення точок, яких робот досягає. Встановлено, що запропонований підхід підвищує повторюваність встановлення початкової точки RBF на 93,4 % порівняно з попереднім методом на основі довільно обраних факторів [155].

У [155, 168] стверджується, що стандарт [170] сприяє впровадженню гібридних виробничих систем, тобто виробничих систем, які характеризуються тісним взаємозв'язком між людьми-операторами та роботами у спільних завданнях. Гібридні системи «людина-робот» можуть мати велику економічну вигоду в дрібносерійному та середньосерійному виробництві, навіть якщо ця нова парадигма запровадить обов'язкові комплексні аспекти безпеки. Існують різноманітні вимоги до робочих просторів для колаборативної роботи, проте гарантія безпеки містить два різні прикладні рівні: алгоритми, що дозволяють безпечно розподіляти простір між людьми та роботами, і технології, що дозволяють отримувати дані з синтезу датчиків і аналізувати дані про навколишнє середовище. У цій роботі розглядаються обидві проблеми: стратегія уникнення зіткнень, що дозволяє онлайн перепланувати траєкторію руху робота, і безпечна мережа пристроїв як запропонована інфраструктура для досягнення функціональної безпеки [155]. Також дослідники спираються на відомі стандарти в сфері

колаборативної робототехніки, проте не стверджують однозначності змісту існуючих термінів або визначень.

Зміст дослідження [171] акцентує увагу на важливості встановлення правильного призначення завдання під час колаборації людини та робота для поєднання продуктивності робота з гнучкістю людини. У зв'язку з цим у цій роботі пропонується алгоритм підходу до проблеми балансування складальної лінії Assembly Line Balancing Problem (ALBP) щодо розподілу часу виконання певних технологічних завдань між Н-R (Human-Robot collaboration) у випадку колаборативної роботи людини (Н) і робота (R) [155]. Вказана робота ставить за мету дослідити та досягти: 1) мінімізації вартості конвеєрної лінії, оціненої відповідно до кількості працівників та обладнання на лінії, враховуючи КПП; 2) мінімізації кількості кваліфікованих робітників на лінії; 3) мінімізації розподілення навантаження серед працівників на основі їх фізичних можливостей у контексті Н-R. Останнє є новим як за змістом, так і за його якісною та кількісною оцінкою. Це дослідження є прикладом роботи, яка успішно вирішує специфічне завдання колаборативної робототехніки, але не враховує однозначність термінів та різні стандарти безпеки [155].

У [155,172] описуються структура динамічного розподілу завдань на основі фізичних властивостей компонентів, характеристики завдань та колаборативного робочого простору. Розроблені розподіли завдань досягаються за допомогою набору атрибутів механоскладального процесу з різними балами за часом циклу, адаптивності та безпеки. Робота ілюструється на прикладі колаборативного механоскладального процесу за участі людини і робота. Дослідження впливу застосування колаборативних роботів у харчовій промисловості на прикладі розробленої для лінії виробництва системи громадського харчування є змістом статті [155, 173]. Запропоновано узагальнюючу методологію для підтримки дослідження техніко-економічної доцільності впровадження такої технології. Ця методологія призначена для менеджерів харчової промисловості, які аналізують автоматизацію процесу і

вимірюють продуктивність системи з точки зору пропускної здатності, ергономіки та економічних переваг, які є результатом впровадження колаборативних роботів. У [155, 174] пропонується алгоритм виявлення колізій без зовнішніх датчиків, які можуть виявляти потенційні зіткнення при взаємодії людини з роботом. Алгоритм заснований на модифікованому програмному спостерігачі відхилення імпульсу першого порядку, який також враховує тертя. Для перевірки алгоритму проведено моделювання та експеримент за допомогою експериментальної платформи колаборативного робота. Результати підтвердили, що зіткнення можна виявити, встановивши відповідні порогові значення сигналів зворотних зв'язків робота. Різні можливі реакції можуть бути імплементовані відповідно до різних стратегій реагування, причому остаточним обмежувачем є те, що сили зіткнення утримуються строго в звичайних межах людських фізичних можливостей (сила фізичного тиску тощо). Це гарантує збереження безпеки в процесах взаємодії системи «людина-робот», та торкається термінології у сфері колаборативної робототехніки.

Стаття [155, 175] описує поточні потреби фінської промисловості з використанням коботів, інший термін щодо колаборативних роботів. Окрім пошуку нових способів спільної роботи з КПР, проблемою залишається безпека. При цьому наголошується на існуванні загальної проблеми щодо недостатності знань про коботи. Ця стаття допомагає зосередити дослідницькі зусилля для підтримки застосування коботів у виробничій галузі без акценту на терміни колаборативної робототехніки.

У [155, 176] розглядається використання популярних ненормованих, тобто сленгових, термінів, пов'язаних із безпекою використання роботів, зокрема, «співіснування», «співпраця», «кооперація», «колаборативний робот», «кобот». Вказані вище терміни використовуються неоднозначно в інформаційних джерелах та на практиці. Порівняння термінів і докази невідповідності надані на основі останніх робіт про безпеку колаборативної

робототехніки. Рекомендації щодо класифікації методів безпеки обговорюються відповідно до сучасних стратегій безпеки змішаних режимів роботи технологічної системи Н-Р. У цій роботі розглядаються терміни та визначення і встановлюється коректність деяких термінів та їх використання. Узагальнення проведеного стислого аналізу наявних інформаційних джерел очевидним чином вказують на наявність нової технологічної системи, яка названа Н-Р, що певним чином функціонує та виконує колаборативні технології. Зміст проаналізованих інформаційних джерел можна умовно розділити на такі, що, з однієї сторони, висвітлюють проблеми безпеки [155,156-160], відповідності міжнародним стандартам [159-161,165] та функціональності [171–174], а з іншої – функціонування колаборативних технологічних систем (КТС), що характеризуються кількісними оцінками колаборації Н-Р, що є здебільшого характерне для складальних виробництв.

Загальна тенденція розширення сфери застосування КПП на основі виконаного аналізу інформаційних джерел показала, по-перше, що тематика колаборативної робототехніки активно обговорюється та досліджується як в наукових колах, так і між інженерами-практиками різних галузей та виробництв; по-друге, що дослідники та інженери успішно розв'язують вузьконаправлені задачі на основі можливості своїх засобів та обладнання, але часто нехтують питанням термінології, змістовності визначень та недотримання тих або інших міжнародних стандартів; по-третє, публікації щодо колаборативної робототехніки в Україні майже відсутні, як і відсутні відповідні ДСТУ з цієї проблематики. На базі вище проведеного аналізу публікацій побудуємо таблицю 1.8, в якій запропонована систематизація доступних інформаційних джерел щодо існуючої термінології використання промислових роботів в рамках концепція Industry 5.0.

Таблиця 1.8 - Систематизація доступних інформаційних джерел щодо існуючої термінології використання промислових роботів в рамках концепції Industry 5.0 [155].

Термін	Джерело				
	ISO 10218-1:2011 [172]	ISO 10218-2:2011 [170]	ISO/TS 15066:2016 [165]	ISO 13482:2014 [164]	ISO 31000 [168]
Колаборативна робототехніка	++	++	++	++	++
Колаборативний робот (кобот)	++	++	++	++	++
Колаборація	++	++	++	+	+
Колаборативна операція	++	++	++	+	+
Колаборативна технологічна система (КТС)	+	+	+	+	+
Колабораційний робочий простір	++	++	++	-	-
Взаємодія людина – робот	++	++	-	-	-
Рівень колаборації	++	++	++	-	-
Співіснування (Coexistence)	++	++	++	-	-
Послідовна співпраця (Sequential Collaboration)	++	++	++	-	-
Співпраця (Cooperation)	++	++	++	-	-
Колаборація зі зворотним зв'язком (Responsive Collaboration)	++	++	++	-	-

Умовні позначення: ++ – означає чітке визначення, або згадування; + – неявне визначення або опосередковане згадування, - – термін у відповідних інформаційних джерелах не згадувався і тим паче не розкривався його зміст

На підставі викладеного вище в проведеного аналізу інформаційних джерелах [155, 156–176] та нормативних документів в рамках даних досліджень рекомендують використовувати наступну термінологію в області колаборативної робототехніки:

Колаборативна робототехніка – це частина робототехніки, яка вивчає, досліджує та практикує застосування КПР у взаємодії (колаборації) з людиною під час виконання різноманітних технологічних операцій та виконання завдань.

Колаборативні роботи (коботи) – це роботи, призначені для прямої взаємодії Н-Р (людина – робот) в спільному робочому просторі або там, де люди та роботи знаходяться в безпосередньому контакті. Програми колаборативних роботів відрізняються від традиційних програм ПР, у яких роботи ізольовані від контакту з людьми [155, 175]. Як визначено Міжнародною організацією стандартизації [155, 162], кобот – це робот, який можна використовувати у колаборативних операціях, де роботи та люди працюють одночасно в межах визначеного робочого простору для виробничих операцій (це не враховує системи «робот і робот» або розміщених людей і роботів, які працюють у різний час).

Колаборація (в контексті промислової робототехніки) – це процес спільного виконання операції, дії чи роботи людиною та роботом для досягнення поставленої мети.

Колаборативні операції – це визначена послідовність дій між КПР та людиною, яка в результаті призводить до виконання конкретного завдання чи роботи.

Колаборативна технологічна система (КТС) – це технологічна система, в якій як універсальні гнучкі засоби автоматизації використовується КПР, з яким спільно працюють люди.

Колабораційний робочий простір – це спільний робочий простір людини та КПР, в якому виконуються колаборативні операції.

Взаємодія людина – робот (Human-Robot interaction HRI) – це процес взаємодії людей та роботів у різних аспектах життєдіяльності. Взаємодія людини з роботом – це багатодисциплінарна галузь, до якої входять взаємодії людини з комп’ютером, штучним інтелектом, робототехнікою.

Рівень колаборації – це умовне якісне визначення глибини співпраці людини та робота під час колаборативних операцій. IFR визначає чотири рівні колаборації між промисловими роботами та людьми [155, 178], а саме:

- *співіснування (Coexistence)* – це рівень колаборації, коли людина та робот працюють поруч один з одним без захисної огорожі, але без спільного робочого простору;

- *послідовна співпраця (Sequential Collaboration)* – це рівень колаборації, коли людина та робот активні в спільному робочому просторі, але їх рухи є послідовними; вони не працюють над однією операцією одночасно;

- *співпраця (Cooperation)* – це рівень колаборації, коли робот і людина працюють над однією частиною одночасно і обидва знаходяться в русі;

- *колаборація зі зворотним зв’язком (Responsive Collaboration)* – це рівень колаборації, коли робот реагує в режимі реального часу на рух працівника.

Варто наголосити, що вказані вище терміни та їх зміст можуть бути прийнятими до використання до періоду формування усталеної, стійкої термінології в сфері колаборативної роботехніки.

У рамках Industry 5.0 колаборативні роботи (Cobot) класифікуються за кількома ключовими параметрами, які відповідають принципам нової індустріальної епохи, що зосереджена на співпраці між людьми та роботами, сталому розвитку, персоналізації та інтелектуальних технологіях. На рисунку 1.5 наведена класифікація колаборативних роботів в контексті Industry 5.0

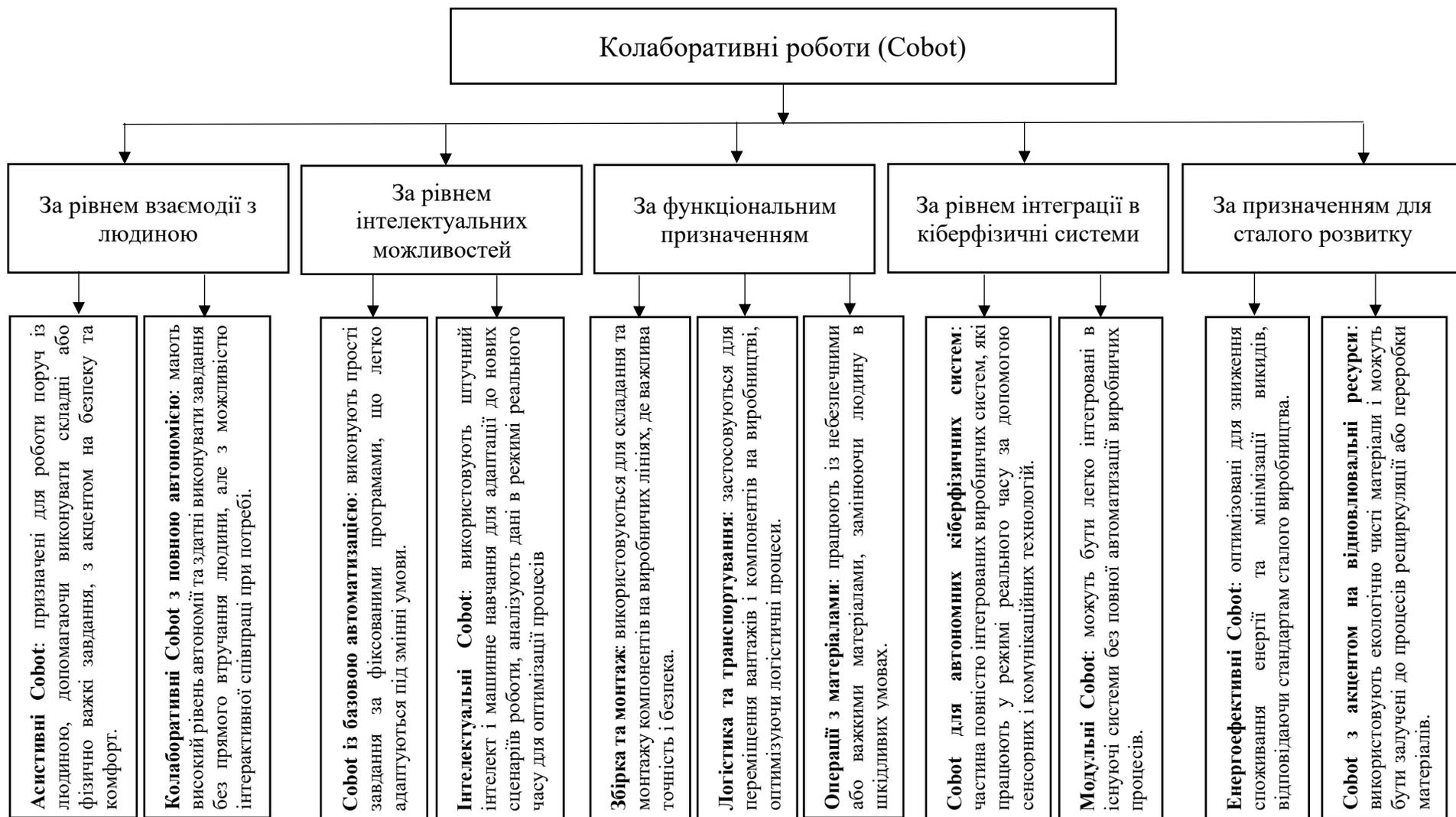


Рисунок 1.5 - Класифікація колаборативних роботів (Cobot) в контексті Industry 5.0

Industry 5.0 передбачає тісну інтеграцію Cobot з людськими працівниками для підвищення ефективності, персоналізації продукції та сталого розвитку, підкреслюючи важливість безпеки, інтелектуальних технологій і гнучкості в різних виробничих процесах.

1.5 Особливості взаємодії людини та колаборативного робота

У сучасній промисловості взаємодія людини і колаборативного робота відіграє ключову роль у підвищенні ефективності та безпеки виробничих процесів [177,178]. На відміну від класичних роботів-маніпуляторів, які працюють в ізольованих умовах, коботи розроблені спеціально для безпечної роботи поруч із людьми, що відкриває нові можливості для оптимізації виробництва та впровадження гнучких підходів до виконання завдань. Класичні роботи-маніпулятори, хоча й мають високу продуктивність та вантажопідйомність, зазвичай працюють в обмежених умовах, де взаємодія з людиною є небезпечною і вимагає жорстких заходів безпеки, таких як захисні огорожі чи спеціальні зони [179].

Основною особливістю колаборативних роботів є їхня здатність взаємодіяти з людьми без ризику для здоров'я та життя оператора. Коботи оснащені численними сенсорами та інтелектуальними системами безпеки, які дозволяють їм відстежувати навколишнє середовище і реагувати на присутність людини, миттєво зупиняючи рухи або змінюючи траєкторію дій [180]. Ця особливість робить їх придатними для використання на малих і середніх підприємствах, де відсутня можливість повної автоматизації процесів або де потрібна гнучка співпраця між людьми та машинами. Колаборативні роботи можуть виконувати допоміжні функції, звільняючи працівників від виконання монотонних або фізично важких завдань, таких як транспортування, складання, або обробка матеріалів.

Класичні роботи-маніпулятори, в свою чергу, зазвичай використовуються на великих виробництвах, де головною метою є виконання масових операцій із високою швидкістю та точністю. Однак такі роботи не можуть адаптуватися до змін у виробничих процесах так само швидко, як це роблять коботи. Класичні роботи потребують детального програмування та спеціальних умов для експлуатації, що ускладнює їх інтеграцію в середовища, де важлива взаємодія з людьми або постійна зміна виробничих завдань.

Ще одним важливим аспектом взаємодії з коботами є їхня інтуїтивність та легкість програмування [181]. Оператори можуть навчати колаборативних роботів нових завдань за допомогою ручного управління або простих інтерфейсів, таких як планшети або смартфони. Це значно знижує поріг входу для впровадження таких систем, дозволяючи компаніям швидше адаптуватися до ринкових змін та індивідуалізувати свої виробничі процеси. У той час як класичні роботи вимагають спеціальних інженерних навичок для програмування та налаштування, коботи можуть бути налаштовані на нові завдання без глибоких технічних знань [182].

Крім того, взаємодія людини та колаборативного робота забезпечує значне зниження фізичного навантаження на працівників. Коботи можуть виконувати важкі або рутинні операції, залишаючи працівникам можливість зосередитися на творчих або інтелектуально складних завданнях. Це призводить до підвищення продуктивності, зниження рівня втоми та зростання задоволення працівників своєю роботою [183].

Таким чином, особливості взаємодії між людиною та коботом є ключовими для побудови сучасних гнучких виробничих систем, які відповідають вимогам Industry 5.0. Порівняно з класичними роботами-маніпуляторами, колаборативні роботи забезпечують вищий рівень гнучкості, безпеки та адаптивності, що робить їх ідеальними для використання в динамічних та мінливих виробничих середовищах [184-187]. Основні аспекти взаємодії людини та колаборативного робота (кобота) відображають ключові принципи Industry 5.0 наведені на рисунку 1.6.

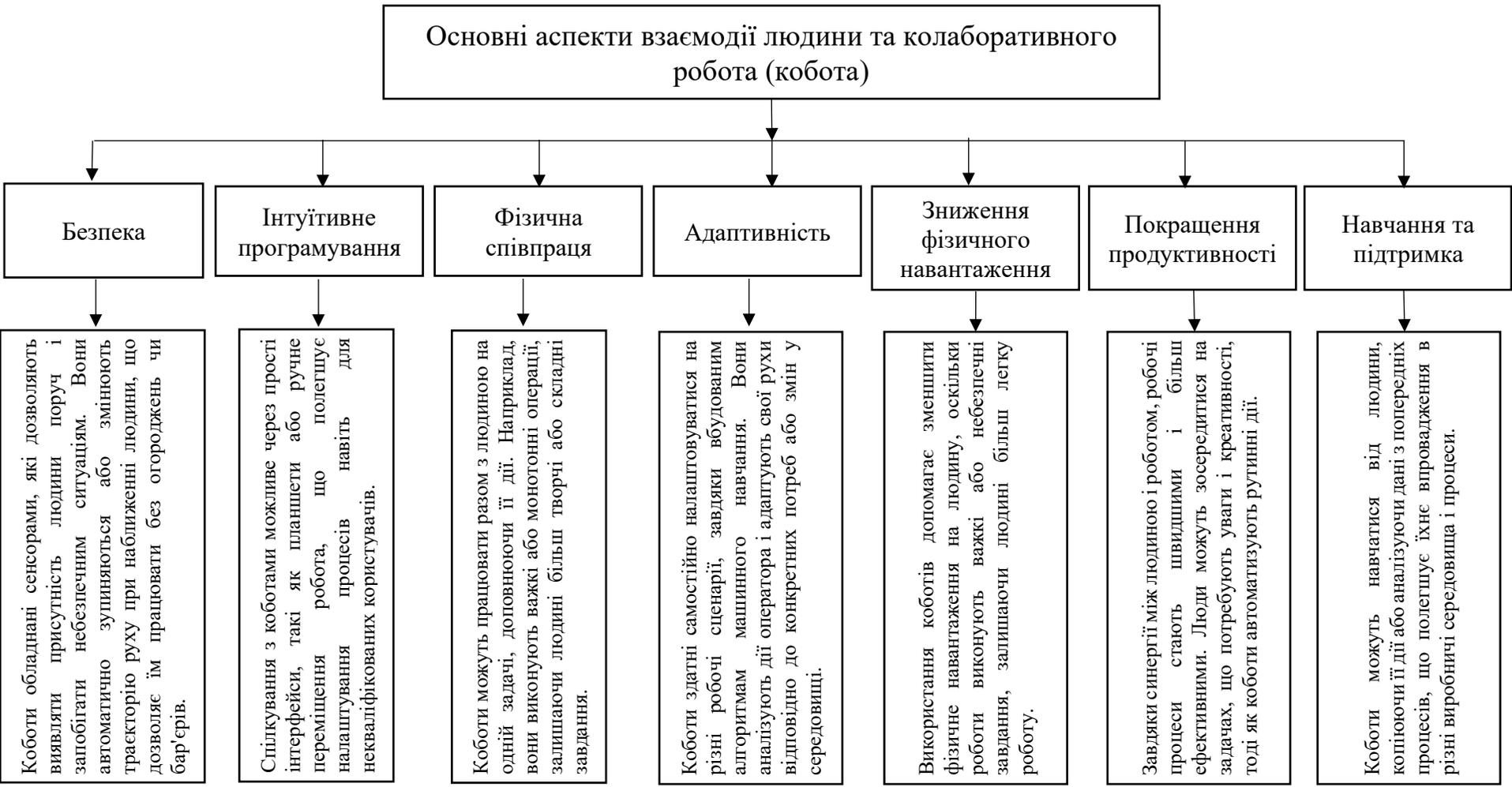


Рисунок 1.6 - Основні аспекти взаємодії людини та колаборативного робота (кобота)

Також потрібно розглянути сучасні методи взаємодії людини та колаборативного робота (кобота), які охоплюють різні способи спільної роботи, адаптації і спілкування, що забезпечують безпеку та ефективність виробничих процесів. Ці методи засновані на інноваційних технологіях сенсорики, програмування та інтерфейсів, що дозволяють людині і роботу працювати разом, гармонійно доповнюючи один одного [188,189]. Наведемо стислий опис кожного метода:

- *безконтактна взаємодія через сенсори*, коботи обладнані різними датчиками, наприклад, оптичними, ультразвуковими або тактильними, що дозволяють їм виявляти присутність людини поруч. Завдяки цьому робот може автоматично зупинитись або уповільнювати рухи, коли людина входить в робочу зону, що забезпечує високий рівень безпеки без необхідності встановлювати фізичні бар'єри;

- *пряме програмування за допомогою переміщення*. Цей метод передбачає, що оператор може програмувати рухи кобота вручну. Людина фізично переміщує маніпулятор в потрібні позиції, а робот запам'ятовує ці рухи для подальшого виконання завдань. Це значно полегшує процес навчання робота новим операціям та скорочує час на налаштування.

- *інтерактивні інтерфейси*. Багато колаборативних роботів підтримують інтуїтивні графічні інтерфейси для програмування і керування. Оператори можуть використовувати планшети, комп'ютери або смартфони для налаштування робота, вибору завдань та відстеження його стану. Такі інтерфейси дозволяють керувати роботом без потреби в глибоких технічних знаннях.

- *спільне виконання завдань*. Коботи можуть виконувати частину завдання спільно з людиною. Наприклад, людина виконує складні операції, а робот виконує допоміжні або повторювані дії, такі як підйом або переміщення важких об'єктів. Це зменшує фізичне навантаження на працівників та підвищує загальну продуктивність.

- *голосове управління*. Деякі колаборативні роботи підтримують голосові команди для керування. Оператор може давати інструкції роботу в реальному часі, що дозволяє швидко змінювати або коригувати завдання без необхідності ручного налаштування.

- *зворотний зв'язок через візуалізацію*. Для полегшення взаємодії з коботом можуть використовуватися візуальні сигнали або індикатори, такі як світлові діоди чи дисплеї. Це дозволяє оператору отримувати інформацію про стан робота, його готовність до виконання завдань або можливі помилки.

- *тактильна взаємодія*. В окремих випадках коботи використовують тактильний зворотний зв'язок для взаємодії з людиною. Наприклад, оператор може відчувати легкий опір або вібрацію при неправильному положенні руки робота або при досягненні критичної точки, що допомагає уникати помилок під час спільної роботи.

Ці методи забезпечують максимально гнучке та безпечне співробітництво людини і робота, що робить їх застосування ідеальним у сучасних виробничих середовищах, де потрібні адаптивність, швидка реакція на зміни і висока ефективність.

1.6 Завдання колаборативних роботів в промислових умовах

Колаборативні роботи (коботи) виконують різноманітні завдання в промислових умовах, орієнтуючись на підвищення ефективності, безпеки та гнучкості виробничих процесів. Їх основна мета – взаємодіяти з людиною та доповнювати людські можливості в таких умовах, де класичні роботи не можуть працювати через необхідність гнучкої адаптації або тісної співпраці з операторами [190-195]. Основні завдання коботів у промислових умовах:

- *автоматизація рутинних операцій*. Коботи широко використовуються для виконання монотонних, рутинних операцій, таких як пакування, складання, переміщення об'єктів або їх сортування. Це дозволяє людям

зосередитися на більш складних і творчих завданнях, підвищуючи продуктивність та знижуючи втому від повторюваної роботи;

- *підтримка складальних процесів.* У виробництві роботи допомагають у складальних операціях, які вимагають високої точності та концентрації. Вони можуть разом із людиною збирати деталі, забезпечуючи послідовність, надійність і швидкість, що особливо важливо на лініях з великою кількістю змін;

- *обробка матеріалів.* Роботи використовуються для переміщення та обробки матеріалів, зокрема важких або великих предметів. Їхня здатність працювати поруч із людьми без захисних бар'єрів дозволяє оптимізувати процеси транспортування, зменшуючи ризик травм для працівників;

- *контроль якості.* У виробництві колаборативні роботи також залучаються до перевірки та тестування продукції. Вони можуть використовувати вбудовані камери та сенсори для виявлення дефектів, нерівностей або інших недоліків у продукції, забезпечуючи стабільність контролю якості на всіх етапах;

- *підтримка персоналізованого виробництва.* Роботи здатні швидко адаптуватися до нових завдань і зміни продукції, що дозволяє їх ефективно використовувати на виробничих лініях для малих партій або персоналізованої продукції, де часто змінюються налаштування або деталі;

- *обслуговування верстатів.* Колаборативні роботи можуть допомагати в обслуговуванні верстатів, виконуючи операції з подачі матеріалів, завантаження та вивантаження деталей, що підвищує ефективність роботи обладнання та зменшує простої;

- *монтаж та налагодження компонентів.* Роботи можуть допомагати в складальних операціях, пов'язаних з монтажем та налагодженням компонентів, таких як електронні частини, вузли механізмів, а також робота з дрібними деталями, де потрібна висока точність та контроль.

Завдяки своїй гнучкості, безпеці та можливості працювати поруч з людьми, колаборативні роботи стають важливими учасниками сучасних промислових процесів, де ключову роль відіграють адаптивність, швидка реакція на зміни та співпраця між людиною і роботом.

2. ДИНАМІЧНИЙ ОПИС НАВКОЛИШНЬОГО СЕРЕДОВИЩА КОЛАБОРАТИВНОГО РОБОТА

Динамічний опис навколишнього середовища є критичним елементом у роботі колаборативних роботів (коботів), що дозволяє їм взаємодіяти з оточенням у режимі реального часу та забезпечувати безпечну співпрацю з людиною [196-198]. На відміну від традиційних роботів, які працюють у строго контрольованих і передбачуваних умовах, колаборативні роботи функціонують у змінних середовищах, що потребує постійного оновлення даних про розташування об'єктів та людей [199]. Основою для цього є використання сенсорних систем, алгоритмів машинного навчання та обробки зображень, що дозволяють створювати та оновлювати модель навколишнього середовища [200]. Ці підходи забезпечують високу гнучкість та безпеку роботи колаборативних роботів, роблячи їх незамінними в сучасних виробничих та промислових умовах, де швидкість і точність взаємодії з мінливим середовищем мають вирішальне значення.

2.1 Методи динамічного опису середовища

Методи динамічного опису середовища колаборативного робота забезпечують здатність робота адаптуватися до змінних умов навколо нього, ефективно реагувати на них та забезпечувати безпечну взаємодію з людиною. Враховуючи специфіку співпраці з людьми, такі роботи мають використовувати складні алгоритми для аналізу оточення в реальному часі та швидкого прийняття рішень. Основні методи динамічного опису навколишнього середовища колаборативного робота в рамках концепції Industry 5.0 представлені на рисунку 2.1 [201-204].

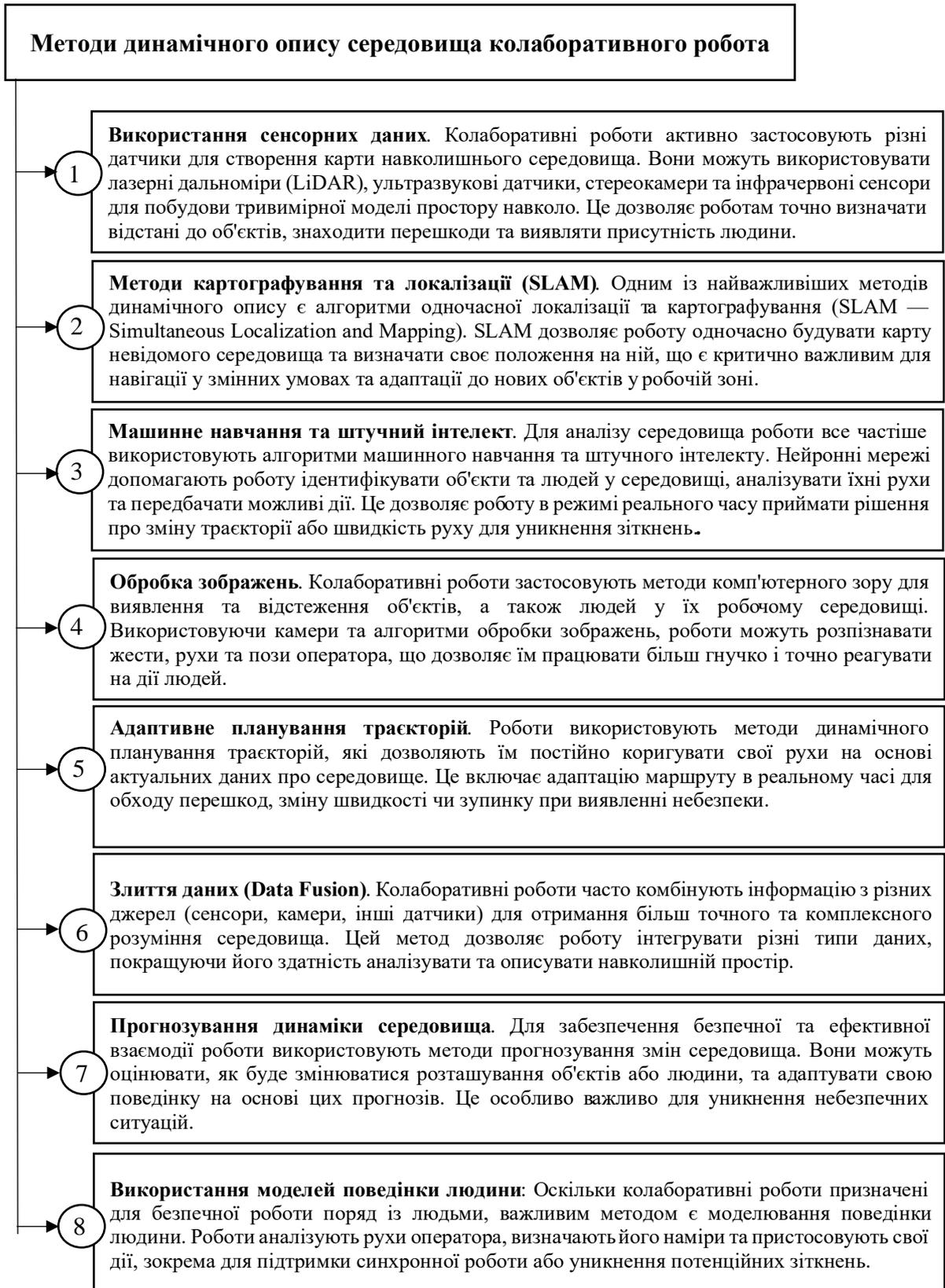


Рисунок 2.1 - Основні методи динамічного опису навколишнього середовища колаборативного робота в рамках концепції Industry 5.0

Колаборативні роботи активно застосовують різні методи для динамічного опису середовища, кожен із яких має свої переваги та недоліки. Проведемо аналіз кожного метода та виявимо їх переваги та недоліки з точки зору використання для опису динамічного навколишнього середовища кобота [205-210].

Використання сенсорних даних є одним із ключових методів, що дозволяє роботам орієнтуватися у просторі та знаходити перешкоди. Лазерні дальноміри (LiDAR), ультразвукові датчики, стереокамери та інфрачервоні сенсори створюють тривимірну модель простору навколо робота. Перевагою такого підходу є висока точність у визначенні відстаней до об'єктів і швидке виявлення перешкод, однак вартість таких сенсорів може бути високою, а їх робота в умовах пилу або відблисків іноді дає помилкові дані.

Методи одночасної локалізації та картографування (SLAM) є надзвичайно ефективними для побудови карти нового середовища, дозволяючи роботу одночасно картографувати простір і визначати власне положення. Це особливо корисно для навігації в невідомих або змінних умовах. Проте SLAM вимагає великих обчислювальних ресурсів, і в середовищах з багатьма рухомими об'єктами алгоритми можуть втрачати точність, що може уповільнювати процес адаптації робота.

Алгоритми машинного навчання та штучного інтелекту забезпечують роботам здатність ідентифікувати об'єкти та людей у середовищі, передбачати їхні дії та приймати рішення щодо уникнення зіткнень. Такі алгоритми дають можливість роботам працювати у складних умовах, але потребують значних ресурсів для навчання та налаштування моделей, що може бути трудомістким процесом. Окрім того, якість розпізнавання значною мірою залежить від якості навчальних даних.

Методи обробки зображень дозволяють колаборативним роботам точно виявляти та відстежувати об'єкти та людей, використовуючи камери. Це

робить роботи більш гнучкими у взаємодії з операторами, оскільки вони можуть розпізнавати жести, рухи та пози людини. Проте обробка зображень також має свої обмеження: в умовах недостатнього освітлення або перешкод на шляху камер точність розпізнавання може знижуватися.

Адаптивне планування траєкторії забезпечує постійне коригування рухів робота на основі актуальних даних про середовище. Це підвищує ефективність роботи у змінних умовах і дозволяє швидко реагувати на появу перешкод. Однак цей метод може потребувати складних розрахунків, що уповільнює процес прийняття рішень у реальному часі. Злиття даних з різних сенсорів покращує точність аналізу, оскільки інтеграція різних типів інформації дозволяє уникати помилкових висновків. Проте це також збільшує складність системи та її вимоги до обчислювальних ресурсів.

Прогнозування динаміки середовища дозволяє роботам оцінювати, як змінюватиметься розташування об'єктів або людей, і заздалегідь коригувати свою поведінку. Це критично важливо для забезпечення безпеки, але точність прогнозів залежить від якості вхідних даних та алгоритмів. Використання моделей поведінки людини є ще одним важливим методом для колаборативних роботів, які працюють у безпосередній близькості з операторами. Аналізуючи рухи людини, робот може передбачити її дії та уникнути можливих зіткнень. Водночас, неправильне трактування намірів оператора може призвести до небажаних результатів, особливо в критичних ситуаціях.

На базі проведеного аналізу порівняння переваг та недоліків вище перерахованих методів, побудуємо таблицю 2.1, в якій узагальнимо отримані результати дослідження [211-216].

Ця таблиця підсумовує основні сильні та слабкі сторони кожного з методів, що використовуються для динамічного опису навколишнього середовища колаборативних роботів.

Таблиця 2.1 - Переваги та недоліки методів для опису динамічного середовища колаборативних роботів

Метод	Переваги	Недоліки
Використання сенсорних даних	Висока точність визначення відстаней до об'єктів; можливість виявлення перешкод та людей; створення тривимірної моделі простору.	Висока вартість сенсорів; помилки у складних умовах (пил, відблиски, шум); залежність від точності окремих сенсорів.
SLAM (локалізація та картографування)	Однчасне побудування карти та визначення позиції робота; критично важливе для навігації в нових чи змінних умовах.	Великі обчислювальні ресурси; втрата точності в середовищах з багатьма рухомими об'єктами.
Машинне навчання та ШІ	Ідентифікація об'єктів і людей; передбачення дій; можливість прийняття рішень у реальному часі для уникнення зіткнень.	Високі вимоги до обчислювальних ресурсів; потреба в якісних навчальних даних; трудомісткість налаштування моделей.
Обробка зображень	Виявлення та відстеження об'єктів і людей; розпізнавання жестів і рухів; гнучкість у взаємодії з операторами.	Низька точність у складних умовах освітлення; потреба у високій якості відеопотоку; можливі затримки в обробці даних.
Адаптивне планування траєкторії	Постійне коригування рухів; швидке реагування на зміни в середовищі; можливість обходу перешкод у реальному часі	Великі вимоги до швидкості обчислень; можливі затримки у прийнятті рішень у складних середовищах.
Злиття даних (Data Fusion)	Підвищена точність через інтеграцію даних з різних сенсорів; покращене розуміння середовища; зниження ризику помилок	Складність реалізації; високі вимоги до обчислювальних ресурсів для одночасної обробки даних з багатьох джерел.
Прогнозування динаміки середовища	Можливість передбачення змін у розташуванні об'єктів та людей; адаптація дій робота для уникнення небезпек.	Залежність від якості вхідних даних; неточність прогнозів може призвести до небажаних ситуацій.
Моделювання поведінки людини	Аналіз намірів і рухів оператора; підтримка синхронної роботи з людиною; уникнення потенційних зіткнень	Можливі помилки в інтерпретації дій людини; висока складність налаштування моделей для точного розпізнавання намірів оператора.

2.2 Використання комп'ютерного зору для спостереження за середовищем

Комп'ютерний зір відіграє вирішальну роль у взаємодії колаборативних роботів з людьми та навколишнім середовищем, забезпечуючи роботам здатність сприймати, аналізувати та адаптуватися до змінних умов роботи в реальному часі. Це дозволяє роботам виконувати завдання, що вимагають високого рівня точності, безпеки та ефективної взаємодії з людиною, що є центральним елементом Industry 5.0. [217-221]

Один із ключових аспектів комп'ютерного зору - це здатність колаборативного робота візуально розпізнавати та відстежувати об'єкти у своєму робочому середовищі. Завдяки камерам та іншим сенсорам, роботи можуть створювати тривимірні моделі простору, ідентифікувати перешкоди та динамічні об'єкти, такі як люди, інструменти або техніка. Алгоритми обробки зображень, включаючи сегментацію та розпізнавання об'єктів, дають можливість роботам виділяти ключові елементи на зображеннях та розуміти їх функціональне значення. Це дозволяє адаптувати дії робота в реальному часі, наприклад, змінювати траєкторію руху для уникнення зіткнень або реагувати на дії оператора [222,223].

Важливим аспектом комп'ютерного зору є можливість розпізнавання людини та її дій. Колаборативні роботи здатні не тільки виявляти присутність людей, а й аналізувати їх жести, рухи та пози. Це критично важливо для забезпечення безпечної взаємодії, особливо у виробничих умовах, де людина і робот можуть працювати пліч-о-пліч [224]. Використовуючи технології машинного навчання та штучного інтелекту, такі як згорткові нейронні мережі (CNN), роботи можуть аналізувати поведінку оператора, передбачати його наміри і відповідно коригувати свою поведінку, що знижує ризик виникнення небезпечних ситуацій [225-227].

Комп'ютерний зір також дозволяє роботам виконувати складні операції в умовах змінного середовища. Роботи здатні збирати інформацію з різних джерел, таких як камери, LiDAR або інші сенсори, і об'єднувати ці дані для більш повного розуміння простору. Цей процес, відомий як злиття даних (Data Fusion), дозволяє роботам не лише бачити об'єкти, але й аналізувати їх розташування, рух, передбачати можливі зміни та адаптувати свої дії відповідно до нових умов. Наприклад, при виявленні нової перешкоди робот може швидко змінити свою траєкторію або зупинити рух для запобігання аваріям [228-230].

Завдяки методам комп'ютерного зору, колаборативні роботи можуть працювати більш автономно, надаючи можливість взаємодії з людьми без постійного контролю з боку оператора. Це підвищує гнучкість та ефективність роботи, оскільки роботи можуть брати на себе рутинні та небезпечні завдання, залишаючи людині можливість виконувати більш творчі та управлінські функції.

У цілому, комп'ютерний зір є невід'ємною складовою колаборативних роботів, надаючи їм можливість інтегруватися у складне виробниче середовище, працювати разом з людьми, забезпечувати безпеку та ефективність виконання завдань. Це сприяє розвитку Industry 5.0, яка орієнтована на тісну співпрацю між людиною і машиною для досягнення спільних цілей.

Фундаментальні поняття, такі як обробка зображень, сегментація та розпізнавання об'єктів, відіграють ключову роль у розвитку колаборативних роботів в рамках концепції Industry 5.0. Ця ера зосереджена на тісній взаємодії роботів з людьми, що вимагає нових підходів до безпеки, гнучкості та адаптивності. Для забезпечення такої співпраці, колаборативні роботи повинні ефективно сприймати навколишнє середовище, розуміти його структуру, визначати об'єкти та їх положення, і це стає можливим завдяки технологіям комп'ютерного зору, що засновані на обробці зображень [231-235].

Обробка зображень є основним етапом, що передбачає цифрову обробку візуальної інформації, яку отримують колаборативні роботи через камери або інші сенсори. Цей процес включає покращення якості зображень, видалення шумів, зміну контрасту та роздільної здатності, а також інші операції, що дозволяють підготувати зображення для подальшого аналізу. Обробка зображень у контексті Industry 5.0 є особливо важливою, оскільки вона дозволяє роботам швидко адаптуватися до змінних умов, що виникають у роботі з людьми в динамічному середовищі [236-240]. Такі алгоритми, як фільтри Гауса, або техніки морфологічної обробки, наприклад, ерозія або дилатація, використовуються для виділення важливих деталей зображення.

Наступним етапом є *сегментація*, яка полягає у поділі зображення на кілька сегментів або кластерів пікселів для кращого розуміння структури зображення. Для колаборативних роботів сегментація має вирішальне значення, оскільки дозволяє виокремити ключові об'єкти в робочій зоні, такі як людина, інструменти, машини або перешкоди. Сегментація може здійснюватися за допомогою різних методів, включаючи сегментацію на основі порогу, кластеризацію методом k-середніх або використання нейронних мереж. В рамках концепції Industry 5.0 це дозволяє роботам працювати в реальному часі, швидко реагувати на присутність людей або зміну положення об'єктів, що забезпечує безпечне та ефективне співробітництво [241,242].

Розпізнавання об'єктів - це процес ідентифікації та класифікації об'єктів на зображеннях, отриманих від сенсорів. Для колаборативних роботів розпізнавання об'єктів є критично важливим, оскільки вони повинні чітко розуміти, що знаходиться в їхньому полі зору, і на основі цього приймати рішення [243-245]. Алгоритми розпізнавання можуть використовувати шаблони або бути засновані на глибокому навчанні, наприклад, використанні згорткових нейронних мереж (CNN). У контексті Industry 5.0, ці алгоритми дозволяють роботам не лише бачити об'єкти, а й прогнозувати їх поведінку,

аналізувати рухи людини та передбачати подальші дії. Це стає основою для адаптивного планування траєкторій та уникнення потенційних загроз під час спільної роботи з людиною.

Отже, *обробка зображень, сегментація та розпізнавання об'єктів* є невід'ємними компонентами колаборативних роботів в Industry 5.0. Вони дозволяють роботам більш точно сприймати навколишнє середовище, інтерактивно взаємодіяти з людьми та адаптувати свою поведінку до змін у робочій зоні [246].

Загальна математична модель використання комп'ютерного зору для спостереження за навколишнім середовищем колаборативного робота-маніпулятора може бути представлена як набір рівнянь та виразів, що описують процес обробки зображень, розпізнавання об'єктів і прийняття рішень. Вона включає декілька основних етапів: отримання вхідних зображень, сегментацію, локалізацію об'єктів, розпізнавання та планування руху.

Процес починається зі збору зображень з камери. Вхідне зображення - це матриця інтенсивностей світла в пікселях, та може бути описана наступним чином:

$$I(x, y) = f(x, y, t) \quad (2.1)$$

Де: $I(x, y)$ - інтенсивність пікселя на координатах (x, y) ;

$f(x, y, t)$ - функція, що описує зміни яскравості в залежності від просторових координат x і y , а також часу t .

Камера перетворює тривимірний світ у двовимірне зображення, тому потрібне калібрування для визначення внутрішніх і зовнішніх параметрів камери. Модель перспективної проєкції представлена у наступному виразі:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.2)$$

Де: (X, Y, Z) - координати точки в просторі;

(u, v) - координати точки на зображенні;

f_x, f_y - фокусні відстані;

c_x, c_y - координати оптичного центру камери.

Сегментація – це процес виділення об'єктів на зображенні. Операцію сегментації можна представити як функцію кластеризації пікселів:

$$S(x, y) = \begin{cases} 1, \text{ якщо піксель належить об'єкту} \\ 0, \text{ інакше} \end{cases} \quad (2.3)$$

Де: $S(x, y)$ - бінарна функція сегментації, яка визначає належність пікселя об'єкту.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} Z \quad (2.4)$$

Де: K - матриця внутрішніх параметрів камери;

(X, Y, Z) - тривимірні координати об'єкта;

(u, v) - координати об'єкта на зображенні.

Алгоритми машинного навчання, такі як згорткові нейронні мережі (CNN), використовуються для розпізнавання об'єктів. Формально процес розпізнавання можна описати як:

$$P(c|I) = f_{\theta}(I) \quad (2.5)$$

Де: $P(c|I)$ – ймовірність того, що об'єкт належить класу c на зображенні I ;

$f_{\theta}(I)$ - модель нейронної мережі з параметрами θ , яка прогнозує клас об'єкта.

Для безпечного та ефективного керування маніпулятором на основі отриманих даних про розташування об'єктів використовують алгоритми планування траєкторій. Це можна представити як мінімізацію функції вартості для руху:

$$J(\tau) = \int_0^T L(x(t), u(t)) dt \quad (2.6)$$

Де: $J(\tau)$ - функція вартості траєкторії τ ;

$L(x(t), u(t))$ - миттєва вартість руху в залежності від стану $x(t)$ та управління $u(t)$;

T - час виконання завдання.

Це мінімізується під обмеженням динаміки робота:

$$\dot{x}(t) = f(x(t), u(t)) \quad (2.7)$$

Де: $\dot{x}(t)$ - зміна стану системи з часом;

$f(x(t), u(t))$ - модель динаміки робота.

Для точнішого розуміння простору використовуються кілька джерел даних (камери, сенсори, LiDAR тощо). Інтеграція даних здійснюється шляхом злиття (Data Fusion):

$$X_f = w_1 X_1 + w_2 X_2 + \dots + w_n X_n \quad (2.8)$$

Де: X_f - результат злиття даних;

X_1, X_2, \dots, X_n - дані з різних джерел;

w_1, w_2, \dots, w_n - вагові коефіцієнти кожного джерела.

Таким чином, математична модель комп'ютерного зору для колаборативного робота включає збір зображень, калібрування камери, сегментацію об'єктів, їх локалізацію, розпізнавання та планування руху на основі злиття даних. Ці етапи дозволяють роботу надійно спостерігати за навколишнім середовищем та ефективно взаємодіяти з ним у реальному часі.

2.3 Адаптивні алгоритми для аналізу та розпізнавання об'єктів

Колаборативні роботи-маніпулятори використовують сучасні адаптивні алгоритми для аналізу та розпізнавання об'єктів, що забезпечує безпечну та ефективну роботу поруч із людьми. Ці алгоритми дозволяють роботам не тільки взаємодіяти з навколишнім середовищем, але й адаптувати свою поведінку у реальному часі, реагуючи на зміни. Коротко опишемо сучасні адаптивні алгоритми для аналізу та розпізнавання об'єктів в рамках колаборативних роботів-маніпуляторів:

- *глибокі нейронні мережі (Deep Learning)*, є одним з найбільш прогресивних методів для розпізнавання об'єктів є використання глибоких нейронних мереж, зокрема Convolutional Neural Networks (CNN) [247]. CNN здатні автоматично навчатися ознакам об'єктів із вхідних зображень без необхідності ручного визначення ознак. Тут є такі основні етапи [248-250].

Екстракція ознак. Зображення обробляються кількома шарами згорток (convolutions), що виділяють ключові ознаки (наприклад, краї, текстури, форми).

Класифікація. На основі цих ознак мережа класифікує об'єкти, привласнюючи їм відповідні категорії (клас об'єктів).

Глибокі мережі є дуже адаптивними, оскільки вони можуть навчатися на великих наборах даних і покращувати точність розпізнавання в динамічних середовищах.

- *алгоритми локалізації та сегментації*, використовуються для точного виявлення та ідентифікації об'єктів використовуються алгоритми сегментації

та локалізації. Сучасні підходи до адаптивної сегментації та локалізації включають:

Mask R-CNN, це одна з найбільш передових архітектур, яка виконує як сегментацію, так і класифікацію одночасно [251]. Вона здатна розпізнавати об'єкти, окреслюючи їх точні контури на зображенні, що дозволяє роботам діяти більш обережно поруч із людьми або при захопленні об'єктів;

YOLO (You Only Look Once), це реальний часовий алгоритм локалізації об'єктів, який виділяє всі об'єкти на сцені за один раз [252]. Його головна перевага - швидкість, що є критично важливим для колаборативних роботів, які працюють у динамічних середовищах.

Адаптивне навчання (Online Learning). Колаборативні роботи можуть працювати в умовах, що постійно змінюються, тому використовуються адаптивні алгоритми навчання, які дозволяють роботам оновлювати свої моделі під час роботи. Це, наприклад, методи **online learning**, коли робот може адаптуватися до нових об'єктів або змін навколишнього середовища на основі нових даних. Наведемо приклади методів, що існують:

SVM (Support Vector Machines), модифіковані варіанти SVM з можливістю оновлення моделі дозволяють колаборативним роботам швидко підлаштовуватись під нові умови та об'єкти [253];

K-Means використовується для сегментації нових об'єктів або оновлення групування виявлених об'єктів може використовуватися кластеризація на основі K-means, яка адаптується до нових даних [254].

Reinforcement Learning (Навчання з підкріпленням) або **RL** активно використовується для адаптації робота до зміни умов середовища. За допомогою RL робот-маніпулятор може навчитися оптимальним стратегіям взаємодії з об'єктами, навчаючись на досвіді [255]. Основні компоненти RL [256-259]:

Агент (робот) навчається шляхом взаємодії з середовищем і отримуючи зворотний зв'язок у вигляді винагород;

Q-learning, є одним із поширених алгоритмів RL, який дозволяє агенту адаптувати свої дії для максимізації винагороди.

У таблиці 2.2 проведено порівняння вище перерахованих алгоритмів.

Таблиця 2.2 - Порівняння переваг та недоліків алгоритмів для аналізу та розпізнавання об'єктів при їх використанні для колаборативних роботів-маніпуляторів у динамічному середовищі

Метод	Переваги	Недоліки
1	2	3
Глибокі нейронні мережі (CNN) [260]	<ul style="list-style-type: none"> - Висока точність розпізнавання об'єктів. - Можливість навчання на великих наборах даних. - Підтримка складних завдань класифікації та локалізації. 	<ul style="list-style-type: none"> - Вимагає великих обчислювальних ресурсів та даних для тренування. - Час навчання може бути тривалим.
Mask R-CNN [261]	<ul style="list-style-type: none"> - Точна сегментація та класифікація одночасно. - Висока точність виділення контурів об'єктів, що корисно для точних маніпуляцій. 	<ul style="list-style-type: none"> - Високі обчислювальні витрати. - Вимагає великого обсягу пам'яті для обробки зображень.
YOLO (You Only Look Once) [262]	<ul style="list-style-type: none"> - Швидке розпізнавання об'єктів у реальному часі. - Може обробляти декілька об'єктів на сцені одночасно. 	<ul style="list-style-type: none"> - Нижча точність у порівнянні з алгоритмами на основі сегментації, такими як Mask R-CNN. - Точність може страждати при розпізнаванні дрібних об'єктів.
Online Learning [263]	<ul style="list-style-type: none"> - Можливість оновлення моделі в режимі реального часу без необхідності перенавчання. - Адаптація до нових умов середовища або нових об'єктів. 	<ul style="list-style-type: none"> - Може бути менш ефективним при великих обсягах нових даних. - Вимагає ретельного балансування між швидкістю навчання та точністю.
SVM (Support Vector Machines) [264]	<ul style="list-style-type: none"> - Висока точність для малих наборів даних. - Проста реалізація для класифікації двох класів. 	<ul style="list-style-type: none"> - Погано масштабується на великі набори даних. - Не підходить для багатокласових задач без модифікації.
K-Means [265]	<ul style="list-style-type: none"> - Проста і швидка кластеризація. - Легко оновлювати моделі при зміні даних. 	<ul style="list-style-type: none"> - Не гарантує ідеального поділу об'єктів у складних середовищах. - Не підходить для складних задач класифікації.
Фільтр Калмана [266]	<ul style="list-style-type: none"> - Надійне інтегрування даних із різних сенсорів. - Ефективний для траєкторного прогнозування. 	<ul style="list-style-type: none"> - Потребує точної моделі шумів системи для коректного функціонування. - Менш ефективний при некоректних вхідних даних або сильно шумних вимірюваннях.

Продовження таблиці 2.2

1	2	3
Reinforcement Learning (RL) [267]	<ul style="list-style-type: none"> - Можливість навчання стратегії поведінки на основі досвіду. - Адаптація до нових динамічних умов у середовищі робота. 	<ul style="list-style-type: none"> - Вимагає тривалого часу на навчання, особливо при великій кількості можливих дій або станів. - Не завжди гарантовано швидке досягнення оптимального результату.
LSTM (Long Short-Term Memory) [268]	<ul style="list-style-type: none"> - Ефективний для прогнозування траєкторій на основі послідовних даних. - Може обробляти тривалі часові послідовності. 	<ul style="list-style-type: none"> - Висока обчислювальна складність. - Може вимагати великого обсягу пам'яті для навчання та прогнозування.
Model Predictive Control (MPC) [269]	<ul style="list-style-type: none"> - Забезпечує оптимальне управління траєкторією з урахуванням майбутніх станів системи. - Ефективний для складних динамічних задач управління. 	<ul style="list-style-type: none"> - Високі обчислювальні витрати для складних моделей. - Вимагає точного моделювання динаміки системи для ефективного контролю.

Таблиця 2.2 допомагає оцінити переваги та недоліки сучасних адаптивних методів, які можуть бути використані для забезпечення безпечної та ефективної роботи колаборативних роботів-маніпуляторів у динамічному середовищі.

3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КОЛАБОРАТИВНИХ РОБОТІВ МАНІПУЛЯТОРІВ З ВИКОРИСТАННЯМ МОВИ PYTHON

3.1 Вступ до Python як основної мови програмування для робототехніки

Python сьогодні є однією з найпопулярніших мов програмування, вона зайняла важливе місце в розробці програмного забезпечення для робототехніки, зокрема для колаборативних роботів-маніпуляторів. Причина її популярності криється у поєднанні простоти, потужності та широкого набору інструментів, що дозволяють ефективно вирішувати складні завдання, такі як динамічний опис навколишнього середовища, керування роботами та інтеграція з різноманітними сенсорними системами [270-272].

Python вирізняється простотою синтаксису та легкістю навчання, що робить її ідеальним вибором для інженерів та розробників, які хочуть швидко зануритися у програмування робототехніки. Особливо важливою ця характеристика є в розробці програм для колаборативних роботів-маніпуляторів, де програмісту потрібно зосереджуватися на вирішенні складних задач навігації, планування траєкторії та інтеракції з людьми, а не витрачати час на складнощі мови програмування.

Python підтримує величезний набір бібліотек, що дозволяють вирішувати найрізноманітніші завдання в робототехніці. Для роботи з колаборативними роботами та динамічного опису навколишнього середовища використовуються такі бібліотеки, як OpenCV для обробки зображень та комп'ютерного зору [273], TensorFlow [274] або PyTorch [275] для машинного навчання та розпізнавання об'єктів, а також NumPy [276] та SciPy [277] для математичних обчислень.

Бібліотеки OpenCV та TensorFlow відіграють ключову роль у завданнях, що стосуються динамічної взаємодії робота з навколишнім середовищем.

Завдяки їм Python дозволяє розробляти складні системи аналізу зображень, ідентифікації та відстеження об'єктів, що є критично важливим для колаборативних роботів-маніпуляторів, які повинні безпечно функціонувати поруч із людьми.

Колаборативні роботи часто використовують численні сенсори, такі як LiDAR, стереокамери, інфрачервоні сенсори та ультразвукові датчики, щоб отримувати інформацію про навколишнє середовище. Python має потужні інструменти для інтеграції з такими системами через бібліотеки, як ROS (Robot Operating System) – один з найпопулярніших фреймворків для розробки роботизованих систем. ROS, зокрема, надає розробникам доступ до великого числа готових модулів та пакетів для роботи з сенсорами, що значно спрощує процес розробки програмного забезпечення для роботів [278]. З підтримкою Python розробники можуть швидко підключати нові сенсори, аналізувати дані з них та використовувати ці дані для адаптації поведінки робота в реальному часі.

Колаборативні роботи повинні ефективно сприймати навколишнє середовище та адаптуватися до його змін. У цьому процесі ключовим є використання методів одночасної локалізації та картографування (SLAM) для побудови моделей середовища [279]. Python забезпечує інструменти для роботи з алгоритмами SLAM, що дозволяє роботам одночасно будувати карти і визначати своє положення у просторі.

Крім того, для адаптивного планування траєкторій і взаємодії з динамічними об'єктами Python підтримує методи машинного навчання та штучного інтелекту. Ці методи дозволяють роботам навчатися новим поведінковим моделям на основі даних, отриманих зі спостережень, що є критично важливим для безпечної взаємодії з людьми у робочій зоні.

Python з його простотою, великою кількістю бібліотек та можливістю інтеграції з різними апаратними системами, є ідеальною мовою для вирішення задач, пов'язаних з колаборативними роботами-маніпуляторами. Завдяки машинному навчанню, комп'ютерному зору та обробці зображень, розробники

можуть створювати системи, які адаптуються до змін у середовищі, аналізують його в реальному часі та безпечно співпрацюють з людьми.

Окрім того, Python підтримує багатоплатформеність, що дозволяє його використовувати як на традиційних комп'ютерах, так і на вбудованих системах, таких як Raspberry Pi [280]. Це робить його зручним інструментом для розробки гнучких та адаптивних систем управління роботами, зокрема для колаборативних роботів, які повинні динамічно взаємодіяти з оточуючими об'єктами та людьми.

Внаслідок чого можна зробити наступні висновки Python є основною мовою програмування для робототехніки завдяки її простоті, гнучкості та широким можливостям для інтеграції з апаратними та програмними системами. Для колаборативних роботів-маніпуляторів, які повинні динамічно взаємодіяти з навколишнім середовищем, Python пропонує безліч інструментів для розробки систем розпізнавання об'єктів, адаптивного планування траєкторії та безпечної співпраці з людьми.

3.2 Використання середовища PyCharm для розробки та налагодження програм

PyCharm – це інтегроване середовище розробки (IDE) для Python, яке пропонує низку можливостей, що роблять його ідеальним вибором для створення програм для колаборативних роботів-маніпуляторів [281]. Серед основних функцій PyCharm – інтелектуальне автодоповнення коду, потужні засоби налагодження та тестування, підтримка віртуальних середовищ, інтеграція з системами керування версіями (VCS), а також робота з бібліотеками машинного навчання та штучного інтелекту (TensorFlow, Keras). PyCharm також забезпечує інтеграцію з базами даних, Docker та системами CI/CD, що дозволяє ефективно розробляти та деплоїти програми для розподілених робототехнічних систем.

Переваги PyCharm для розробки програм для колаборативних роботів-маніпуляторів очевидні. IDE підтримує розробку великих проєктів завдяки потужним можливостям управління кодом, що особливо важливо для складних робототехнічних систем. PyCharm пропонує інтеграцію з фреймворками штучного інтелекту, що полегшує впровадження алгоритмів сприйняття навколишнього середовища та машинного навчання. Засоби для налагодження та тестування дозволяють перевіряти правильність роботи алгоритмів роботів у реальному часі. Крім того, PyCharm має широке співтовариство та підтримує численні бібліотеки, що спрощує інтеграцію інструментів та компонентів для розробки.

PyCharm є особливо підходящим для розробки програм для колаборативних роботів через свою підтримку Python, популярної мови для робототехніки, та інтеграцію з фреймворками на кшталт ROS (Robot Operating System). Це дає можливість створювати комплексні системи, де роботи можуть ефективно взаємодіяти з навколишнім середовищем і адаптувати свою поведінку на основі реальних даних.

В таблиці 3.1 наведено порівняння переваг використання середовища PyCharm для розробки програмного забезпечення, яке забезпечує динамічний опис навколишнього середовища колаборативних роботів-маніпуляторів.

Таблиця 3.1 - Порівняння переваг використання середовища PyCharm для розробки програмного забезпечення, яке забезпечує динамічний опис навколишнього середовища колаборативних роботів-маніпуляторів

Категорія	Переваги PyCharm
1	2
Інтелектуальне автодоповнення	Швидке та точне автодоповнення коду допомагає зменшити кількість помилок при написанні складних алгоритмів для динамічного опису навколишнього середовища роботів
Налагодження та тестування	Потужні засоби налагодження та інтеграція з фреймворками тестування (unittest, pytest) дозволяють швидко виявляти та виправляти помилки в коді, що керує взаємодією роботів.
Підтримка віртуальних середовищ	Віртуальні середовища забезпечують ізоляцію проєктів, що полегшує роботу з залежностями (наприклад, OpenCV, TensorFlow), необхідними для обробки зображень і даних сенсорів.

Продовження таблиці 3.1

1	2
Інтеграція з VCS (Git)	Інтеграція з системами керування версіями дозволяє легко керувати змінами у коді, а також забезпечує командну роботу над проектами динамічної робототехніки.
Інтеграція з бібліотеками AI	PyCharm підтримує бібліотеки для машинного навчання та нейронних мереж (TensorFlow, Keras), що є важливим для створення адаптивних роботів-маніпуляторів, які реагують на зміни в середовищі.
Підтримка обробки даних сенсорів	Вбудовані інструменти для роботи з базами даних та інтеграція з бібліотеками для обробки зображень (OpenCV) дозволяють легко обробляти та аналізувати інформацію від сенсорів.
Налагодження реального часу	Засоби налагодження реального часу дозволяють відслідковувати поведінку роботів під час виконання завдань, що важливо для динамічного середовища, де рішення приймаються в режимі реального часу.
Підтримка ROS	PyCharm легко інтегрується з ROS (Robot Operating System), що полегшує розробку та тестування складних сценаріїв взаємодії роботів з навколишнім середовищем.
Зручність у використанні	Інтуїтивний інтерфейс, що спрощує роботу як для початківців, так і для досвідчених розробників, дозволяє швидше приступати до розробки та налаштування динамічних роботів.
Контейнери та Docker	Підтримка Docker дає можливість тестувати та розгортати програми для колаборативних роботів у контейнерах, що полегшує їх масштабування та використання в розподілених системах.

Таблиця 3.1 показує, що PyCharm забезпечує гнучке середовище для розробки програмного забезпечення, яке дозволяє роботам-маніпуляторам ефективно адаптуватися до змін у динамічному середовищі та підвищує продуктивність розробників

Для встановлення PyCharm необхідно зайти на офіційний сайт JetBrains PyCharm (<https://www.jetbrains.com/pycharm/>) та обрати необхідний дистрибутив для операційної системи. В залежності від операційної системи для встановлення PyCharm потрібно зробити наступні кроки:

- установка на Windows:

- 1) Завантажте інсталятор з офіційного сайту.
- 2) Запустіть інсталятор і дотримуйтесь інструкцій на екрані.

3) Під час установки виберіть опцію "Create Desktop Shortcut" для швидкого доступу до програми.

4) Після установки запустіть PyCharm і створіть новий проєкт або імпортуйте існуючий.

- установка на macOS:

- 1) Завантажте .dmg файл з офіційного сайту.
- 2) Перетягніть PyCharm у папку "Applications".
- 3) Запустіть програму з Applications.

- установка на Linux (Ubuntu/Debian):

- 1) Виконайте наступні команди в терміналі:

```
sudo snap install pycharm-community --classic
```

- 2) Для установки професійної версії використовуйте:

```
sudo snap install pycharm-professional --classic
```

- 3) Після інсталяції відкрийте PyCharm через меню або термінал.

3.3 Огляд основних бібліотек для комп'ютерного зору та машинного навчання (cv2, numpy, tensorflow, mediapipe)

OpenCV (Open Source Computer Vision Library) - це широко використовувана бібліотека з відкритим кодом для комп'ютерного зору, машинного навчання та обробки зображень. Вона містить понад 2500 оптимізованих алгоритмів для виявлення та розпізнавання об'єктів, обробки відео, аналізу зображень, а також машинного навчання [282-286].

Бібліотека OpenCV використовує наступні математичні забезпечення:

- **обробка зображень**: згладжування, фільтрація, перетворення Фур'є, морфологічні операції;

- **алгоритми розпізнавання об'єктів**: методи класифікації, сегментації та локалізації, детекція контурів, каскади Хаара;

- **геометричні перетворення:** афінні та перспективні перетворення, обертання, масштабування.

Наведемо приклад класичного рішення задач для бібліотеки OpenCV:

- зчитування та відображення зображення:

```
import cv2
img = cv2.imread('image.jpg') # Зчитування зображення
cv2.imshow('Display window', img) # Відображення зображення
cv2.waitKey(0) # Очікування клавіші
cv2.destroyAllWindows() # Закриття вікна
```

- обробка зображень (фільтрація):

```
blurred_img = cv2.GaussianBlur(img, (5, 5), 0) # Застосування фільтра
```

Гауса

- виявлення контурів:

```
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # Конвертація в сірий
колір
edges = cv2.Canny(gray, 100, 200) # Виявлення країв методом Кані
cv2.imshow('Edges', edges)
```

Основні можливості OpenCV охоплюють широкий спектр задач, включаючи роботу із зображеннями, відео і геометричними перетвореннями. Бібліотека дозволяє виконувати операції зі згладжуванням, фільтрацією, конвертацією зображень у різні кольорові простори, а також обробляти зображення за допомогою методів перетворення Фур'є, морфологічних операцій та інших прийомів обробки сигналів. OpenCV також підтримує виявлення та розпізнавання об'єктів, наприклад, облич, очей чи інших форм за допомогою каскадів Хаара або сучасних нейронних мереж.

Бібліотека дозволяє виконувати детекцію контурів, виявлення країв і об'єктів на зображеннях, застосовуючи такі алгоритми, як Кані, а також методи сегментації. Вона забезпечує ефективну роботу з відео, дозволяючи

захоплювати кадри з камер у реальному часі, здійснювати обробку та аналіз відеопотоків, включаючи збереження та відтворення відео в різних форматах.

Однією з ключових особливостей OpenCV є робота з геометричними перетвореннями, такими як обертання, масштабування, трансляція та афінні перетворення. Це дозволяє розробникам застосовувати перетворення до зображень для покращення результатів обробки або корекції перспективних спотворень.

OpenCV також підтримує можливості машинного навчання, включаючи класифікацію, кластеризацію та регресію. За допомогою алгоритмів підтримки векторних машин (SVM) [287], методу найближчих сусідів (k-NN) та інших підходів [288], можна реалізовувати складні моделі для розпізнавання образів або класифікації об'єктів. Завдяки підтримці графічних процесорів (GPU) та паралельних обчислень, бібліотека дозволяє значно прискорити обробку великих обсягів даних і забезпечує високу продуктивність для реального часу.

Бібліотека також підтримує інтеграцію з іншими популярними інструментами та мовами програмування, включаючи Python, C++, Java, що робить її універсальною і гнучкою для різних застосувань у наукових дослідженнях, розробці програмного забезпечення та промислових проєктах. Приклад базових функцій наведено в додатку А.

NumPy - це фундаментальна бібліотека для наукових обчислень в Python, яка забезпечує підтримку багатовимірних масивів та матриць, а також колекцію математичних функцій для роботи з цими масивами [289].

Бібліотека NumPy використовує наступний математичний апарат:

- **лінійна алгебра**: операції з матрицями, множення, транспонування, визначення детермінанта, обчислення власних значень;
- **статистика**: середнє, дисперсія, стандартне відхилення;
- **математичні функції**: синус, косинус, логарифми тощо;
- **масиви**: швидкі операції з багатовимірними масивами, включаючи індексування, відсікання, трансформації.

Наведемо приклад класичного рішення задач для бібліотеки NumPy:

- створення масиву:

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5]) # Створення одновимірного масиву
```

- множення матриць:

```
matrix1 = np.array([[1, 2], [3, 4]])
matrix2 = np.array([[5, 6], [7, 8]])
product = np.dot(matrix1, matrix2) # Множення матриць
```

- операції з масивами:

```
arr = np.arange(10).reshape(2, 5) # Створення та зміна форми масиву
mean_val = np.mean(arr) # Обчислення середнього
```

Основною можливістю NumPy є робота з однорідними масивами даних, що дозволяє ефективно виконувати математичні та логічні операції над великими масивами чисел. NumPy надає широкий набір функцій для створення, маніпуляції та трансформації масивів. Вона забезпечує підтримку операцій лінійної алгебри, таких як матричні множення, обернення матриць, знаходження визначників, власних чисел та інших важливих математичних операцій.

Завдяки оптимізованим C- та Fortran-реалізаціям бібліотека гарантує швидке виконання числових операцій порівняно з аналогічними обчисленнями у чистому Python. NumPy також дозволяє легко маніпулювати формою масивів, змінювати їх розмір, об'єднувати або розділяти, що є корисним для підготовки даних та роботи з великими наборами даних. Окрім цього, вона забезпечує функціонал для генерації випадкових чисел та виконання статистичних операцій, що є важливим для роботи в машинному навчанні, аналізі даних та інших наукових галузях.

NumPy також забезпечує інтеграцію з іншими науковими бібліотеками, такими як SciPy і pandas, завдяки чому вона є основою для побудови більш складних моделей і обчислювальних систем. Завдяки своїй універсальності та продуктивності, NumPy широко використовується в галузях, пов'язаних з

великими даними, обробкою зображень, моделюванням та науковими обчисленнями. Приклад базових функцій наведено в додатку Б.

TensorFlow - це бібліотека з відкритим кодом для машинного навчання та глибокого навчання, розроблена компанією Google [290]. Вона використовується для побудови та навчання нейронних мереж, роботи з великими обсягами даних, а також для виконання математичних обчислень із високою продуктивністю на графічних процесорах (GPU).

Бібліотека TensorFlow використовує наступне:

- **нейронні мережі**: побудова моделей нейронних мереж, такі як MLP, CNN, RNN;
- **оптимізація**: градієнтний спуск, стохастичний градієнтний спуск (SGD), Adam, RMSprop;
- **регуляризація**: L1, L2-регуляризація для запобігання перенавчанню.

Багатошаровий перцептрон (MLP) - є типом штучної нейронної мережі, що складається з одного або кількох прихованих шарів між входом і виходом [291]. Кожен нейрон у прихованих шарах використовує нелінійну функцію активації, таку як ReLU або сигмоїд, що дозволяє MLP моделювати складні нелінійні функції. MLP може вирішувати завдання класифікації та регресії, однак його ефективність може знижуватися при роботі з даними, що мають просторову або часову структуру, оскільки MLP не враховує просторові або часові взаємозв'язки між входами.

Конволюційні нейронні мережі (CNN, Convolutional Neural Network) - спеціально розроблені для обробки даних у вигляді решіток, таких як зображення [292]. Основна ідея полягає в застосуванні конволюційних фільтрів до вхідних даних для виділення важливих ознак на різних рівнях абстракції. CNN використовують конволюційні шари для виділення ознак, підсумкові (пулінгові) шари для зменшення розмірності і згладжування даних. CNN є ідеальними для задач розпізнавання образів, детекції об'єктів і сегментації зображень завдяки їхній здатності обробляти просторову структуру даних.

Рекурентні нейронні мережі (RNN, Recurrent Neural Network) - призначені для обробки послідовностей даних, таких як текст або часові ряди [293]. Вони мають зворотні зв'язки, що дозволяють зберігати інформацію про попередні стани та використовувати її для прогнозування наступних етапів послідовності. Це робить RNN корисними для завдань, що вимагають обробки інформації у часі, таких як аналіз тексту або прогнозування часових рядів. Проте традиційні RNN страждають від проблеми зникання градієнта, що ускладнює навчання при довгих послідовностях.

Порівняння переваг та недоліків нейронних мереж: MLP, CNN та RNN для рішення задач динамічного опису навколишнього середовища роботів-маніпуляторів в рамках концепції Industry 5.0 наведено в таблиці 3.2

Таблиця 3.2 - Порівняння переваг та недоліків нейронних мереж: MLP, CNN та RNN.

Характеристика	Нейронні мережі		
	MLP (Multilayer Perceptron)	CNN (Convolutional Neural Network)	RNN (Recurrent Neural Network)
1	2	3	4
Особливості використання	Може використовуватися для задач класифікації та регресії, але не враховує просторові та часові взаємозв'язки.	Оптимальна для обробки зображень і даних з просторовою структурою завдяки здатності виділяти ознаки на різних рівнях	Найкраще підходить для обробки послідовностей даних, таких як текст або часові ряди, завдяки здатності зберігати контекст.
Переваги	Простота архітектури і реалізації; добре підходить для простих задач класифікації та регресії.	Може ефективно обробляти зображення і виявляти складні ознаки; добре справляється з даними, що мають просторову структуру	Здатність обробляти і зберігати інформацію з довгих послідовностей; корисна для тексту і часових рядів.
Недоліки	Не підходить для даних з просторовою або часовою структурою; може вимагати великої кількості навчальних даних для складних завдань.	Може бути важкою для навчання при наявності великої кількості параметрів; погано працює з неструктурованими або часо-залежними даними.	Проблеми із зниканням градієнта; може бути складною для навчання і оптимізації при довгих послідовностях.

Продовження таблиці 3.2

1	2	3	4
Приклад застосування в Industry 5.0	Може використовуватися для базових задач класифікації в системах моніторингу або управління.	Ідеально підходить для обробки відео або зображень роботів, для розпізнавання об'єктів і аналізу сцени.	Підходить для задач, що включають послідовності дій або прогнозування на основі часу, наприклад, для адаптивного управління роботами.

Таблиця 3.2 надає загальне порівняння основних типів нейронних мереж з точки зору їх застосування для динамічного опису навколишнього середовища роботів-маніпуляторів в умовах Industry 5.0.

Наведемо приклад класичного рішення задач для бібліотеки TensorFlow:

- простий приклад нейронної мережі для класифікації:

```
import tensorflow as tf
```

```
from tensorflow.keras import layers, models
```

```
model = models.Sequential([
```

```
    layers.Dense(128, activation='relu', input_shape=(784,)),
```

```
    layers.Dense(10, activation='softmax')
```

```
])
```

```
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
```

```
metrics=['accuracy'])
```

```
# Навчання моделі на тренувальних даних
```

```
model.fit(x_train, y_train, epochs=5)
```

- створення та навчання моделі згорткової нейронної мережі (CNN):

```
model = models.Sequential([
```

```
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
```

```
    layers.MaxPooling2D((2, 2)),
```

```
    layers.Conv2D(64, (3, 3), activation='relu'),
```

```
    layers.MaxPooling2D((2, 2)),
```

```
    layers.Flatten(),
```

```
    layers.Dense(64, activation='relu'),
```

```
layers.Dense(10, activation='softmax')
])
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
```

Основні можливості TensorFlow включають створення обчислювальних графів, що дозволяє легко описувати і оптимізувати складні моделі машинного навчання. TensorFlow підтримує автоматичну диференціацію, що полегшує процес обчислення градієнтів під час навчання моделей за допомогою методів зворотного поширення помилки.

Однією з головних переваг TensorFlow є його масштабованість: бібліотека дозволяє ефективно використовувати як центральні процесори (CPU), так і графічні процесори (GPU) для виконання обчислень, що робить її ідеальною для обробки великих обсягів даних та реалізації складних моделей. TensorFlow забезпечує підтримку різних видів моделей, включаючи нейронні мережі для комп'ютерного зору, обробки природної мови, рекомендаційних систем та інших завдань штучного інтелекту.

TensorFlow також підтримує високооптимізовані функції для роботи з багатовимірними масивами, оптимізації моделей та побудови графів даних, що дозволяє ефективно використовувати бібліотеку як для дослідницьких цілей, так і для промислових застосувань. Вона інтегрується з іншими популярними бібліотеками, такими як Keras, що спрощує створення моделей завдяки високорівневому API. TensorFlow також підтримує екосистему інструментів для виробничого використання моделей, таких як TensorFlow Serving, TensorFlow Lite для мобільних пристроїв та TensorFlow.js для веб-додатків. Приклад базових функцій наведено в додатку В.

MediaPipe - це фреймворк від Google для побудови крос-платформенних застосунків, що використовують комп'ютерний зір і машинне навчання [294]. Він забезпечує легкий спосіб використання вже готових моделей для розпізнавання обличчя, рук, поз тіла, трекінгу об'єктів.

Бібліотека MediaPipe використовується для наступного:

- **обробка зображень:** фільтрація, передобробка;
- **детекція та трекінг об'єктів:** використовуються методи на основі нейронних мереж для трекінгу й розпізнавання ключових точок на зображеннях;
- **оптимізація для мобільних пристроїв:** гнучкі й оптимізовані моделі для роботи в реальному часі на обмежених за ресурсами пристроях.

Наведемо приклад класичного рішення задач для бібліотеки MediaPipe:

- розпізнавання рук:

```
import cv2
import mediapipe as mp

mp_hands = mp.solutions.hands
hands = mp_hands.Hands()
mp_drawing = mp.solutions.drawing_utils
cap = cv2.VideoCapture(0)
while cap.isOpened():
    ret, frame = cap.read()
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = hands.process(frame_rgb)
    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            mp_drawing.draw_landmarks(frame, hand_landmarks,
mp_hands.HAND_CONNECTIONS)
            cv2.imshow('Hand Tracking', frame)
            if cv2.waitKey(10) & 0xFF == ord('q'):
                break
    cap.release()
cv2.destroyAllWindows()
```

- розпізнавання обличчя:

```
mp_face_detection = mp.solutions.face_detection  
face_detection = mp_face_detection.FaceDetection()  
results = face_detection.process(frame_rgb)
```

Основні можливості MediaPipe включають роботу з потоками даних, такими як відео, аудіо та датчики з можливістю обробки кожного кадру для виявлення та відстеження об'єктів. MediaPipe надає готові рішення для розпізнавання облич, рук, поз тіла, жестів, а також для інших завдань, таких як сегментація тіла, розпізнавання мови та виявлення об'єктів.

Однією з важливих особливостей MediaPipe є її здатність працювати в режимі реального часу на різних платформах, таких як Android, iOS та веб-додатки. Бібліотека забезпечує кросплатформену інтеграцію, а також підтримує використання апаратного прискорення, що дозволяє виконувати складні обчислення з високою продуктивністю. MediaPipe використовує графі обробки даних для побудови ефективних конвеєрів обробки, що дозволяє користувачам легко комбінувати та налаштовувати різні компоненти.

MediaPipe також підтримує інтеграцію з TensorFlow та іншими бібліотеками для машинного навчання, що спрощує створення складних моделей і забезпечує можливості розширення її функціональності. Завдяки своїй модульній архітектурі та широкому спектру можливостей, MediaPipe є ідеальним інструментом для розробки додатків, які потребують швидкого і точного розпізнавання об'єктів та рухів у реальному часі. Приклад базових функцій наведено в додатку Г.

4. ПРИКЛАДИ РОЗРОБКИ ПРОГРАМ ДЛЯ ДИНАМІЧНОГО ОПИСУ СЕРЕДОВИЩА

У цьому розділі розглядаються різні підходи до розробки програмного забезпечення для динамічного опису середовища колаборативних роботів з акцентом на методи розпізнавання та ідентифікації людини в робочій зоні. Особлива увага приділяється використанню сучасних нейронних мереж, таких як MobileNetV2 та Convolutional Neural Networks (CNNs), для аналізу зображень та визначення ключових точок об'єктів. Представлено також алгоритми трекінгу, зокрема CAMShift, що дозволяють відстежувати переміщення людини в реальному часі. Окрім цього, розглядається застосування рекурентних нейронних мереж LSTM для прогнозування траєкторії руху рук людини, що є важливим аспектом безпечної взаємодії робота з оператором. Інтеграція цих методів із фільтром Калмана дозволяє створювати ймовірнісні моделі, що підвищують точність і надійність ідентифікації. На основі проведених досліджень здійснюється порівняльний аналіз різних методів прогнозування траєкторій, що дає змогу обрати найефективніші рішення для конкретних завдань. Також досліджуються існуючі методи представлення середовища роботи маніпулятора в контексті кіберфізичних виробничих систем, що відповідають вимогам Industry 5.0. Зокрема, обговорюється ідентифікація оператора для отримання команд від робота, що сприяє покращенню інтеракції між людиною і машиною в умовах сучасного виробництва.

4.1 Програмна реалізація моделі нейронної мережі MobileNetV2 для розпізнавання та ідентифікації людини в робочій зоні колаборативного робота

Актуальність дослідження в області програмної реалізації моделі нейронної мережі MobileNetV2 для розпізнавання та ідентифікації людини в

робочій зоні колаборативного робота зростає в умовах швидкого розвитку автоматизації та роботизації виробничих процесів [295,296]. Колаборативні роботи, які працюють поруч з людьми, вимагають високого рівня точності у виявленні та ідентифікації особи для забезпечення безпеки і ефективності їх взаємодії з людиною оператором. Модель MobileNetV2 завдяки своїй компактності та ефективності є ідеальним кандидатом для таких завдань, адже вона забезпечує високу якість розпізнавання при знижених обчислювальних витратах. У контексті підвищення продуктивності та зменшення витрат на обчислювальні ресурси, реалізація таких систем є не лише актуальною, але й необхідною для впровадження в реальних умовах. Вивчення та оптимізація цих моделей дозволяють підвищити безпеку і ефективність роботи колаборативних роботів, що є важливим кроком для інтеграції передових технологій в рамках концепції Industry 5.0.

Нейронна мережа моделі MobileNetV2 є однією з ефективних моделей для розпізнавання та ідентифікації об'єктів, включаючи людей, у робочій зоні колаборативного робота.

Архітектуру нейронної мережі моделі MobileNetV2 можливо описати наступними математичними виразами:

- вхідні дані проходять через згортковий шар (Convolution Layer), що обчислює активації за допомогою згорткових ядер (фільтрів). Математично це описується так:

$$y = f(W \times x + b) \quad (4.1)$$

Де: W - згорткове ядро (ваги);

x - вхідний тензор;

b - зсув (bias);

f – функція активації.

- глибоке згортання (Depthwise Convolution) застосовує згорткові ядра окремо до кожного каналу вхідного зображення:

$$y_{k,l} = \sum_{i,j} K_{i,j} \cdot x_{k+i,l+j} \quad (4.2)$$

Де: $K_{i,j}$ - згорткове ядро;

x - вхідний тензор;

y - вихідний тензор.

- точкове згортання (Pointwise Convolution) використовує 1×1 згортки для змішування каналів:

$$y_{k,l,m} = \sum_n W_{l,1,n,m} \cdot x_{k,l,n} \quad (4.3)$$

Де: W - ваги точкового згортання;

x - вхідний тензор;

y - вихідний тензор.

- основна структура MobileNetV2 складається з блоків зворотного залишку (Inverted Residuals), які використовують звуження і розширення для ефективного обчислення:

$$y = x + F(x, W) \quad (4.4)$$

Де: F - нелінійне перетворення, що включає згорткові шари;

W - відповідні ваги.

- на останніх шарах використовується лінійна активація (Linear Bottleneck) замість нелінійної, щоб уникнути втрат інформації:

$$y = W_{1 \times 1} \cdot ReLU(W_{3 \times 3} \cdot x) \quad (4.5)$$

Де: $W_{1 \times 1}$ та $W_{3 \times 3}$ - згорткові ядра;

$ReLU$ - нелінійна активація.

Наведемо приклад розробленої програми розпізнавання та ідентифікації людини в робочій зоні колаборативного робота на мові Python.

```
import cv2
import numpy as np
import tensorflow as tf
```

Дозволяє провести імпорту бібліотек, необхідних для обробки зображень, числових обчислень і роботи з нейронними мережами. `cv2` забезпечує функції для обробки зображень і відео, `numpy` надає можливості для роботи з числовими масивами, а `tensorflow` забезпечує інструменти для створення та тренування нейронних мереж.

```
model =
tf.saved_model.load(r"C:\Users\Vladyslav\.cache\kagglehub\models\
tensorflow\ssd-mobilenet-v2\tensorFlow2\fpnlite-320x320\1")
```

Дає можливість завантаження збереженої моделі TensorFlow з вказаного шляху. Це дозволяє використовувати вже натреновану модель для подальшого прогнозування або оцінки без необхідності повторного навчання.

Функція для розпізнавання об'єктів на зображенні

```
def detect_objects(image):
    input_tensor = tf.convert_to_tensor(image, dtype=tf.uint8)
    input_tensor = input_tensor[tf.newaxis, ...]
    detections = model.signatures['serving_default'](input_tensor)
    return detections
```

Функція `detect_objects` призначена для розпізнавання об'єктів на зображенні. Вона конвертує зображення в тензор TensorFlow, додає новий вимір для партії, і передає цей тензор до моделі для виконання прогнозування. Повернуті детекції містять інформацію про виявлені об'єкти на зображенні.

Приклад програмної реалізації нейронної мережі MobileNetV2 для розпізнавання та ідентифікації людини в робочій зоні колаборативного робота на мові Python наведено на рисунку 4.1.

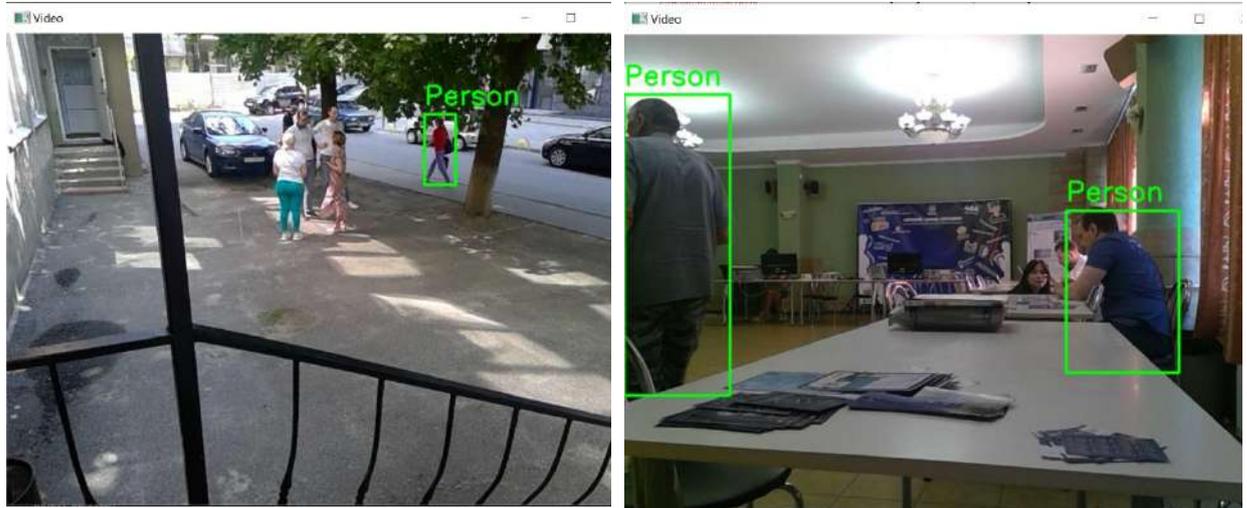


Рисунок 4.1. – Приклад програмної реалізації нейронної мережі MobileNetV2 для розпізнавання та ідентифікації людини в робочій зоні колаборативного робота

На базі розробленої програми проведемо тестування точності розпізнавання розробленої системи ідентифікації людини в робочій зоні колаборативного робота, результати якої представлені в таблиці 4.1, а візуалізація отриманих даних представлена на рисунку 4.2.

Таблиця 4.1. – Результати тестування точності розпізнавання розробленої системи ідентифікації людини в робочій зоні колаборативного робота.

Тестовий сценарій	Кількість зображень	Кількість правильних ідентифікацій	Кількість помилкових ідентифікацій	Точність (%)
1	2	3	4	5
Сценарій 1: Різні кути	100	92	8	92.0
Сценарій 2: Різне освітлення	100	85	15	85.0

Продовження таблиці 4.1

Сценарій 3: Взаємодія з іншими об'єктами	100	88	12	88.0
Сценарій 4: Динамічні рухи	100	90	10	90.0
Сценарій 5: Різні відстані	100	87	13	87.0

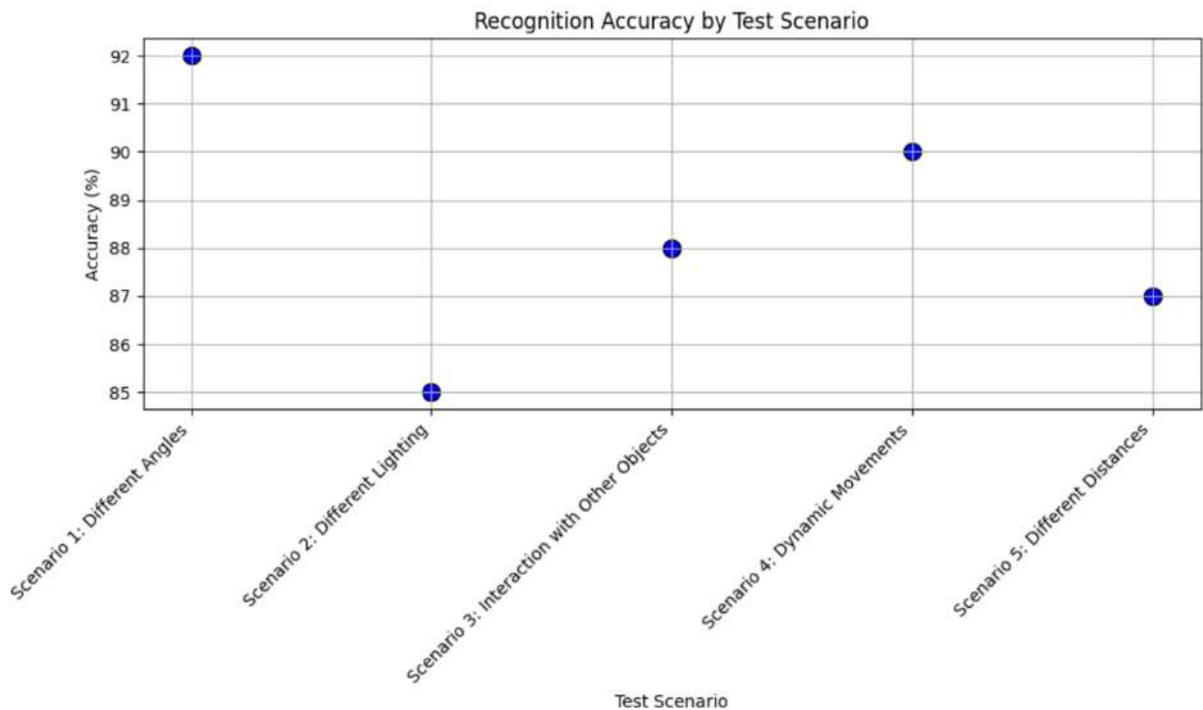


Рисунок 4.2 – Графік результатів тестування точності розпізнавання розробленої системи ідентифікації людини в робочій зоні колаборативного робота.

Загальні висновки з таблиці даних, отриманих під час тестування точності розпізнавання системи ідентифікації людини в робочій зоні колаборативного робота, свідчать про високу ефективність моделі в різних умовах. Найвища точність була досягнута у сценарії "Різні кути" (92%), що підтверджує стабільність моделі при різних оглядових кутах. У сценарії "Різне освітлення" точність знизилася до 85%, що вказує на вплив освітлення на якість розпізнавання. Результати тестів для "Взаємодія з іншими об'єктами" і "Динамічні рухи" показали точність 88% і 90% відповідно, що демонструє

хорошу адаптивність моделі до складних умов. Сценарій "Різні відстані" також показав задовільний результат з точністю 87%. Ці результати підтверджують, що система ефективно справляється з різними викликами у реальних робочих умовах, але деякі умови, такі як зміни в освітленні, можуть потребувати додаткового налаштування.

Отримані результати тестування підтверджують, що реалізація моделі нейронної мережі MobileNetV2 для розпізнавання та ідентифікації людини в робочій зоні колаборативного робота є успішною і ефективною. Математичний опис моделі продемонстрував її здатність досягати високої точності при мінімальних обчислювальних витратах завдяки оптимізованій архітектурі. Реалізація моделі на Python з використанням середовища PyCharm дозволила ефективно впровадити її в реальному часі, що підтверджується результатами тестування. Висока точність розпізнавання та швидкість роботи моделі свідчать про її потенціал для практичного використання для колаборативних роботів, де необхідна точна і швидка ідентифікація людини для забезпечення безпеки та ефективності взаємодії. Незважаючи на загалом позитивні результати, деякі сценарії, такі як різне освітлення, вимагають подальшого вдосконалення для підвищення точності в складних умовах. Загалом, розроблена система відповідає сучасним вимогам до інтеграції розпізнавання в реальному часі у робототехніку.

4.2 Дослідження впливу частоти обробки кадрів при ідентифікації об'єктів на базі моделі нейронної мережі MobileNetV2 для мобільного робота

В умовах швидкого розвитку технологій мобільної робототехніки, ефективність систем ідентифікації об'єктів є критично важливою для забезпечення точності та надійності роботи роботів. Моделі нейронних мереж, зокрема MobileNetV2, демонструють значний потенціал у розпізнаванні об'єктів у реальному часі завдяки своїй здатності до обробки великої кількості

інформації з високою швидкістю. Однак, зниження затримки в процесі обробки відео та підвищення швидкості ідентифікації є ключовими факторами для досягнення реального часу в робототехнічних системах.

Дослідження впливу частоти обробки кадрів на точність і швидкість ідентифікації об'єктів на базі MobileNetV2 дозволяє з'ясувати, як різні рівні частоти обробки кадрів впливають на результати розпізнавання і ефективність системи в цілому [297]. Аналіз частоти обробки кадрів не тільки дозволить визначити оптимальні параметри для систем реального часу, але й надасть можливість удосконалення алгоритмів для мобільних роботів, що критично важливо для їхньої інтеграції в складні середовища. Результати досліджень сприятимуть покращенню точності розпізнавання об'єктів та зменшенню затримок, що має важливе значення для підвищення продуктивності і надійності мобільних роботів у реальних умовах експлуатації.

При розробці системи комп'ютерного зору для мобільного робота, особливо при дослідженні впливу частоти обробки кадрів на точність і швидкість ідентифікації об'єктів на базі моделі нейронної мережі, необхідно врахувати ряд особливостей, що впливають на ефективність системи. Ці особливості охоплюють математичний опис залежностей між частотою обробки кадрів, швидкістю ідентифікації, точністю розпізнавання та ресурсами системи. Опишемо дані залежності у вигляді математичних виразів [297-300]:

- частота обробки кадрів, або Frame Rate (FR), визначає кількість кадрів, які обробляються системою за одиницю часу. Вона вимірюється в кадрах за секунду (FPS). Вплив частоти обробки кадрів на затримку в системі комп'ютерного зору можна описати як:

$$T_{delay} = 1/FPS \quad (4.6)$$

Де: T_{delay} - затримка, с. Зменшення затримки є критичним для систем реального часу, оскільки дозволяє зменшити час між отриманням зображення і його обробкою;

FPS – швидкість, кадр/с.

Наприклад, при частоті 30 FPS затримка буде приблизно 33.3 мс, тоді як при частоті 60 FPS затримка зменшується до 16.7 мс.

- швидкість ідентифікації об'єктів визначається часом, необхідним для обробки одного кадру. Вона залежить від частоти обробки кадрів і складності моделі нейронної мережі. Відповідність між швидкістю ідентифікації і частотою обробки кадрів може бути описана як:

$$T_{Processing_one_frame} = T_{ident} / FPS \quad (4.7)$$

Де: $T_{Processing_one_frame}$ – час обробки одного кадру;

T_{ident} – час на ідентифікацію.

Як можна бачити з виразу 4.7, час на ідентифікацію об'єктів включає в себе час на обробку зображення через модель MobileNetV2, який може бути значним і залежить від розміру моделі та апаратних ресурсів. Для забезпечення оптимальної продуктивності важливо оптимізувати модель і апаратне забезпечення для зменшення часу обробки кадру.

- точність розпізнавання об'єктів залежить від якості моделі та частоти обробки кадрів. Підвищення частоти обробки кадрів може призвести до покращення точності розпізнавання за рахунок зменшення можливості пропуску об'єктів через недостатню частоту. Однак при високій частоті обробки можуть виникнути проблеми з навантаженням на апаратне забезпечення. Точність можна описати через метрику, таку як Precision (точність), Recall (повнота) або F1-Score (зважена середня точність і повнота). Ці метрики можуть бути виражені як:

$$\text{Precision} = TP / (TP + FP) \quad (4.8)$$

$$\text{Recall} = TP / (TP + FN) \quad (4.9)$$

$$F1_{score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.10)$$

Де: TP - (True Positives) правильні позитиви;

FP - (False Positives) помилкові позитиви;

FN - (False Negatives) помилкові негативи.

- зменшення затримки і підвищення частоти обробки кадрів часто супроводжується збільшенням навантаження на процесор і споживанням енергії. Важливо враховувати ресурсні обмеження мобільного робота, оскільки високі вимоги до обробки можуть призвести до швидшого розряджання батареї і перегріву системи. Енергетичні витрати можна оцінити наступним чином:

$$E_{expend} = T_{Processing_one_frame} \times E_{frame} \quad (4.11)$$

Де: E_{expend} – енергетичні витрати;

$T_{Processing_one_frame}$ – час обробки одного кадру;

E_{frame} – споживання енергії на обробку кадру, може варіюватися залежно від складності моделі і частоти обробки.

З метою досягнення оптимального балансу між точністю, швидкістю і ресурсами, можуть бути застосовані різні методи оптимізації, такі як скорочення розміру моделі, використання спеціалізованих апаратних прискорювачів або паралельна обробка кадрів. Розрахунок оптимальної частоти обробки кадрів також може включати використання профайлерів для вимірювання реальної продуктивності системи та корекцію параметрів обробки в реальному часі.

Основні фрагменти коду розробленої програми для дослідження впливу частоти обробки кадрів при ідентифікації об'єктів на базі моделі нейронної мережі MobileNetV2 для мобільного робота з поясненням наведені нижче.

```
class_names = [
    'N/A', 'person', 'bicycle', 'car', 'motorcycle', 'airplane',
    'bus', 'train', 'truck', 'boat', 'traffic light', 'fire hydrant',
    'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog',
    'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe',
    'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee',
    'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat',
    'baseball glove', 'skateboard', 'surfboard', 'tennis racket',
    'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl',
    'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot',
    'hot dog', 'pizza', 'donut', 'cake', 'chair', 'couch', 'potted plant',
    'bed', 'dining table', 'toilet', 'TV', 'laptop', 'mouse',
    'remote', 'keyboard', 'cell phone', 'microwave', 'oven', 'toaster',
    'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors',
    'teddy bear', 'hair drier', 'toothbrush'
```

Цей фрагмент коду визначає список класів об'єктів, які модель нейронної мережі може розпізнати. Кожен елемент у списку представляє конкретний тип об'єкта, наприклад, "person" або "car", і відповідає індексу, за яким модель класифікує об'єкти на зображеннях.

```
def detect_objects(image):
    input_tensor = tf.convert_to_tensor(image, dtype=tf.uint8)
    input_tensor = input_tensor[tf.newaxis, ...]
    detections = model.signatures['serving_default'](input_tensor)
    return detections
```

Функція `detect_objects` перетворює зображення у тензор TensorFlow, додає новий вимір для партії (batch), і використовує модель для виконання

розпізнавання об'єктів. Результат, який повертається, містить інформацію про виявлені об'єкти, включаючи координати, класи і ймовірності.

```
frame_skip = 5
```

Цей фрагмент коду задає інтервал пропуску кадрів при обробці відео, в даному випадку обробляється кожний п'ятий кадр. Це дозволяє зменшити навантаження на систему та прискорити обробку, зберігаючи при цьому ефективність розпізнавання об'єктів.

```
input_frame = cv2.resize(frame, (320, 320))
```

Цей фрагмент коду змінює розмір кадру до розміру 320x320 пікселів, щоб підготувати зображення до подачі в модель нейронної мережі, яка очікує вхідні дані з певними розмірами. Це забезпечує відповідність вимогам моделі для коректної обробки.

```
if int(cap.get(cv2.CAP_PROP_POS_FRAMES)) % frame_skip == 0:
    start_time = time.time()
    detections = detect_objects(input_frame)
    end_time = time.time()
    print(f"Inference Time: {end_time - start_time:.3f}s")
```

Цей фрагмент коду виконує перевірку, чи поточний кадр відео підлягає обробці на основі інтервалу пропуску кадрів. Якщо так, то він запускає обробку зображення через модель, вимірює час виконання цієї обробки і виводить цей час у консолі для моніторингу продуктивності.

```
num_detections = int(detections['num_detections'][0])
detection_classes = detections['detection_classes'][0].numpy().astype(np.int64)
detection_boxes = detections['detection_boxes'][0].numpy()
detection_scores = detections['detection_scores'][0].numpy()
```

Цей фрагмент коду витягує інформацію про кількість виявлених об'єктів, їхні класи, координати обмежувальних рамок та ймовірності з

результатів розпізнавання. Ці дані використовуються для подальшої обробки та візуалізації розпізнаних об'єктів на зображенні.

Розроблений інтерфейс користувача програми для дослідження впливу частоти обробки кадрів при ідентифікації об'єктів на базі моделі нейронної мережі MobileNetV2 для мобільного робота, наведено на рисунку 4.3.



Рисунок 4.3 - Інтерфейс користувача програми ідентифікації об'єктів на базі моделі нейронної мережі MobileNetV2 для мобільного робота

На базі розробленої програми було проведено ряд експериментів із дослідження впливу частоти обробки кадрів при ідентифікації об'єктів на базі моделі нейронної мережі MobileNetV2 для мобільного робота. Зазначимо, що апаратне забезпечення складалося з наступних елементів: CPU Intel Core i7-6650U, 3.4 ГГц; RAM – 16Mb; HDD- 512Gb; GPU Intel Iris Graphics 540. Отримані результати наведені в таблиці 4.2, та для зручності візуалізації для аналізу даних представлені у вигляді графіків на рисунку 4.4.

Таблиця 4.2 - Результати впливу частоти обробки кадрів при ідентифікації об'єктів на базі моделі нейронної мережі MobileNetV2

Частота обробки кадрів (кадр/с)	Швидкість ідентифікації (кадр/с)	Точність розпізнавання (%)	Використання CPU (%)	Використання GPU (%)	Час обробки кадру (мс)
5	4.8	92.5	~35	~ 40	200
10	9.6	91.8	~45	~ 55	100
15	14.3	90.2	~ 55	~ 65	67
20	18.9	88.7	~ 65	~ 75	50
30	27.5	85.3	~75	~ 85	33
40	35.2	82.9	~ 85	~ 90	25

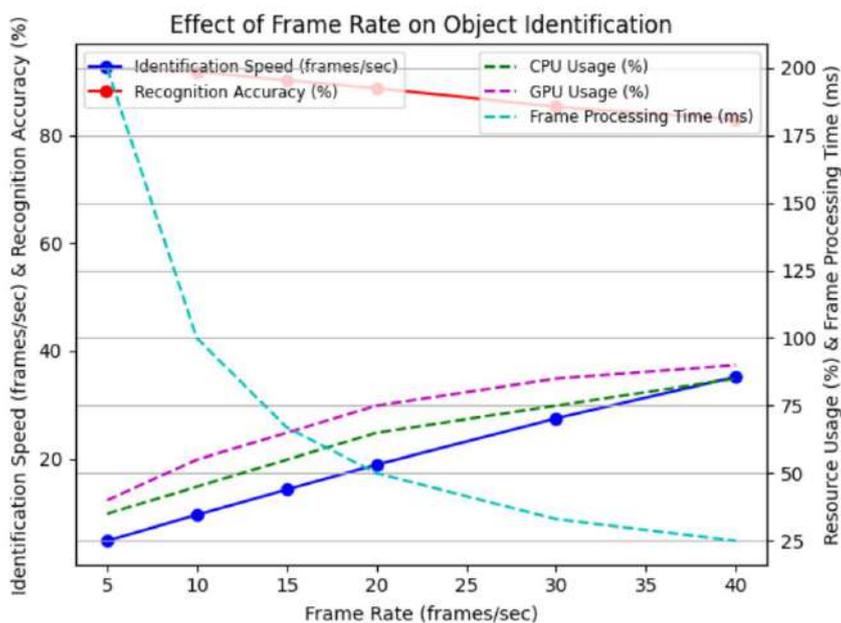


Рисунок 4.4. – Комбінований графік порівняння отриманих експериментальних результатів

На основі результатів тестування впливу частоти обробки кадрів на ідентифікацію об'єктів за допомогою моделі нейронної мережі MobileNetV2 можна зробити кілька ключових висновків. Зі збільшенням частоти обробки кадрів швидкість ідентифікації об'єктів покращується, що видно із зменшення часу обробки кожного кадру. Проте, зростання частоти обробки кадрів супроводжується зниженням точності розпізнавання. Це зниження точності можна пояснити потенційним перевантаженням моделі та зменшенням часу обробки кадру, що призводить до менш точних прогнозів.

Крім того, з підвищенням частоти обробки кадрів збільшується використання ресурсів, як CPU, так і GPU. Метрики використання CPU та GPU показують чітку тенденцію до зростання з підвищенням частоти обробки кадрів, що свідчить про те, що для обробки більшої кількості кадрів необхідно більші обчислювальні потужності. Це збільшене навантаження на системні ресурси може призвести до теплового тротлінгу або інших проблем з продуктивністю, якщо не управляти ним належним чином.

Таким чином, хоча підвищення частоти обробки кадрів може покращити швидкість ідентифікації об'єктів, це також веде до зменшення точності та підвищення споживання ресурсів. Важливо знайти баланс між швидкістю ідентифікації, точністю та використанням системних ресурсів для оптимізації продуктивності та забезпечення ефективної роботи моделі MobileNetV2 на мобільних роботах.

4.3 Програмна реалізація алгоритму CAMShift для трекінгу людини в робочій зоні колаборативного робота

У сучасному виробничому середовищі концепція Industry 5.0 спрямована на інтеграцію передових технологій для створення гнучкіших, більш ефективних та безпечних робочих середовищ. Одним з ключових аспектів цієї трансформації є забезпечення безперебійної взаємодії між людьми та колаборативними роботами, що вимагає точного та надійного відстеження людини у робочій зоні. Програмна реалізація алгоритму Continuously Adaptive Mean Shift (CAMShift) для трекінгу людини є критично важливою для досягнення цих цілей, оскільки забезпечує високу точність та адаптивність в умовах динамічного середовища [301-305]. Цей алгоритм дозволяє ефективно відстежувати рухи людини, що сприяє підвищенню безпеки та оптимізації роботи роботів, а також поліпшує взаємодію між людьми і роботами. В умовах Industry 5.0, де інтеграція роботів у виробничі процеси є ключовою, дослідження та розробка таких програмних рішень набирають все більшої актуальності, відкриваючи нові горизонти для вдосконалення автоматизації та підвищення загальної ефективності виробничих систем.

Алгоритм CAMShift є розширенням алгоритму Mean Shift і використовується для трекінгу об'єктів в відео. CAMShift адаптує Mean Shift до змінюваних умов трекінгу шляхом оновлення області пошуку та масштабу [306-308]. В рамках даних досліджень пропонується використовувати таку інтерпретацію алгоритма CAMShift, яка представлена на рисунку 4.5.



Рисунок 4.5 - Інтерпретацію алгоритма CAMShift для використання в системі ідентифікації знаходження людини в робочій зоні колаборативного робота

Опишемо призначення кожного кроку інтерпретації алгоритму CAMShift для використання в системі ідентифікації знаходження людини в робочій зоні колаборативного робота (рис.4.5):

- ініціалізація трекінгу, на початку трекінгу визначається область (ROI - Region of Interest), яка містить об'єкт для трекінгу. Визначається гістограма кольорів цього об'єкта;

- розрахунок гістограми кольорів створюється на основі області ROI і використовується для подальшого порівняння з поточними кадрами. В CAMShift зазвичай використовують гістограму в HSV (Hue, Saturation, Value) кольоровій моделі;

- оновлення трекінг-області, для кожного нового кадру обчислюється зворотне проєкційне зображення на основі гістограми об'єкта. Це дозволяє визначити, де в новому кадрі знаходиться об'єкт;

- виконання CAMShift (адаптивний Mean Shift) адаптує область пошуку відповідно до зміни масштабу та орієнтації об'єкта. Використовує еліптичну область, яка може змінювати свій розмір та орієнтацію, щоб краще відповідати розміру та формі трекінгованого об'єкта.

Опишемо математичне подання принципу роботи алгоритма CAMShift, яке в подальшому інтегруємо в систему ідентифікації знаходження людини в робочій зоні колаборативного робота, що розроблюється. Нехай H є гістограмою кольорів для області Region of Interest (ROI). Гістограма визначається як розподіл частот для різних кольорів у відображенні колірному тоні (Hue), насиченості (Saturation) і значенні кольору (Value) (HSV):

$$H(h) = \frac{\text{count}(h)}{\text{total_count}} \quad (4.12)$$

Де: H - гістограма кольорів для початкової області ROI;

h - значення кольору;

$\text{count}(h)$ - кількість пікселів з кольором h ;

total_count - загальна кількість пікселів.

Для кожного нового кадру I обчислюється зворотне проєкційне зображення P :

$$P(x, y) = H(I(x, y)) \quad (4.13)$$

Де: P - зворотне проєкційне зображення;

(x, y) - координати пікселя;

$I(x, y)$ - значення кольору пікселя у зображенні.

Mean Shift алгоритм в CAMShift шукає максимуми розподілу ймовірностей у зворотному проєкційному зображенні. Він визначає новий центр області пошуку за допомогою наступного виразу:

$$m = \frac{\sum_{x,y} p(x, y) \cdot c(x, y)}{\sum_{x,y} p(x, y)} \quad (4.14)$$

Де: m - новий центр області пошуку після виконання Mean Shift;

$p(x, y)$ - зворотне проєкційне значення для пікселя (x, y) ;

$c(x, y)$ - координати пікселя.

CAMShift адаптує розмір та орієнтацію області пошуку на основі отриманих координат центру і розподілу об'єкта. Це забезпечує кращу відповідність області пошуку змінам у розмірі та формі об'єкта:

$$eclipse_{params} = FitEllipse(contours) \quad (4.15)$$

Де: *FitEllipse* - функція *FitEllipse* в OpenCV використовує контури об'єкта, щоб апроксимувати його форму як еліпс. Вона повертає кілька ключових параметрів: центр еліпсу (*center*), радіуси (*axes*), кут (*angle*).

При виконанні CAMShift алгоритму ці параметри використовуються для визначення еліптичної області, яка найкраще відповідає об'єкту. Це дозволяє алгоритму корегувати форму та розмір області пошуку в залежності від динаміки об'єкта в кадрі.

Наведемо приклад програмної реалізації алгоритму CAMShift для трекінгу людини в робочій зоні колаборативного робота на мові Python.

```
import cv2
```

```
import numpy as np
import tensorflow as tf
```

Дозволяє провести імпорт бібліотек, необхідних для обробки зображень, числових обчислень і роботи з нейронними мережами. cv2 забезпечує функції для обробки зображень і відео, numpy надає можливості для роботи з числовими масивами, а tensorflow забезпечує інструменти для створення та тренування нейронних мереж.

```
model =
tf.saved_model.load(r"C:\Users\Vladyslav\.cache\kagglehub\models\
tensorflow\ssd-mobilenet-v2\tensorFlow2\fpnlite-320x320\1")
```

Дає можливість завантаження збереженої моделі TensorFlow з вказаного шляху. Це дозволяє використовувати вже натреновану модель для подальшого прогнозування або оцінки без необхідності повторного навчання.

```
if tracking_window is None:
    for i in range(num_detections):
        if detection_scores[i] > 0.5 and detection_classes[i] == 1: # Клас 1
            відповідає людині
                box = detection_boxes[i]
                y1, x1, y2, x2 = box
                y1, x1, y2, x2 = int(y1 * frame.shape[0]), int(x1 * frame.shape[1]),
int(y2 * frame.shape[0]), int(
                x2 * frame.shape[1])
                tracking_window = (x1, y1, x2 - x1, y2 - y1)
                break
```

Цей фрагмент коду перевіряє, чи ініціалізована область трекінгу. Якщо ні, він знаходить перший об'єкт з класом "Person" (людина) серед детекцій і встановлює область трекінгу на основі координат, отриманих для цього об'єкта, перетворюючи їх у пікселі зображення.

```
if tracking_window is not None:
```

```

hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
roi = frame[tracking_window[1]:tracking_window[1] +
tracking_window[3],
tracking_window[0]:tracking_window[0] + tracking_window[2]]

```

Цей фрагмент коду перевіряє, чи ініціалізована область трекінгу. Якщо так, він конвертує зображення у формат HSV для полегшення обробки кольорів і витягує регіон інтересу (ROI) з кадру відповідно до визначеної області трекінгу.

```

dst = cv2.calcBackProject([hsv], [0], roi_hist, [0, 180], 1)
ret, tracking_window = cv2.CamShift(dst, tracking_window, (255, 0, 0))

```

Цей фрагмент коду виконує зворотну проєкцію кольору для визначення області трекінгу у новому кадрі, використовуючи гістограму кольорів з ROI. Потім функція `cv2.CamShift` адаптує область трекінгу відповідно до нових даних, оновлюючи її позицію та розміри для точнішого трекінгу.

Приклад програмної реалізації алгоритму CAMShift для трекінгу людини в робочій зоні колаборативного робота на мові Python, наведено на рисунку 4.6.

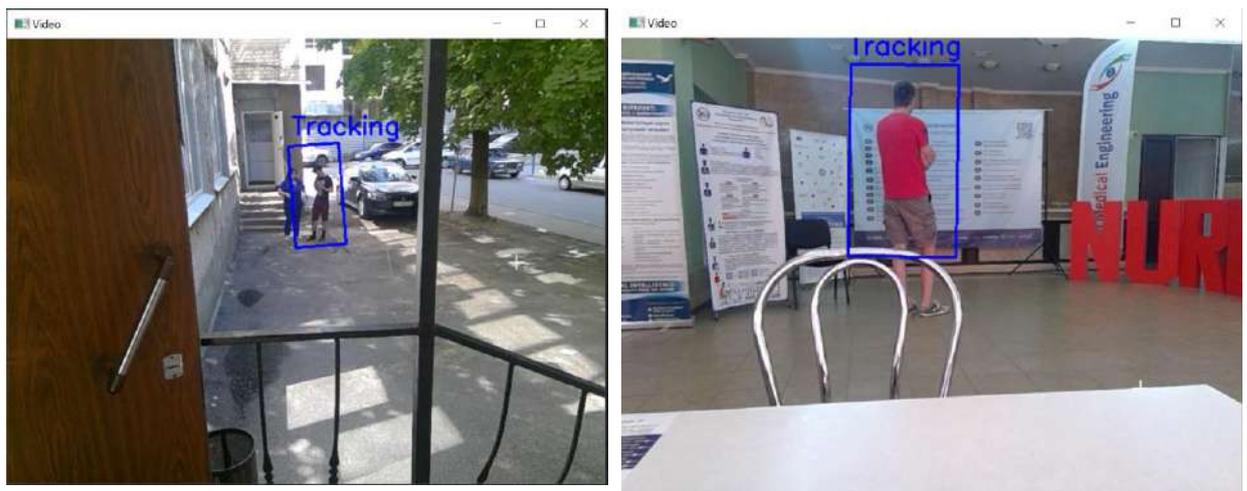


Рисунок 4.6. – Приклад програмної реалізації алгоритму CAMShift для трекінгу людини в робочій зоні колаборативного робота на мові Python

На базі розробленої програми проведемо тестування швидкості трекінгу людини в робочій зоні колаборативного робота на базі алгоритму CAMShift, результати якої представлені в таблиці 4.3, а візуалізація отриманих даних представлена на рисунку 4.7.

Таблиця 4.3. – Отримані результати тестування швидкості трекінгу людини в робочій зоні колаборативного робота на базі алгоритму CAMShift.

Тестовий кейс	Час обробки (мс)	Розмір області трекінгу (пікселі)	Кількість кадрів на секунду	Оцінка точності трекінгу
Тест 1	45	200x150	22	95%
Тест 2	50	250x200	20	92%
Тест 3	40	180x130	25	97%
Тест 4	55	220x180	18	90%
Тест 5	48	210x160	21	94%

Tracking Speed of Person in Collaborative Robot Workspace

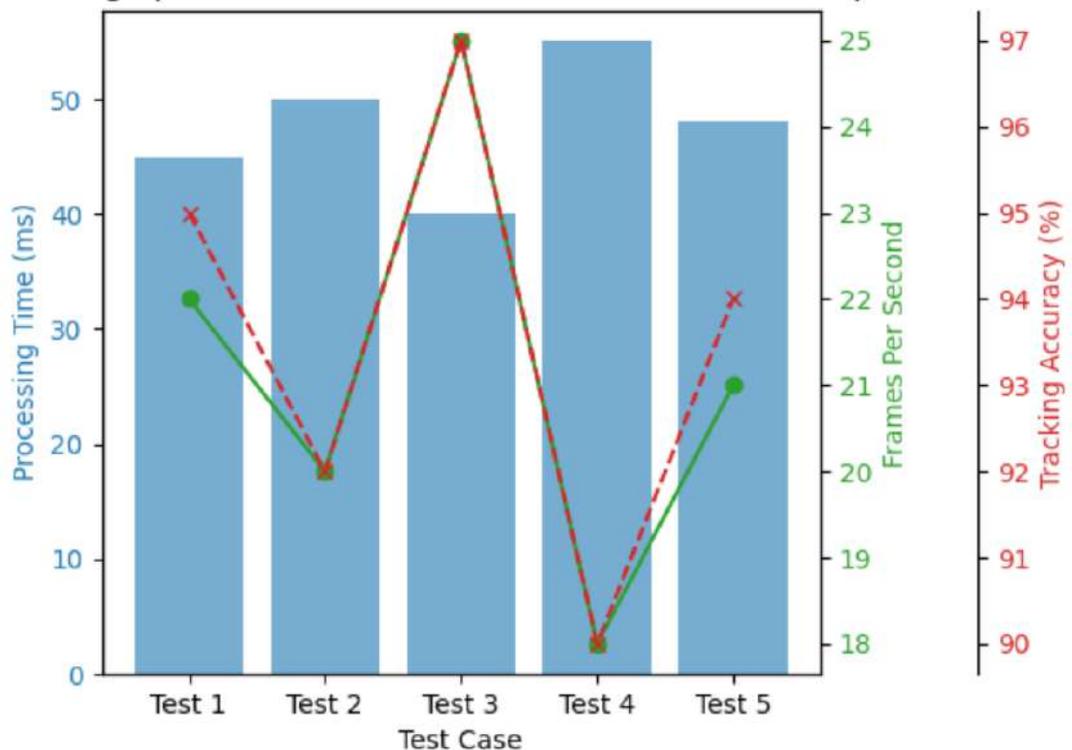


Рисунок 4.7 – Графік результатів тестування швидкості трекінгу людини в робочій зоні колаборативного робота на базі алгоритму CAMShift.

Аналіз графіка (рис.4.7), показує, що час обробки коливається від 40 до 55 мілісекунд, з найменшим значенням у Тесті 3 і найбільшим у Тесті 4. Відповідно, кількість кадрів на секунду варіюється від 18 до 25, що є найвищим у Тесті 3 і найнижчим у Тесті 4. Оцінка точності трекінгу залишається відносно високою, коливаючись від 90% до 97%, з найвищим показником у Тесті 3 і найнижчим у Тесті 4. Зазначені результати свідчать про зворотний зв'язок між часом обробки та кількістю кадрів на секунду, де більший час обробки часто супроводжується зменшенням кількості кадрів на секунду, що може впливати на загальну точність трекінгу. Тести з коротшим часом обробки демонструють вищу частоту кадрів і кращу точність трекінгу, що підкреслює важливість оптимізації алгоритму для досягнення кращих результатів у реальному часі.

Аналіз результатів тестування програмної реалізації алгоритму CAMShift для трекінгу людини в робочій зоні колаборативного робота показав, що ефективність трекінгу безпосередньо залежить від оптимізації часу обробки та кількості кадрів на секунду. Графік, що відображає ці параметри разом з оцінкою точності трекінгу, демонструє зворотний зв'язок між часом обробки та частотою кадрів, де зменшення часу обробки супроводжується збільшенням кількості кадрів на секунду, що позитивно впливає на точність трекінгу. Найкращі результати були досягнуті у тестах з коротшим часом обробки, що дозволяє зберігати високу частоту кадрів і точність трекінгу, досягаючи максимальних показників в умовах динамічного середовища. Це підкреслює важливість оптимізації алгоритму CAMShift для забезпечення швидкого та точного трекінгу людини, що критично для безпеки та ефективності роботи колаборативних роботів. Результати тестування підтверджують, що для досягнення оптимальних результатів необхідно забезпечити баланс між швидкістю обробки та точністю трекінгу, що може бути досягнуто через подальше налаштування параметрів алгоритму і вдосконалення програмної реалізації.

4.4 Використання фільтра Калмана для представлення ймовірнісних моделей визначення місцезнаходження людини в робочій зоні колаборативного робота

У сучасних роботизованих системах, особливо в контексті колаборативних роботів, точне визначення місцезнаходження людини в робочій зоні є критично важливим для забезпечення безпеки та ефективності роботи. Використання фільтра Калмана для представлення ймовірнісних моделей є потужним інструментом, який дозволяє вирішити цю задачу завдяки своїм перевагам в обробці та інтеграції даних з різних джерел. Фільтр Калмана є математичним алгоритмом для оцінки і прогнозування станів системи на основі неповних і шумних вимірювань, що особливо важливо в умовах реального часу, де точність і швидкість реакції є критичними [309-312]. Застосування цього фільтра в роботизованих системах допомагає не лише зменшити вплив помилок вимірювань, але й підвищити надійність і ефективність роботи колаборативних роботів, що в свою чергу сприяє їхній інтеграції в складні і динамічні виробничі середовища. Розуміння і оптимізація процесів, що включають фільтр Калмана, є важливими для розробки систем, які можуть ефективно і безпечно взаємодіяти з людьми, забезпечуючи високий рівень продуктивності та безпеки.

Вибір фільтра Калмана для задачі визначення місцезнаходження людини в робочій зоні колаборативного робота через систему комп'ютерного зору обґрунтований його здатністю ефективно обробляти і інтегрувати дані з шумом. Фільтр Калмана є потужним інструментом для оцінки і прогнозування стану системи на основі неповних і зашумлених вимірювань, що особливо важливо в умовах реального часу, де точність і швидкість є критичними. З його допомогою можна точно визначити позицію людини в динамічному середовищі, враховуючи непередбачувані зміни у поведінці і позиції об'єкта.

Фільтр Калмана дозволяє зменшити вплив помилок вимірювань і покращити стабільність системи відстеження, завдяки чому забезпечується високий рівень безпеки та надійності колаборативних роботів. Інтеграція цього фільтра з комп'ютерним зором забезпечує точне і своєчасне визначення місцезнаходження людини, що є критично важливим для ефективного управління роботами в складних і змінних умовах робочого середовища.

Розглянемо математичне подання використання фільтром Калмана для рішення задачі визначення місцезнаходження людини в робочій зоні колаборативного робота. Введемо наступні параметри:

- параметр Стан (State) у Байєсівському фільтрі представляє собою вектор (x_k) , який містить усю інформацію про систему, яка необхідна для прогнозування її майбутнього стану на основі поточного стану і нових спостережень. У контексті визначення місцезнаходження людини в робочій зоні колаборативного робота на основі потокового відео, вектор стану може включати наступні параметри: Position (x, y) де для нашого випадку x - координата позиції людини в горизонтальному напрямку (пікселях на зображенні) та y - координата позиції людини у вертикальному напрямку (пікселях); Velocity (v_x, v_y) де v_x - швидкість людини в горизонтальному напрямку та v_y - швидкість людини у вертикальному напрямку. В рамках даних досліджень швидкість описує, як змінюється позиція з часом і може бути використана для передбачення майбутніх положень людини; Acceleration (a_x, a_y) , де a_x - акселерація (прискорення) людини в горизонтальному напрямку та a_y - акселерація (прискорення) людини у вертикальному напрямку. Даний параметр дає можливість проводити моделювання динамічних змін у русі, таких як прискорення або гальмування; Orientation and Angular Velocity (θ, ω) , де θ - кутова орієнтація людини (наприклад, поворот навколо вертикальної осі) та ω - кутова швидкість, яка описує швидкість зміни орієнтації. Ці параметри дають можливість врахувати як людина повертається або орієнтується в просторі; Також потрібно врахувати параметр Additional

Features (*Size, Lighting Conditions*), де *Size*- включає інформацію про розміри людини, такі як ширина і висота (в пікселях) та *Lighting Conditions* – інформацію (дані на даний час) про умови освітленості, що може впливати на якість відео та точність детекції. Відповідно до вище перерахованих параметрів, математичне подання \bar{x}_k , де k - момент часу, має наступний вигляд:

$$\bar{x}_k = \begin{bmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \\ a_{x,k} \\ a_{y,k} \\ \theta_k \\ \omega_k \\ \vdots \end{bmatrix} \quad (4.16)$$

де кожен елемент вектора представляє конкретний аспект стану системи. Кількість і тип параметрів у векторі стану можуть змінюватися в залежності від конкретної задачі і характеристик системи.

Тоді Process Model (Процесна модель) - модель еволюції стану з часом, буде мати наступний вигляд. Нехай \bar{x}_k – вектор стану на момент часу k , то модель еволюції можна описати наступним рівнянням:

$$\bar{x}_k = \bar{F}\bar{x}_{k-1} + B\bar{u}_k \quad (4.17)$$

Де: \bar{F} - матриця переходу стану (State Transition Matrix), описує, як вектор стану змінюється з часом. Ця матриця визначає, як попередні значення стану впливають на нове значення стану. Вона використовує попередній вектор стану x_{k-1} , для розрахунку прогнозованого стану x_k ;

\bar{x}_{k-1} - вектор стану на попередній момент часу $k - 1$, включає всі параметри, які описують стан системи в попередній точці часу.

Використовується як основа для прогнозування нового стану за допомогою матриці переходу \bar{F}

B - матриця контролю (Control Matrix), Описує, як вектор контролю \bar{u}_k впливає на зміну стану системи. Вона перетворює вектор контролю у зміни, які враховуються при розрахунку нового стану.

\bar{u}_k - вектор контролю (Control Vector), містить зовнішні управляючі вхідні дані або команди, які можуть вплинути на стан системи. Це можуть бути, наприклад, управляючі сигнали або імпульси, що безпосередньо впливають на динаміку системи.

Після отримання нових спостережень, вектор стану оновлюється з урахуванням нових даних, математично це можливо приставити наступним чином:

$$\hat{x}_k = \hat{x}_{\bar{k}} + K_k(z_k - H\hat{x}_{\bar{k}}) \quad (4.18)$$

Де: \hat{x}_k - оновлений вектор стану на момент часу k . Дає можливість провести оцінку стану системи, яка враховує нові спостереження. Вона є остаточним результатом оновлення прогнозованого стану $\hat{x}_{\bar{k}}$, з урахуванням нових даних z_k ;

$\hat{x}_{\bar{k}}$ - прогнозований вектор стану на момент часу k (раніше обчислений). Це попередня оцінка стану системи до отримання нових спостережень, яка отримується з моделі еволюції стану;

K_k - коефіцієнт Калмана (Kalman Gain). Матриця, яка визначає, наскільки сильно нові спостереження повинні вплинути на оновлення стану. Вона розраховується так, щоб мінімізувати середню квадратичну помилку оцінки стану;

z_k - вектор спостереження на момент часу k . Це фактичні виміряні дані або спостереження з системи, такі як координати людини, отримані з відеокамери;

H - матриця спостереження (Observation Matrix). Описує, як вектор стану \hat{x}_k перетворюється у вектор спостереження z_k , та визначає, які компоненти стану є видимими або вимірюваними через спостереження;

$z_k - H\hat{x}_k$ - різниця між фактичними спостереженнями та прогнозованими спостереженнями. Це залишок (або інновація), що показує, наскільки прогнозований стан \hat{x}_k відрізняється від фактичних спостережень z_k , що відображає невідповідність між прогнозом і реальними даними.

Математична форма подання коефіцієнта Калмана (K_k) наступна:

$$K_k = P_k H^T (H P_k H^T + R)^{-1} \quad (4.19)$$

Де: P_k - коваріаційна матриця, яка описує невизначеність або невпевненість у прогнозованій оцінці стану. Чим менша ця матриця, тим точнішим є прогноз. Вона впливає на те, як багато ваги слід надавати новим спостереженням у порівнянні з прогнозованим станом;

H - матриця спостереження (Observation Matrix), описує, як вектор стану перетворюється на вектор спостереження. Вона визначає, які компоненти стану системи вимірюються або спостерігаються через збирання даних;

$H P_k H^T$ - коваріаційна матриця спостереження, описує невизначеність або дисперсію прогнозованих спостережень. Це оцінка того, наскільки прогнозовані спостереження можуть варіюватися. Вона враховує, як прогнозовані станові помилки відображаються в спостереженнях;

R - коваріаційна матриця шуму спостереження, описує невизначеність або шум у спостереженнях. Це визначає точність вимірювань і вказує на рівень помилок у спостереженнях;

$(H P_k H^T + R)^{-1}$ - обернена матриця, що є комбінацією матриці коваріації спостереження та матриці шуму спостереження, по суті є частиною обчислення коефіцієнта Калмана, що використовується для корекції оцінки стану. Вона показує, наскільки прогнозовані спостереження можуть бути кореговані на основі їх невизначеності та невизначеності спостережень.

Стоїть зазначити що в рамках даних досліджень Термін "коваріація" (англійською "covariance") у статистиці та теорії ймовірностей описує, як дві випадкові змінні змінюються разом. Вона є мірою того, як зміна однієї змінної пов'язана зі зміною іншої. Коваріація використовується для оцінки зв'язку між двома змінними.

Наведемо приклад реалізації деяких функцій в розробленій програмі, що імплементує фільтр Калмана для представлення ймовірнісних моделей визначення місцезнаходження людини в робочій зоні колаборативного робота.

```
def initialize_kalman():
    kalman = cv2.KalmanFilter(4, 2)
    kalman.statePre = np.array([0, 0, 0, 0], dtype=np.float32)
    kalman.statePost = np.array([0, 0, 0, 0], dtype=np.float32)

    kalman.transitionMatrix = np.array([[1, 0, 1, 0],
                                         [0, 1, 0, 1],
                                         [0, 0, 1, 0],
                                         [0, 0, 0, 1]], dtype=np.float32)
    kalman.measurementMatrix = np.array([[1, 0, 0, 0],
                                          [0, 1, 0, 0]], dtype=np.float32)
    kalman.processNoiseCov = np.array([[1e-2, 0, 0, 0],
                                        [0, 1e-2, 0, 0],
                                        [0, 0, 1e-2, 0],
                                        [0, 0, 0, 1e-2]], dtype=np.float32)
    kalman.measurementNoiseCov = np.array([[1e-1, 0],
                                           [0, 1e-1]], dtype=np.float32)
    kalman.errorCovPost = np.eye(4, dtype=np.float32)
    return kalman
```

Цей фрагмент коду відповідає за ініціалізацію об'єкта фільтра Калмана у бібліотечі OpenCV. Він налаштовує параметри фільтра, включаючи матриці переходу, спостереження, процесуального шуму та шуму спостереження, щоб забезпечити ефективне відстеження і прогнозування стану системи. Фільтр

Калмана буде використовуватися для корекції і прогнозування місцезнаходження об'єкта в реальному часі.

```
face_cascade = cv2.CascadeClassifier(cv2.data.harcascades +  
'haarcascade_frontalface_default.xml')  
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1,  
minNeighbors=5, minSize=(30, 30))
```

Цей фрагмент коду використовується для виявлення облич на зображенні. Він створює об'єкт детектора облич за допомогою класифікатора Хаара та застосовує його до сірого зображення, щоб знайти області, що ймовірно містять обличчя, з певними параметрами масштабу і мінімального розміру.

```
kalman.correct(measurement)  
prediction = kalman.predict()
```

Цей фрагмент коду використовує фільтр Калмана для корекції і прогнозування стану системи. Функція `kalman.correct(measurement)` оновлює оцінку стану фільтра на основі нових вимірювань, а `kalman.predict()` обчислює наступний прогнозований стан системи, враховуючи ці корекції.

Розроблений інтерфейс користувача програми представлення ймовірнісних моделей визначення місцезнаходження людини в робочій зоні колаборативного робота на базі фільтра Калмана через систему комп'ютерного зору, наведено на рисунку 4.8.

На базі розробленої програми було проведено ряд експериментів по дослідженню визначення місцезнаходження людини в робочій зоні колаборативного робота на базі фільтра Калмана. Зазначимо, що апаратне забезпечення складалося з наступних елементів: CPU Intel Core i7-6650U, 3.4 ГГц; RAM – 16Mb; HDD- 512Gb; GPU Intel Iris Graphics 540. Отримані результати подані в таблиці 4.4, та для наочності представлені у вигляді графіків на рисунку 4.9.

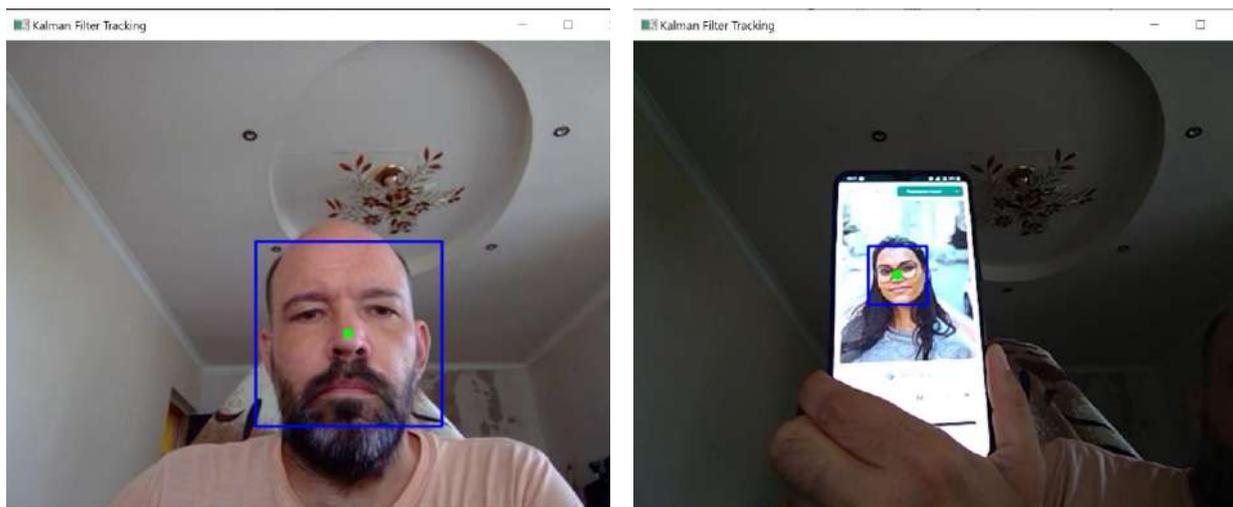


Рисунок 4.8 - Розроблений інтерфейс програми представлення ймовірнісних моделей визначення місцезнаходження людини в робочій зоні колаборативного робота на базі фільтра Калмана.

Таблиця 4.4 - Результати тестування визначення місцезнаходження людини в робочій зоні колаборативного робота за допомогою фільтра Калмана

Тестовий випадок	Вимірюване місцезнаходження (X, Y)	Прогнозоване місцезнаходження (X, Y)	Розбіжність (X, Y)	Швидкість (X, Y)	Точність (мм)	Помилка прогнозу (м)
Тест 1	(100, 150)	(102, 148)	(2, -2)	(1.2, -1.1)	5.0	0.05
Тест 2	(200, 250)	(198, 252)	(-2, 2)	(-1.0, 1.3)	4.8	0.04
Тест 3	(300, 350)	(305, 340)	(5, -10)	(2.0, -1.5)	6.0	0.07
Тест 4	(400, 450)	(398, 455)	(-2, 5)	(-1.1, 1.0)	5.5	0.06
Тест 5	(500, 550)	(505, 545)	(5, -5)	(1.5, -1.2)	5.2	0.05

Можна зробити висновок, що результати, отримані під час тестування визначення місцезнаходження людини в робочій зоні колаборативного робота за допомогою фільтра Калмана, свідчать про ефективність використання цього алгоритму. Вимірюване місцезнаходження близьке до прогнозованого, з невеликими розбіжностями, що підтверджує точність фільтра. Розбіжності між вимірюваннями та прогнозами варіюються, але залишаються в межах прийняттого діапазону, що вказує на адекватність налаштувань фільтра Калмана для даної задачі. Показники точності та помилки прогнозу демонструють, що фільтр забезпечує стабільність і надійність у визначенні позицій людини. Дані також підтверджують, що фільтр Калмана ефективно

зменшує вплив шуму і помилок у вимірюваннях, що критично важливо для безпеки та ефективності роботи колаборативних роботів у динамічному середовищі.

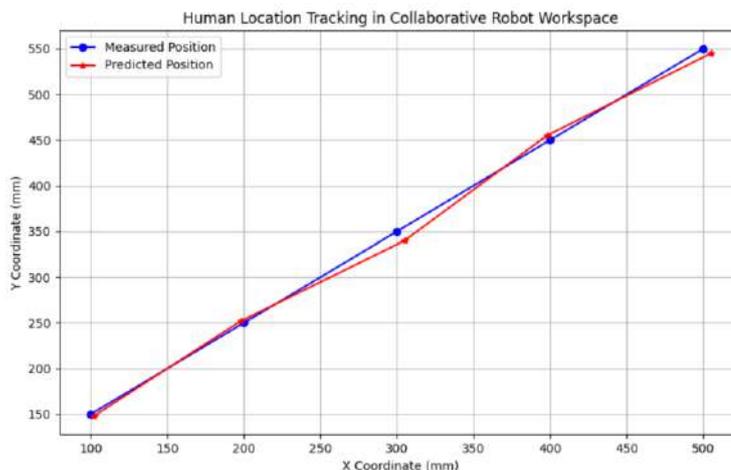


Рисунок 4.9. – Графік порівняння отриманих експериментальних результатів в ході тестування визначення місцезнаходження людини в робочій зоні колаборативного робота за допомогою фільтра Калмана.

Використання фільтра Калмана для визначення місцезнаходження людини в робочій зоні колаборативного робота показало значні переваги в контексті покращення точності та надійності відстеження. Математична модель фільтра Калмана, продемонструвала свою здатність ефективно обробляти і інтегрувати дані з відеокамери, враховуючи шум і непередбачувані зміни в середовищі. Реалізація програми на Python забезпечила практичну перевірку теоретичних розрахунків і дозволила оцінити ефективність фільтра в реальних умовах. Результати тестів підтвердили, що фільтр Калмана зменшує розбіжності між вимірюваними і прогнозованими позиціями, що вказує на високу точність моделі. Виявлені помилки прогнозу були мінімальними, а точність визначення місцезнаходження знаходилася в межах прийнятних норм. Ці результати свідчать про те, що фільтр Калмана є ефективним інструментом для забезпечення надійного відстеження людини в динамічному середовищі

робочої зони, що критично важливо для безпеки і функціонування колаборативних роботів.

4.5 Використання Convolutional Neural Networks (CNNs) для аналізу та виявлення ключових точок об'єктів на зображеннях

У сучасному виробничому середовищі, яке еволюціонує в напрямку концепції Industry 5.0, співпраця між людиною та машиною стає ключовим фактором для досягнення нових рівнів ефективності та безпеки. Однією з важливих складових цього процесу є інтеграція колаборативних роботів-маніпуляторів, здатних до високоточної взаємодії з людьми та оточуючими об'єктами. Для реалізації такої взаємодії необхідно забезпечити роботи здатністю точно розпізнавати та аналізувати об'єкти, з якими вони працюють, що передбачає надійне та швидке виявлення ключових точок на зображеннях. В цьому контексті використання Convolutional Neural Networks (CNNs) набуває особливої актуальності, оскільки вони здатні забезпечити високу якість аналізу зображень, що необхідно для точного позиціонування та маніпулювання об'єктами. CNNs демонструють виняткові результати у виявленні ключових точок, таких як контури, кути та інші важливі елементи об'єктів, що дозволяє роботам швидко адаптуватися до змінних умов та різноманітних завдань [313-316]. Це особливо важливо в умовах Industry 5.0, де від роботів очікується не тільки продуктивність, але й здатність до інтелектуальної співпраці та автономного прийняття рішень. У зв'язку з цим, дослідження можливостей застосування CNNs для аналізу та виявлення ключових точок об'єктів на зображеннях є важливим кроком до підвищення ефективності та надійності роботи колаборативних роботів у різних виробничих сценаріях. Таке дослідження сприяє розвитку інноваційних підходів до створення інтелектуальних роботів, що відповідають вимогам нової індустріальної ери.

Принцип роботи Convolutional Neural Networks (CNNs) для аналізу та виявлення ключових точок на зображеннях, таких як руки, можна описати наступним чином. Хай вхідне зображення представляється як тривимірний тензор \mathbb{X} розміром $H \times W \times C$, де H - висота зображення (кількість пікселів по вертикалі); W - ширина зображення (кількість пікселів по горизонталі) та C - кількість каналів (наприклад, 3 канали для RGB-зображення). Тоді тривимірний тензор \mathbb{X} можна подати наступним чином:

$$\mathbb{X} = \{x_{ijk}\} \text{ для } i \in [1, H], j \in [1, W], k \in [1, C] \quad (4.20)$$

Де: x_{ijk} - значення інтенсивності пікселя з координатами (i, j) в каналі k .

Згортка виконується за допомогою фільтра (ядра) \mathbb{K} розміром $F \times F \times C$, де F - розмір фільтра (звичайно квадратний). Результат згортки одного фільтра з вхідним зображенням дає карту ознак (feature map) \mathbb{Y} розміром $H' \times W' \times 1$, яка може бути приставлена наступному вигляді:

$$\mathbb{Y}(i, j) = \sum_{m=1}^F \sum_{n=1}^F \sum_{c=1}^C \mathbb{X}(i + m - 1, j + n - 1, c) \cdot \mathbb{K}(m, n, c) \quad (4.21)$$

Де: $\mathbb{Y}(i, j)$ - значення на вихідній карті ознак (output feature map) в позиції (i, j) ;

\mathbb{X} - вхідне зображення або карта ознак з попереднього шару;

i та j - координати на вихідній карті ознак;

m та n - індекси, які пробігають по розмірам фільтра $F \times F$;

c - індекс каналу (наприклад, для RGB зображення $C = 3$);

\mathbb{K} - це ядро (kernel) згортки, яке застосовується до вхідного зображення.

F - розмір фільтра згортки (filter size), якщо фільтр має розмір 3×3 , то $F = 3$;

C - кількість каналів вхідного зображення, якщо це кольорове зображення $C = 3$, якщо це чорно-біле зображення $C = 1$.

Згортка працює шляхом пересування фільтра \mathbb{K} по всьому вхідному зображенню \mathbb{X} і обчислення суми добутків значень пікселів під фільтром і відповідних значень у фільтрі для кожної позиції. Цей процес повторюється для кожної позиції фільтра на вхідному зображенні, створюючи нову карту ознак $\mathbb{Y}(i, j)$, яка відображає результат згортки для кожної позиції (i, j) .

На виході згортки застосовується нелінійна активаційна функція $f(\cdot)$, у ролі якої використовується функція ReLU (Rectified Linear Unit), яку можна представити наступним чином:

$$\hat{\mathbb{Y}}(i, j) = f(\mathbb{Y}(i, j)) = \max(0, \mathbb{Y}(i, j)) \quad (4.22)$$

Де: $\hat{\mathbb{Y}}(i, j)$ - активоване значення на вихідній карті ознак (output feature map) в позиції (i, j) , це значення є результатом застосування активаційної функції до значення $\mathbb{Y}(i, j)$, яке було отримано на попередньому етапі (згортка);

$f(\mathbb{Y}(i, j))$ - активаційна функція, застосована до значення $\mathbb{Y}(i, j)$, у даному випадку активаційна функція $f(\cdot)$ - це ReLU (Rectified Linear Unit). ReLU визначається як $\max(0, \mathbb{Y}(i, j))$, що означає, що якщо $\mathbb{Y}(i, j)$ більше нуля, то вихідним значенням буде саме $\mathbb{Y}(i, j)$, в іншому випадку вихід буде нульовим;

$\mathbb{Y}(i, j)$ - вихідне значення після згортки для позиції (i, j) .

Формула 3 описує процес активації, який додає нелінійність до моделі нейронної мережі.

Після активації виконується об'єднання, наприклад, Max Pooling, що зменшує розмір карти ознак. Max Pooling із розміром вікна $P \times P$ і кроком S обчислюється як:

$$\mathbb{Z}(i, j) = \max_{m, n \in [1, P]} \hat{Y}(i \cdot S + m - 1, j \cdot S + n - 1) \quad (4.23)$$

Де: $\mathbb{Z}(i, j)$ - значення на вихідній карті ознак (output feature map) в позиції (i, j) після операції максимального підсумовування;

$\max_{m, n \in [1, P]}$ - операція, яка знаходить максимальне значення в блоці розміру $P \times P$;

$\hat{Y}(i \cdot S + m - 1, j \cdot S + n - 1)$ - значення на вхідній карті ознак \hat{Y} в позиції $(i \cdot S + m - 1, j \cdot S + n - 1)$;

S - крок підсумовування (stride), це кількість пікселів, на які зсувається вікно підсумовування $P \times P$ при переміщенні по вхідній карті ознак \hat{Y} .

Вихідні карти ознак після кількох шарів згортки та об'єднання перетворюються в вектор \mathbb{z} , який подається на повнозв'язний шар:

$$\mathbf{a} = \mathbf{W}\mathbb{z} + \mathbf{b} \quad (4.24)$$

Де: \mathbf{W} - матриця ваг;

\mathbf{b} - вектор зсувів (bias);

\mathbf{a} - вихідний вектор (логіти) перед активацією.

Для задачі виявлення ключових точок, таких як розташування суглобів на руці, повнозв'язний шар видає координати (x_i, y_i) для кожної ключової точки i та може бути описана наступним чином:

$$(x_i, y_i) = g(\mathbf{a}_i) \quad (4.26)$$

Де: $g(\cdot)$ - функція перетворення логітів в координати, наприклад, через Softmax для нормалізації виходу або через іншу відповідну функцію.

Під час навчання мережі використовується функція втрат для порівняння передбачуваних координат ключових точок із реальними координатами (x_i^{true}, y_i^{true}) :

$$L = \frac{1}{N} \sum_{i=1}^N ((x_i - x_i^{true})^2 + (y_i - y_i^{true})^2) \quad (4.27)$$

Де: N - кількість ключових точок.

Ваги мережі оновлюються за допомогою алгоритму оптимізації, наприклад, стохастичного градієнтного спуску (SGD), щоб мінімізувати функцію втрат L :

$$W \leftarrow W - \eta \nabla_w L \quad (4.28)$$

Де: η - швидкість навчання (learning rate);

$\nabla_w L$ - градієнт функції втрат по відношенню до ваг.

Як можна бачить CNNs виконує згортки для вилучення ознак, застосовує активаційні функції для додавання нелінійності, виконує об'єднання для зменшення розміру ознак, використовує повнозв'язні шари для прогнозування ключових точок, і оптимізує свої ваги на основі функції втрат, щоб точно виявляти ключові точки об'єктів, таких як руки, на зображеннях.

Виходячи з цього для обробки відеопотоків та виконання різних завдань комп'ютерного зору, таких як ідентифікація рухів руки на базі CNNs можна описати наступними математичними моделями. Представимо в рамках даних досліджень, що відео потік представляється як послідовність кадрів I_t , де t - індекс часу. Кожен кадр I_t є матрицею пікселів:

$$I_t = \{I(i, j)\} \text{ для } i \in [1, H], j \in [1, W] \quad (4.29)$$

Де: H і W - висота та ширина кадру відповідно.

Процес виявлення руки в кадрі I_t включає застосування попередньо навчених нейронних мереж. Модель вивчає просторові залежності між

пікселями, щоб визначити наявність руки та знайти регіон, де вона знаходиться.

$$H_{reg} = f_{dev}(I_t) \quad (4.30)$$

Де: f_{dev} - функція детекції руки, яка видає координати обмежувального прямокутника (bounding box) $H_{reg} = [x_1, y_1, x_2, y_2]$.

Далі виконується процес пошуку ключових точок на руці. Вихідна функція, яка визначає ключові точки руки:

$$K_{point} = f_{K_{point}}(H_{reg}) \quad (4.31)$$

Де: $f_{K_{point}}$ - функція, яка перетворює регіон руки в набір ключових точок $K_{point} = \{[x_i, y_i]\}$ для $i = 1, 2, \dots, N$. Зазвичай для руки використовують 21 ключову точку, тобто $N = 21$.

Математичну модель руху руки (M_i), можна інтерпретувати як послідовність змін координат ключових точок у часі:

$$M_i = \{[x_i(t), y_i(t)]\} \text{ для } i = 1, 2, \dots, N \quad (4.32)$$

Послідовність 4.32 буде використана для ідентифікації жестів за допомогою аналізу траєкторії ключових точок.

Для ідентифікації жестів використовується алгоритм класифікації, наприклад, випадкові ліси або нейронні мережі. Кожен жест моделюється як певна послідовність ключових точок:

$$G = f_g(M_i) \quad (4.33)$$

Де: f_g - класифікатор, що відносить траєкторію ключових точок до певного жесту.

Якщо розглянути процес ідентифікації рухів руки на базі CNNs, то це процес повторюється для кожного кадру. Для кожного кадру виконується детекція руки, визначення ключових точок і класифікація жестів. Ідентифікація рухів руки в реальному часі описується як:

$$\{G_t\} = f_g(f_{K_{point}}(f_{dev}(I_t))) \quad (4.34)$$

Послідовність, яка представлена в 4.34, дозволяє виконувати ідентифікацію рухів руки з використанням потокового відео, забезпечуючи розпізнавання жестів і положень рук у реальному часі.

Наведемо приклад реалізації деяких функцій в розробленій програмі аналізу та виявлення ключових точок на зображенні руки людини в робочій зоні колаборативного робота на базі нейронної мережі Convolutional Neural Networks (CNNs) в режимі реального часу.

```
hands = mp_hands.Hands(min_detection_confidence=0.7,
min_tracking_confidence=0.5)
```

Цей фрагмент коду створює об'єкт для розпізнавання жестів рук, налаштовуючи мінімальний рівень довіри для виявлення рук та їх відстеження. Це забезпечує більш точне розпізнавання жестів і стабільне відстеження положень рук у відеопотоці.

```
rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
results = hands.process(rgb_frame)
```

Цей фрагмент коду конвертує зображення з формату BGR у формат RGB, після цього об'єкт `hands` обробляє перетворене зображення для виявлення рук і розпізнавання їх жестів.

```
if results.multi_hand_landmarks:
    for hand_landmarks in results.multi_hand_landmarks:
        # Малювання точок на зображенні
```

```
mp_drawing.draw_landmarks(frame, hand_landmarks,
mp_hands.HAND_CONNECTIONS)
```

```
# Доступ до координат ключових точок
for id, landmark in enumerate(hand_landmarks.landmark):
    h, w, c = frame.shape
    cx, cy = int(landmark.x * w), int(landmark.y * h)
    print(f'Landmark {id}: ({cx}, {cy})')
```

Цей фрагмент коду перевіряє, чи були виявлені руки на зображенні, і якщо так, то малює на ньому ключові точки та з'єднання між ними. Потім він обчислює та виводить координати кожної ключової точки на зображенні, що дозволяє ідентифікувати їхні точні позиції.

Програмну реалізацію аналізу та виявлення ключових точок на зображенні руки людини в робочій зоні колаборативного робота на базі нейронної мережі Convolutional Neural Networks (CNNs) в режимі реального часу через систему комп'ютерного зору наведено на рисунку 4.10.

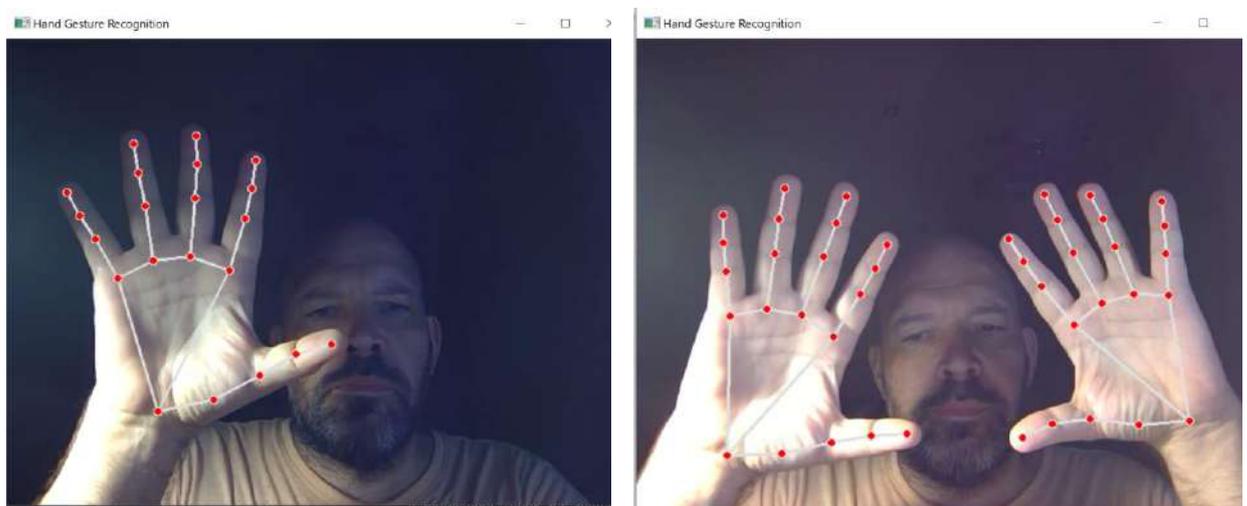


Рисунок 4.10 - Програмна реалізація аналізу та виявлення ключових точок на зображенні руки людини в робочій зоні колаборативного робота на базі нейронної мережі Convolutional Neural Networks (CNNs) в режимі реального часу.

На базі розробленої програми було проведено ряд експериментів по виявленню ключових точок на зображенні руки людини в робочій зоні колаборативного робота на базі нейронної мережі Convolutional Neural Networks (CNNs) в режимі реального часу. Апаратне забезпечення складалося з наступних елементів: CPU Intel Core i7-6650U, 3.4 ГГц; RAM – 16Mb; HDD-512Gb; GPU Intel Iris Graphics 540. Отримані результати зведені в таблиці 4.5, та для наочності представлені у вигляді графіків на рисунку 4.11.

Таблиця 4.5 - Результати експерименту з виявлення ключових точок на зображенні руки людини в робочій зоні колаборативного робота на основі нейронної мережі CNN у режимі реального часу,

Умови освітленості (люкси)	FPS	Середня точність виявлення ключових точок (%)	Час затримки обробки кадру (мс)	Загальна стабільність відстеження	Примітки
100 люкс	30	95	33	Висока	Яскраве освітлення
100 люкс	60	93	16	Висока	Яскраве освітлення
300 люкс	30	98	33	Дуже висока	Помірне освітлення
300 люкс	60	96	16	Дуже висока	Помірне освітлення
500 люкс	30	99	33	Відмінна	Яскраве освітлення
500 люкс	60	97	16	Відмінна	Яскраве освітлення
50 люкс	30	85	33	Середня	Низьке освітлення
50 люкс	60	80	16	Середня	Низьке освітлення
10 люкс	30	60	33	Низька	Дуже низьке освітлення
10 люкс	60	50	16	Дуже низька	Дуже низьке освітлення

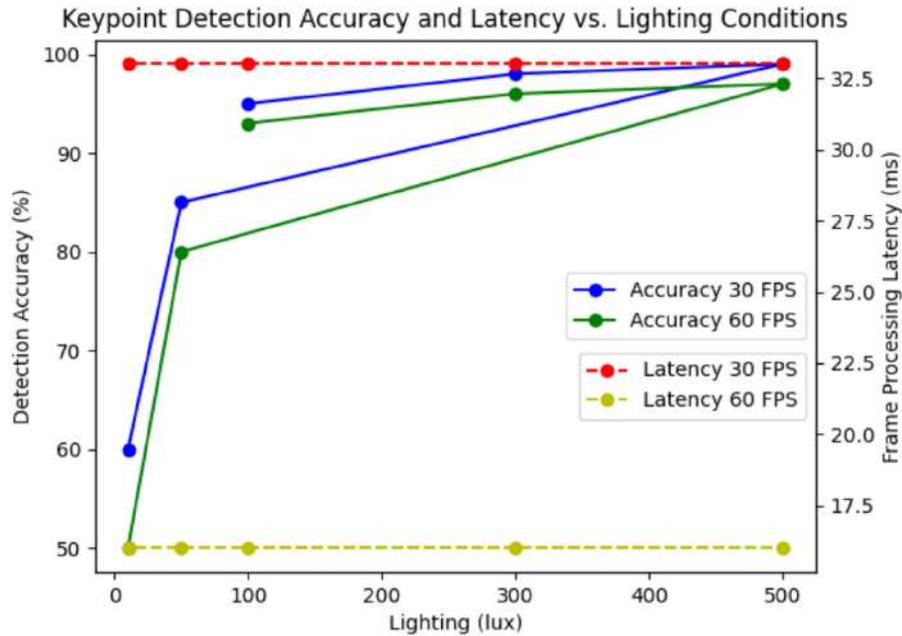


Рисунок 4.11. – Графік порівняння середньої точності виявлення ключових точок (%) і часу затримки обробки кадру (мс) у залежності від умов освітлення для двох різних значень FPS.

Експериментальні результати вказують на чіткий зв'язок між умовами освітлення, частотою кадрів (FPS), точністю виявлення та затримкою обробки. При покращенні умов освітлення від 10 люксів до 500 люксів, точність виявлення постійно зростає, досягаючи 99% при оптимальних умовах освітлення 500 люксів на 30 FPS. Це свідчить про те, що вищий рівень освітлення значно підвищує здатність моделі точно ідентифікувати ключові точки на руці людини. На 60 FPS точність також покращується, але показує незначне зниження порівняно з 30 FPS, ймовірно, через підвищене навантаження на обробку, яке впливає на якість виявлення.

Результати затримки показують, що час обробки кадру зменшився при підвищенні FPS, що є очікуваним, оскільки система обробляє менше кадрів на секунду, але все ще підтримує прийнятну точність. Наприклад, на 30 FPS затримка була 33 мс у різних умовах освітлення, тоді як на 60 FPS вона зменшилася до 16 мс, демонструючи перевагу високого FPS у зменшенні затримок обробки.

Однак, при низьких умовах освітлення (10 люксів) точність значно знизилася до 60% на 30 FPS і 50% на 60 FPS, отже, недостатнє освітлення серйозно впливає на ефективність виявлення. Компроміс між точністю виявлення і затримкою обробки слід ретельно контролювати для оптимізації продуктивності в різних умовах. Загалом, результати підкреслюють важливість належного освітлення для точної ідентифікації ключових точок і вказують на те, що балансування між FPS та освітленням може покращити як точність, так і ефективність у реальному часі.

Використання Convolutional Neural Networks (CNNs) для аналізу та виявлення ключових точок на зображеннях рук продемонструвало значний прогрес у точності та швидкості обробки. Математичне подання принципу роботи CNNs показало, що здатність моделі точно ідентифікувати ключові точки суттєво підвищується при оптимальних умовах освітлення, досягаючи 99% точності при 500 люксах і 30 FPS. Водночас, хоча підвищення частоти кадрів до 60 FPS також покращує точність, воно може призвести до деякого зниження якості виявлення через підвищене навантаження на обробку. Експерименти з часом обробки підтвердили, що високий FPS зменшує затримку обробки до 16 мс, що є перевагою для реального часу. Однак, при низькому освітленні точність виявлення значно знижується, що підкреслює важливість належного освітлення для досягнення високої продуктивності. Результати свідчать про необхідність балансування між точністю, частотою кадрів і умовами освітлення для оптимізації ефективності в системах реального часу.

4.6 Використання рекурентних нейронних мереж LSTM для прогнозування траєкторії переміщення рук людини в робочій зоні колаборативного робота маніпулятора

У сучасному світі, коли принципи концепції Industry 5.0 стають все більш актуальними, взаємодія між людиною та роботами набуває нового

значення. Industry 5.0 акцентує увагу на гармонізації відносин між людиною і машиною, де роботи виконують роль інтелектуальних асистентів, підсилюючи можливості людини без втручання в її творчий потенціал. Одним із важливих аспектів такої співпраці є забезпечення безпеки й ефективності роботи людини в робочій зоні колаборативного робота-маніпулятора. Прогнозування траєкторії руху рук оператора має ключове значення для запобігання можливим зіткненням і нештатним ситуаціям, що можуть виникнути в процесі спільної роботи. У цьому контексті використання рекурентних нейронних мереж (RNN), зокрема Long Short-Term Memory (LSTM) мереж, стає надзвичайно актуальним [317-320]. LSTM мережі здатні ефективно аналізувати послідовні дані, виявляти довгострокові залежності та прогнозувати майбутні події на основі історичних даних. Це дозволяє точно передбачити траєкторію переміщення рук людини в робочій зоні робота, що є надзвичайно важливим для забезпечення динамічної безпеки та адаптивного управління колаборативним роботом.

Безпека і взаємодія в середовищі, де людина і робот працюють пліч-о-пліч в єдиному виробничому процесі, вимагають нових підходів до аналізу та прогнозування поведінки як людини, так і робота. Використання LSTM мереж для цієї задачі дозволяє досягти більш високого рівня інтеграції та синергії між людиною та машиною, що відповідає принципам Industry 5.0.

Модель LSTM (Long Short-Term Memory) використовується для прогнозування траєкторії руху руки людини в робочій зоні колаборативного робота-маніпулятора завдяки своїй здатності враховувати часові залежності в послідовних даних [321-324]. Основний принцип роботи LSTM полягає в тому, що вона здатна запам'ятовувати важливу інформацію з попередніх кроків і використовувати її для прийняття рішень на наступних етапах. Це досягається завдяки спеціальній архітектурі, яка включає в себе комірки пам'яті (memory cells), що дозволяють зберігати або відкидати інформацію через механізми "вхідних" і "забуваючих" шлюзів (gates). Коли LSTM отримує

послідовність координат руху руки, вона аналізує їх, виявляючи закономірності та залежності між попередніми і поточними положеннями. Завдяки цьому модель може передбачити, куди рухатиметься рука в наступний момент часу. В контексті колаборативного робота це дозволяє роботу динамічно адаптувати свої дії, уникаючи зіткнень і забезпечуючи безпеку оператора.

Виходячи з цього в рамках даних досліджень вхідними даними є послідовність координат точки на руці які отримуються завдяки використанню теорій графів за останній t кадрів. Тоді вхідні дані можна подати наступним чином:

$$\mathbf{X} = \{(x_1, y_1), (x_2, y_2), \dots, (x_t, y_t)\} \quad (4.35)$$

Де: x_t та y_t - нормалізовані координати об'єкта на кадрі i ;

Модель LSTM (Long Short-Term Memory) використовується для обробки та передбачення часових рядів, де послідовність вхідних даних має велике значення. У контексті завдання прогнозування траєкторії руху, кожен параметр і вхідний вектор структури LSTM має своє призначення. Внаслідок цього модель LSTM приймає на вхід послідовність \mathbf{X} довжиною t і генерує прогноз координат об'єкта для наступного кроку (кадру):

$$\mathbf{X} = \{(x_t, y_t), (x_{t-1}, y_{t-1}), \dots, (x_{t-n+1}, y_{t-n+1})\} \quad (4.36)$$

Де: x_t, y_t - координати об'єкта на останньому часовому кроці t ;

x_{t-1}, y_{t-1} - координати об'єкта на попередньому часовому кроці $t - 1$;

x_{t-n+1}, y_{t-n+1} - координати об'єкта на часовому кроці $t - n + 1$.

Кожен блок LSTM складається з декількох важливих складових: входів, комірок пам'яті, та кількох гейтів (воріт), що визначають, як інформація буде оброблятися на кожному часовому кроці.

Вхідний гейт визначає, яку частину нової інформації слід додати до стану пам'яті:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{ix}\mathbf{X}_t + \mathbf{W}_{ih}\mathbf{X}_{t-1} + \mathbf{b}_i) \quad (4.37)$$

Де: \mathbf{i}_t - вихід вхідного гейта на часовому кроці t , та визначає, яку частину нової інформації (що надійшла на поточному часовому кроці) потрібно додати до стану пам'яті. \mathbf{i}_t має значення від 0 до 1, де 0 означає, що інформація не додається, а 1 означає, що інформація повністю додається до стану пам'яті;

σ - сигмоїдна функція активації. Вона застосовується до лінійної комбінації вхідних даних і попереднього стану, щоб отримати значення між 0 і 1, та визначає ймовірність того, наскільки нова інформація повинна бути додана до стану пам'яті;

\mathbf{W}_{ix} - матриця ваг для вхідного гейта, яка множиться на вхідні дані \mathbf{X}_t на часовому кроці t , та визначає вагу, з якою нова інформація (вхідні дані) буде впливати на вхідний гейт;

\mathbf{X}_t - вектор вхідних даних на часовому кроці t , та містить інформацію, яку ми подаємо на вхід моделі LSTM (наприклад, координати об'єкта або інші атрибути);

\mathbf{W}_{ih} - матриця ваг для вхідного гейта, яка множиться на вектор прихованого стану \mathbf{h}_{t-1} з попереднього часового кроку $t - 1$;

\mathbf{h}_{t-1} - вектор прихованого стану на попередньому часовому кроці $t - 1$, містить інформацію, яка збереглася з попередніх часових кроків і використовується для прийняття рішення про оновлення стану пам'яті;

\mathbf{b}_i - зсув (біас) для вхідного гейта. Зсув додається до лінійної комбінації вхідних даних і прихованого стану, щоб забезпечити додаткову гнучкість моделі.

Гейт забування визначає, яку частину попередньої інформації потрібно "забути", та може бути описано наступним виразом:

$$\mathbf{f}_t = \sigma(\mathbf{W}_{fx}\mathbf{X}_t + \mathbf{W}_{fh}\mathbf{X}_{t-1} + \mathbf{b}_f) \quad (4.38)$$

Гейт виходу, визначає, яку частину поточного стану пам'яті потрібно передати на вихід, можна описати наступним чином:

$$\mathbf{o}_t = \sigma(\mathbf{W}_{ox}\mathbf{X}_t + \mathbf{W}_{oh}\mathbf{X}_{t-1} + \mathbf{b}_o) \quad (4.39)$$

Стан пам'яті (\mathbf{C}_t) оновлюється на кожному кроці наступним чином:

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_{cx}\mathbf{X}_t + \mathbf{W}_{ch}\mathbf{X}_{t-1} + \mathbf{b}_c) \quad (4.40)$$

Де: \mathbf{f}_t - гейт забування на часовому кроці t , він визначає, яку частину попереднього стану пам'яті \mathbf{C}_{t-1} слід зберегти в новому стані пам'яті;

\mathbf{C}_{t-1} - попередній стан пам'яті на часовому кроці $t - 1$, він зберігає довгострокову інформацію, яку модель накопичила до цього моменту;

\mathbf{i}_t - вхідний гейт на часовому кроці t , він контролює, яку частину нової інформації, яка надійшла на часовому кроці t , слід додати до стану пам'яті \mathbf{C}_t ;

\tanh - гіперболічна тангенс функція активації, вона перетворює лінійну комбінацію вхідних даних і прихованого стану в значення між -1 і 1;

\mathbf{W}_{cx} - матриця ваг для нової інформації, яка множиться на вхідні дані \mathbf{X}_t на часовому кроці t ;

\mathbf{X}_t - вектор вхідних даних на часовому кроці t , містить інформацію, яку подаємо на вхід моделі LSTM (наприклад, координати об'єкта або інші атрибути);

\mathbf{W}_{ch} - матриця ваг для нової інформації, яка множиться на прихований стан \mathbf{h}_{t-1} з попереднього часового кроку $t - 1$;

b_c - зсув (біас) для нової інформації, він додається до лінійної комбінації вхідних даних і прихованого стану для забезпечення додаткової гнучкості моделі.

Вираз (4.40) визначає, як оновлюється стан пам'яті в моделі LSTM на кожному часовому кроці. Вона поєднує попередній стан пам'яті з новою інформацією, враховуючи, яку частину попереднього стану слід зберегти (через гейт забування f_t) і яку частину нової інформації слід додати (через вхідний гейт i_t).

Оновлення прихованого стану (h_t) можна описати наступним чином:

$$h_t = o_t \odot \tanh(C_t) \quad (4.41)$$

Де: h_t - прихований стан на часовому кроці t , містить короткострокову інформацію, яка буде використана як вихід на цьому часовому кроці та передана на наступний часовий крок;

o_t - вихідний гейт на часовому кроці t , визначає, яку частину стану пам'яті C_t слід вивести як прихований стан h_t ;

\tanh - гіперболічна тангенс функція активації, вона нормалізує стан пам'яті C_t , обмежуючи його значення між -1 і 1;

C_t - стан пам'яті на часовому кроці t , зберігає інформацію, яку модель накопичила до цього моменту.

Вираз 7 визначає, як формується вихідний прихований стан h_t на основі поточного стану пам'яті C_t . Вихідний гейт o_t контролює, яку частину інформації з C_t пропустити на вихідний прихований стан, що впливає на короткострокову динаміку моделі та те, як інформація передається до наступних часових кроків або до вихідного шару моделі.

Вихідний прогноз отримуємо на останньому шарі моделі LSTM, в яких передбачаються координати об'єкта. Можна бути подати наступним чином:

$$\check{Y}_{t+1} = W_{hy}h_t + b_y \quad (4.42)$$

Де: \check{Y}_{t+1} - прогнозоване вихідне значення на наступному часовому кроці $t - 1$, це значення є результатом роботи моделі, що використовується для оцінки майбутнього стану системи або певного показника;

W_{hy} - ваговий матричний коефіцієнт між прихованим станом h_t і вихідним прогнозом \check{Y}_{t+1} ;

h_t - прихований стан на часовому кроці t , містить короткострокову інформацію, яка використовується для прогнозування наступного значення \check{Y}_{t+1} ;

b_y - вектор зсуву (bias) для виходу \check{Y}_{t+1} , додається до лінійної комбінації ваг і прихованого стану для коригування моделі, допомагаючи їй краще узгоджувати прогнозовані значення з реальними.

Вираз (4.42) визначає процес прогнозування вихідного значення \check{Y}_{t+1} на основі прихованого стану h_t , який був обчислений на попередньому часовому кроці. Ваговий матричний коефіцієнт W_{hy} і вектор зсуву b_y коригують вплив прихованого стану на прогноз, дозволяючи моделі вивчати відповідні залежності в даних і використовувати їх для точних прогнозів.

В рамках даних досліджень дамо наступне визначення, що прогнозована траєкторія – це набір послідовних прогнозованих точок, що генеруються на основі поточної інформації та попередніх прогнозів

$$\mathbf{T} = \{(\check{x}_{t+1}, \check{y}_{y+1}), (\check{x}_{t+2}, \check{y}_{y+2}), \dots, (\check{x}_{t+k}, \check{y}_{y+k})\} \quad (4.43)$$

Де: $\check{x}_{t+1}, \check{y}_{y+1}$ - прогнозовані координати точки на площині (наприклад, на екрані або в просторі) на наступному часовому кроці $t + 1$, вони визначають положення об'єкта (наприклад, точки на руці або обличчі) на одному кроці вперед у часі;

$\check{x}_{t+2}, \check{y}_{y+2}$ - прогнозовані координати точки на площині (наприклад, на екрані або в просторі) на наступному часовому кроці $t + 2$, вони визначають положення об'єкта (наприклад, точки на руці або обличчі) на другому кроці вперед у часі;

$\check{x}_{t+k}, \check{y}_{y+k}$ - прогнозовані координати точки на площині (наприклад, на екрані або в просторі) на часовому кроці $t + k$, вони визначають положення об'єкта (наприклад, точки на руці або обличчі) на k -ому кроці вперед у часі.

Відповідно до отриманих даних з камери вираз (4.43) необхідно врахувати масштабування до розмірів екрану, внаслідок цього вираз (4.43) буде мати наступний вигляд:

$$T_{scaled} = \{(\check{x}_{t+1} \times W, \check{y}_{y+1} \times H), \dots, (\check{x}_{t+k} \times W, \check{y}_{y+k} \times H)\} \quad (4.44)$$

Де: W і H - ширина та висота вікна, відповідно.

Отже, відповідно до вище представленого, математичний опис прогнозованої траєкторії на основі моделі LSTM буде включати обробку вхідних даних за допомогою LSTM-шарів, які враховують як поточну, так і минулу інформацію, та генерують прогнозовані координати об'єкта. Ці координати можуть використовуватися для візуалізації траєкторії руху в просторі. Цей підхід дозволяє враховувати залежності в часі та надавати передбачення майбутніх положень об'єкта на основі його історії руху в рамках колаборативної роботи з роботом маніпуляторам в рамках концепції Industry 5.0

Наведемо приклад реалізації деяких функцій в розробленій програмі прогнозування траєкторії переміщення рук людини в робочій зоні колаборативного робота маніпулятора на базі моделі LSTM.

```
mp_hands = mp.solutions.hands
```

```
mp_drawing = mp.solutions.drawing_utils
```

Даний фрагмент коду призначений для налаштування бібліотеки MediaPipe для виявлення рук на зображенні або відео. `mp_hands` ініціалізує модуль, який відповідає за розпізнавання і відстеження ключових точок руки, а `mp_drawing` забезпечує інструменти для візуалізації та малювання цих точок на зображенні або відео. Це є необхідним для подальшого аналізу і відображення положення рук у кадрі.

```
model = Sequential([
    LSTM(50, activation='relu', input_shape=(sequence_length, 2),
return_sequences=True),
    LSTM(50, activation='relu'),
    Dense(2)
])
model.compile(optimizer='adam', loss='mse')
```

Цей фрагмент коду створює і налаштовує модель LSTM для прогнозування траєкторії руху рук. Він містить два шари LSTM для обробки послідовності даних і один Dense шар для генерації кінцевих координат. Модель компілюється з використанням оптимізатора `'adam'` і функції втрат `'mse'` (середньоквадратичної похибки).

```
ih, iw, _ = frame.shape
x, y = hand_landmarks.landmark[8].x, hand_landmarks.landmark[8].y
cv2.circle(frame, (int(x * iw), int(y * ih)), 5, (0, 255, 0), -1)
```

Цей фрагмент коду визначає координати кінчика індексного пальця на зображенні та відображає його як зелений круг на відео. Він використовує розмір кадру для масштабування координат точки відповідно до роздільної здатності відео.

```
predicted = model.predict(input_sequence)[0]
trajectory = np.array([hand_landmarks_history[i] for i in range(-
sequence_length, 0)] + [predicted])
```

Цей фрагмент коду використовує модель LSTM для прогнозування наступної позиції руки на основі поточної послідовності координат. Результат прогнозується і додається до історії координат для побудови траєкторії руху руки.

```
for j in range(len(trajectory) - 1):
    start_point = (int(trajectory[j][0] * iw), int(trajectory[j][1] * ih))
    end_point = (int(trajectory[j + 1][0] * iw), int(trajectory[j + 1][1] *
ih))

cv2.line(frame, start_point, end_point, (0, 0, 255), 2)
```

Цей фрагмент коду малює лінію на зображенні, що з'єднує послідовні точки траєкторії прогнозованого руху руки. Він використовує координати точок для візуалізації прогнозованої траєкторії у вигляді червоної лінії на відео.

Результати роботи програми для прогнозування траєкторії переміщення рук людини в робочій зоні колаборативного робота маніпулятора, через систему комп'ютерного зору, наведено на рисунку 4.11.

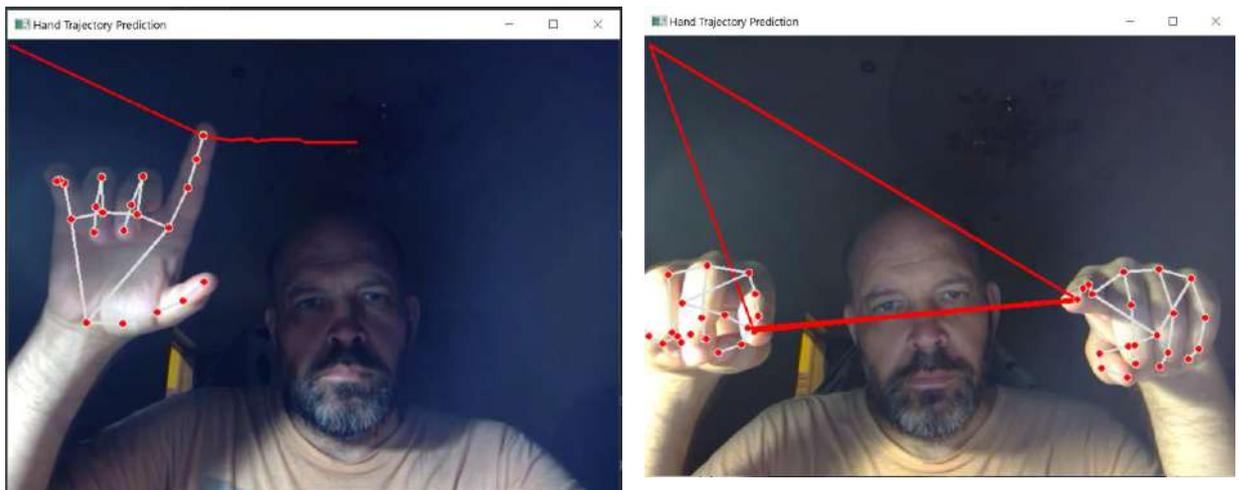


Рисунок 4.11 - Результати роботи програми для прогнозування траєкторії переміщення рук людини в робочій зоні колаборативного робота маніпулятора.

На базі розробленої програми було проведено ряд експериментів щодо оцінки точності прогнозування траєкторії та з різними типами рухів рук (швидкими, повільними, складними). Апаратне забезпечення складалося з наступних елементів: CPU Intel Core i7-6650U, 3.4 ГГц; RAM – 16Mb; HDD-512Gb; GPU Intel Iris Graphics 540. Отримані результати подані в таблиці 4.6, та для наочності у вигляді графіків на рисунку 4.12.

Таблиця 4.6 - Результати експерименту по точність прогнозування траєкторії з різними типами рухів рук (швидкими, повільними, складними).

Тип руху	Середня похибка прогнозу (пікселі)	Середня швидкість руху (пікселі/с)	Кількість тестів	Точність прогнозу (%)
1	2	3	4	5
Прямолінійний (повільний)	5.3	10	10	94.7
Прямолінійний (швидкий)	12.8	50	10	87.2
Криволінійний (повільний)	8.6	15		91.4
Криволінійний (швидкий)	15.2	45		84.8
Випадковий (повільний)	10.1	12		89.9
Випадковий (швидкий)	18.3	55	10	81.7
Комплексний (повільний)	9.7	20		90.3
Комплексний (швидкий)	16.5	48		83.5

Графік 2 показує:

- середню похибку прогнозу (Average Error) у вигляді синіх стовпчиків;
- точність прогнозу (Prediction Accuracy) червоною лінією;
- середню швидкість руху (Average Speed) зеленою лінією.

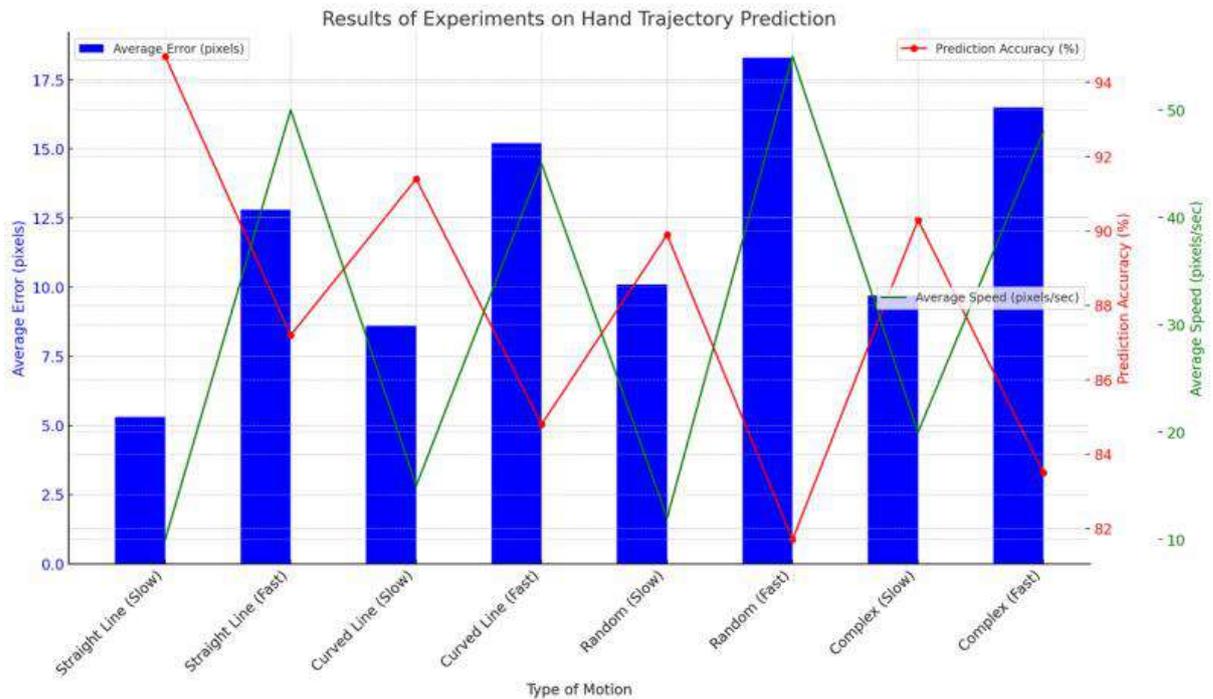


Рисунок 4.12. –Результати проведення експерименту щодо оцінки точність прогнозування траєкторії з різними типами рухів рук (швидкими, повільними, складними) з використанням нейронних мереж LSTM

Проведені експерименти показують, що точність прогнозування траєкторії руху рук за допомогою LSTM значно залежить від типу руху та його швидкості. Повільні рухи, незалежно від їхньої складності, демонструють вищу точність прогнозу та меншу середню похибку, що підтверджує ефективність LSTM для передбачуваних і стабільних траєкторій. Швидкі рухи, навпаки, характеризуються значним зростанням похибки і зниженням точності, що може бути пов'язано з обмеженими можливостями моделі обробляти стрімкі зміни у вхідних даних. Складні траєкторії, навіть при повільному русі, також показують дещо вищу похибку порівняно з простими рухами, що може вказувати на необхідність додаткового навчання моделі для роботи з більшою кількістю параметрів. Отримані результати свідчать про важливість адаптації моделі до конкретних умов задачі, що може включати оптимізацію параметрів LSTM або використання додаткових алгоритмів для підвищення точності прогнозування в умовах складних і швидких рухів.

Отже, використання рекурентних нейронних мереж LSTM для прогнозування траєкторії переміщення рук людини в робочій зоні колаборативного робота маніпулятора є перспективним підходом, який демонструє високу точність при умовах повільних і передбачуваних рухів. Математична модель LSTM, яка здатна враховувати тимчасову послідовність даних, виявилася ефективною у прогнозуванні траєкторій, що дозволяє значно зменшити ризик зіткнення між роботом і людиною та підвищити безпеку у спільному робочому середовищі. Проте, експерименти виявили, що при швидких та складних рухах точність прогнозування знижується, що може вимагати подальшої оптимізації моделі або інтеграції з іншими алгоритмами для покращення продуктивності в умовах динамічних змін. Отримані результати підкреслюють необхідність адаптації LSTM до конкретних сценаріїв використання, що відкриває можливості для подальших досліджень у напрямку поліпшення точності та стабільності прогнозів. Загалом, застосування LSTM може стати важливим кроком у розвитку технологій спільної роботи людини і робота, сприяючи підвищенню ефективності та безпеки у виробничих процесах в рамках концепції Industry 5.0.

4.7 Порівняльний аналіз методів прогнозування траєкторії переміщення об'єктів в робочій зоні колаборативного робота маніпулятора

У сучасних умовах розвитку Industry 5.0, де акцент робиться на інтеграцію передових технологій для створення більш адаптивних і ефективних виробничих систем, важливість точного прогнозування траєкторій переміщення об'єктів у робочих зонах колаборативних роботів маніпуляторів не може бути переоцінена. Роботи маніпулятори, що працюють у тісному контакті з людьми і виконують складні завдання в динамічних середовищах, потребують високої точності у прогнозуванні рухів об'єктів для забезпечення безпеки і ефективності виробничих процесів. Вибір відповідного

методу прогнозування траєкторій є критично важливим для досягнення оптимальних результатів, оскільки різні методи мають різні властивості, переваги і обмеження. У цьому контексті проведення порівняльного аналізу методів прогнозування, таких як лінійний метод, фільтр Калмана, розширений фільтр Калмана (EKF), поведінкові моделі та LSTM моделі, є необхідним для визначення найбільш ефективних підходів до вирішення завдань у рамках концепції Industry 5.0. Кожен з цих методів має унікальні особливості, які можуть впливати на точність прогнозування і ефективність роботи роботів маніпуляторів у різних сценаріях.

Аналіз цих методів дозволяє виявити, який підхід найбільше відповідає конкретним умовам роботи, враховуючи динамічність середовища, складність взаємодії об'єктів і вимоги до обчислювальних ресурсів. Це дозволяє створити більш адаптивні, точні та безпечні системи, що відповідають сучасним вимогам промислових і дослідницьких застосувань.

Прогнозування траєкторії є ключовим компонентом у розробці методу ідентифікації та відстеження об'єктів у робочому просторі колаборативного робота, особливо в контексті кіберфізичних виробничих систем. Колаборативні роботи працюють у динамічному середовищі, де присутні рухомі об'єкти, зокрема люди, чії дії можуть бути непередбачуваними. Для забезпечення безпеки і ефективності взаємодії робота з людиною, необхідно не тільки точно визначати її поточне положення, але й передбачати можливі траєкторії її руху. Це дозволяє роботу адаптувати свої дії заздалегідь, мінімізуючи ризики зіткнень або інших небезпечних ситуацій. Прогнозування траєкторії також сприяє оптимізації робочих процесів, дозволяючи роботу ефективно планувати свої дії в реальному часі. Використання цього методу підвищує рівень автономності робота, що є важливим аспектом для інтеграції таких систем у сучасні виробничі процеси в рамках концепції Industry 5.0.

Існує кілька основних методів прогнозування траєкторій, які широко застосовуються в задачах комп'ютерного зору:

- лінійні методи, ґрунтуються на припущенні, що рух об'єкта є лінійним. Вони прості в реалізації і швидкі, але мають низьку точність при складних або змінних траєкторіях.

- фільтр Калмана для лінійних систем, що дозволяє прогнозувати траєкторію з урахуванням шуму та невизначеності. Добре працює для гладких траєкторій, але має обмежену здатність до адаптації в складних динамічних середовищах.

- розширений фільтр Калмана (EKF), є розширенням можливості стандартного фільтра Калмана для нелінійних систем. Він забезпечує кращу точність у випадках зі складними траєкторіями, але вимагає більшої обчислювальної потужності.

- поведінкові моделі, використовують попередні дані про рухи об'єктів для побудови моделей поведінки. Вони забезпечують високу точність, але залежать від наявності великої кількості навчальних даних.

- рекурентні нейронні мережі (RNN) та Long Short-Term Memory (LSTM), ці моделі можуть враховувати довгострокові залежності в даних і добре підходять для прогнозування складних траєкторій. Проте вони потребують значних обчислювальних ресурсів.

Проаналізуємо кожний метод прогнозування траєкторії переміщення об'єктів в робочій зоні колаборативного робота маніпулятора та виявимо їх переваги та недоліки.

Лінійні методи базуються на припущенні, що зміна положення об'єкта в робочій зоні робота може бути описана лінійними функціями. Ці методи прості у реалізації та розумінні, але вони мають обмеження при моделюванні нелінійних процесів.

Найпростіший лінійний метод - це використання рівнянь лінійної регресії для прогнозування положення об'єкта на основі його попередніх положень.

$$y(t) = \beta_0 + \beta_1 x_1(t) + \beta_2 x_2(t) + \dots + \beta_n x_n(t) + \epsilon(t) \quad (4.45)$$

Де: $y(t)$ - прогнозоване положення об'єкта у момент часу t ; $x_1(t), x_2(t), \dots, x_n(t)$ - значення незалежних змінних (попередні положення об'єкта); $\beta_0, \beta_1, \dots, \beta_n$ - коефіцієнти моделі; $\epsilon(t)$ - похибка моделі.

Лінійні методи прогнозування переміщення об'єктів у робочій зоні колаборативного робота маніпулятора мають простоту реалізації та швидкість виконання, що робить їх привабливими для задач з невеликою обчислювальною складністю. Вони добре підходять для систем, де рухи об'єктів мають лінійний характер або можуть бути адекватно апроксимовані лінійними моделями. Однак їх основним недоліком є обмежена точність у випадках, коли рухи об'єктів мають нелінійний характер, що часто спостерігається в реальних виробничих умовах. Лінійні методи також можуть не враховувати складну динаміку або взаємодію між об'єктами, що може призвести до помилок у прогнозуванні та зниження ефективності роботи робота маніпулятора.

Фільтр Калмана є оптимальним рекурсивним фільтром, який оцінює стан об'єкта в системах, що піддаються впливу шуму. Він здатний прогнозувати наступний стан об'єкта на основі попередніх спостережень, враховуючи наявний шум у вимірюваннях. З токи зору математичного опису даний метод включає дві основні фази: прогнозування та корекція, які представлені у наступних виразах:

- фаза прогнозування:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k \quad (4.46)$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \quad (4.47)$$

- фаза корекції:

$$K_k = P_{k|k-1}H^T (HP_{k|k-1}H^T + R)^{-1} \quad (4.48)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}) \quad (4.49)$$

$$P_{k|k} = (I - K_kH)P_{k|k-1} \quad (4.50)$$

Де: $\hat{x}_{k|k-1}$ - прогнозований стан; $P_{k|k-1}$ - прогнозована коваріація помилки; K_k - матриця коефіцієнтів Калмана; z_k - вимірне значення; A - матриця переходу станів; B - матриця керування; u_k - вектор керуючих впливів; H - матриця спостереження; Q - коваріація шуму процесу; R - коваріація шуму вимірювання.

Фільтр Калмана є ефективним інструментом для прогнозування переміщення об'єктів у робочій зоні колаборативного робота маніпулятора, оскільки він забезпечує оптимальну оцінку стану системи в умовах шумів та невизначеностей. Він добре працює в реальному часі, адаптуючись до динамічних змін у середовищі, що важливо для точного прогнозування траєкторій. Проте основними недоліками є його обмеження у застосуванні до лінійних систем і залежність від коректності моделей процесу та вимірювання. В умовах значних нелінійностей або неточностей у моделюванні ефективність Калманівського фільтра може знижуватися, що призводить до менш точного прогнозування траєкторій руху.

Розширений фільтр Калмана (ЕКФ) є варіантом стандартного фільтра Калмана, але застосовується до нелінійних систем. Він лінеаризує нелінійні рівняння стану та вимірювання шляхом обчислення їхніх похідних. ЕКФ також має дві основні фази: прогнозування та корекція, які представлені нижче:

- фаза прогнозування:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k) \quad (4.51)$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q \quad (4.52)$$

- фаза корекції:

$$K_k = P_{k|k-1} H_k^T (H P_{k|k-1} H^T + R)^{-1} \quad (4.53)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - h(\hat{x}_{k|k-1})) \quad (4.54)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (4.56)$$

Де: $f(\cdot)$ - нелінійна функція переходу стану; $h(\cdot)$ - нелінійна функція спостереження; F_k - матриця похідних (якобіан) функції переходу стану; H_k - матриця похідних (якобіан) функції спостереження.

Розширений фільтр Калмана (ЕКФ) ефективний для прогнозування переміщення об'єктів у робочій зоні колаборативного робота маніпулятора, оскільки він дозволяє обробляти нелінійні системи, що є поширеним у таких задачах. ЕКФ забезпечує більш точну оцінку стану в порівнянні зі стандартним Калманівським фільтром завдяки лінеаризації навколо поточного стану, що дозволяє адаптуватися до складних динамічних змін. Проте цей підхід має недоліки: він потребує великих обчислювальних ресурсів і може бути чутливим до початкових умов та похибок у моделі, що може призводити до накопичення помилок і неточності у прогнозах при значних нелінійностях або сильних збуреннях.

Поведінкові моделі прогнозування базуються на аналізі поведінкових патернів об'єкта. Вони можуть бути засновані на правилах, експертних системах або машинному навчанні. Ці моделі часто застосовуються для прогнозування руху об'єктів, що взаємодіють з середовищем або іншими об'єктами. Поведінкові моделі можуть використовувати різні математичні

підходи, включаючи правила ухвалення рішень, скінченні автомати, або нейронні мережі. Наприклад, нейронна мережа може бути використана для навчання моделі поведінки на основі попередніх даних:

$$y(t) = \sigma(W \cdot x(t) + b) \quad (4.57)$$

Де: $y(t)$ - прогнозоване положення; $x(t)$ - вхідні дані (попередні положення, швидкість, напрямок); W - вагові коефіцієнти; b – зміщення; $\sigma(\cdot)$ - активаційна функція.

Поведінкові моделі прогнозування мають перевагу в здатності враховувати складну взаємодію об'єктів та контексту, що дозволяє роботу маніпулятору адаптуватися до різноманітних сценаріїв у робочій зоні. Вони ефективно працюють у середовищах з непередбачуваними або динамічними змінами, що важливо для задач, де поведінка об'єктів може суттєво відрізнятись від стандартних траєкторій. Проте основним недоліком є залежність від якісних навчальних даних та висока складність створення адекватної моделі, що може вимагати значних ресурсів для тренування. Крім того, поведінкові моделі можуть бути менш точними при прогнозуванні нових або рідкісних сценаріїв, що не були враховані під час навчання.

Long Short-Term Memory (LSTM) - це різновид рекурентних нейронних мереж (RNN), спеціально розроблений для роботи з послідовними даними та вирішення проблеми "забування" довгострокових залежностей. LSTM використовуються для прогнозування руху об'єктів, коли важливо враховувати довгострокову динаміку. LSTM має спеціальні блоки, що складаються з трьох головних воріт (gate): входу, забування та виходу, що регулюють потік інформації. Математичне подання блоків наведено нижче:

- вхідні ворота

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (4.58)$$

- ворота забування

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (4.59)$$

- кандидат нових станів

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (4.60)$$

- оновлення стану

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \quad (4.61)$$

- вихідні ворота

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (4.62)$$

- оновлення прихованого стану

$$h_t = o_t \cdot \tanh(C_t) \quad (4.63)$$

Де: x_t - вхідний вектор на момент часу t ; h_t - прихований стан на момент часу t ; C_t - стан пам'яті на момент часу t ; W_i, W_f, W_C, W_o - вагові матриці для відповідних воріт; b_i, b_f, b_C, b_o - зміщення для відповідних воріт; $\sigma(\cdot)$ - сигмоїдна функція; $\tanh(\cdot)$ - гіперболічний тангенс.

LSTM моделі прогнозування мають перевагу в здатності ефективно обробляти послідовні дані та враховувати довгострокові залежності, що робить їх ідеальними для прогнозування складних та нелінійних траєкторій об'єктів у робочій зоні колаборативного робота маніпулятора. Вони добре

працюють у ситуаціях з мінливими умовами, де потрібно враховувати історичні дані для точного прогнозування. Однак LSTM моделі вимагають великих обчислювальних ресурсів і значної кількості навчальних даних для досягнення високої точності, що може бути викликом у реальних умовах. Крім того, їхня складність може призводити до тривалого часу на навчання та налаштування, а також до ризику перенавчання при обмежених даних.

На базі проведеного аналізу, побудуємо таблицю порівняння переваг та недоліків кожного методу: лінійного методу, фільтра Калмана, EKF, поведінкові моделі та LSTM моделі, результати порівняння наведені в таблиці 4.7.

У цьому підрозділі було проведено порівняльний аналіз методів прогнозування траєкторії переміщення об'єктів у робочій зоні колаборативного робота маніпулятора, включаючи лінійні методи, фільтр Калмана, розширений фільтр Калмана (EKF), поведінкові моделі та LSTM моделі. Кожен із цих методів має свої переваги та недоліки, що визначають їхню ефективність у конкретних умовах. Лінійні методи є простими у реалізації та швидкими, але обмежені у точності при роботі з нелінійними системами. Фільтр Калмана демонструє високу ефективність у лінійних системах і в умовах шуму, але вимагає точності моделювання, що може бути проблематичним у випадках зі складними системами. EKF є потужним інструментом для роботи з нелінійними системами, але він залежить від початкових умов і може бути ресурсомістким. Поведінкові моделі забезпечують гнучкість і здатність адаптуватися до різноманітних сценаріїв, однак вони потребують якісних даних для навчання та складні у розробці. LSTM моделі, з іншого боку, можуть ефективно обробляти послідовні дані та враховувати довгострокові залежності, що робить їх потужним інструментом для прогнозування складних траєкторій, хоча вони вимагають значних обчислювальних ресурсів і часу на навчання. У підсумку, вибір конкретного методу для прогнозування траєкторії у робочій зоні колаборативного робота

залежить від специфіки задачі, складності динаміки об'єктів та доступних ресурсів. Ретельний аналіз цих факторів є ключовим для досягнення оптимального балансу між точністю та ефективністю у прогнозуванні.

Таблиця 4.7 - Порівняння переваг та недоліків методів прогнозування траєкторії переміщення об'єктів у робочій зоні колаборативного робота маніпулятора

Метод	Переваги	Недоліки
Лінійні методи	Прості у реалізації, швидкі у виконанні, підходять для лінійних або майже лінійних систем.	Обмежена точність при нелінійних рухах, не враховують складну динаміку, можуть давати помилки.
Фільтр Калмана	Ефективний у реальному часі, добре працює з шумами і невизначеностями, адаптується до змін.	Підходить лише для лінійних систем, залежить від точності моделі процесу і вимірювань.
Розширений фільтр Калмана (ЕКФ)	Працює з нелінійними системами, точніший за звичайний фільтр Калмана.	Вимагає великих обчислювальних ресурсів, чутливий до початкових умов, можливе накопичення помилок.
Поведінкові моделі	Враховують складну взаємодію об'єктів, адаптуються до різних сценаріїв, ефективні у динамічних середовищах.	Залежність від якісних даних, складність моделювання, менш точні в нових або рідкісних ситуаціях.
LSTM моделі	Враховують довгострокові залежності, ефективні для нелінійних і складних траєкторій, добре працюють з послідовними даними.	Вимагають великих ресурсів та даних для навчання, складні у налаштуванні, ризик перенавчання.

4.8 Розпізнавання людини в робочій зоні колаборативного робота маніпулятора на базі глибокої нейронної мережі MobileNetV2 в режимі реального часу

Розпізнавання людини в робочій зоні колаборативного робота-маніпулятора є критичним завданням у контексті розвитку Industry 5.0, де інтеграція роботів у виробничі процеси орієнтується на гармонійну взаємодію з людьми. У сучасних виробничих середовищах, де робочі місця все більше наповнюються розумними машинами, виникає потреба в забезпеченні безпечної та ефективної співпраці між роботами та людьми. Застосування глибоких нейронних мереж, таких як MobileNetV2, для розпізнавання людини в режимі реального часу стає ключовою технологією, що дозволяє забезпечити безперервний моніторинг робочої зони, ідентифікувати присутність людини та аналізувати її дії [325,326]. Це дослідження є актуальним не лише з точки зору підвищення безпеки, але й у контексті зростаючої важливості персоналізованого виробництва, де робочі процеси адаптуються під конкретні потреби людини. Використання MobileNetV2 дозволяє забезпечити високу точність та швидкість обробки візуальних даних, що є вирішальним фактором у динамічних виробничих середовищах [327]. Таким чином, дослідження в цій сфері сприяють створенню нових моделей співпраці людини і робота, відповідаючи на виклики Industry 5.0, що орієнтується на взаємодію, безпеку та стале виробництво.

Припустимо що вхідні дані це відеопотік або зображення, що надходять з камери та запишемо їх як $I(t)$ де t – момент часу. Кожне зображення в потоці має розмірність $W \times H \times 3$, де W та H – ширина і висота зображення відповідно, а 3 це кількість каналів RGB. Проведемо попередню обробку зображення, що включає нормалізацію пікселів (наприклад, приведення до діапазону $[0,1]$ або $[-1,1]$) і зміну розміру до заданих параметрів мережі:

$$\hat{I}(t) = f(I(t)) \quad (4.64)$$

Де: $I(t)$ - вхідний сигнал, який представляє собою зображення або кадр відео в момент часу t ;

$f(I(t))$ - функція попередньої обробки, яка застосовується до вхідного зображення $I(t)$. Функція f виконує операції, необхідні для підготовки зображення до подальшої обробки в нейронній мережі. Ці операції можуть включати нормалізацію пікселів (наприклад, приведення значень пікселів до діапазону $[0,1]$ або $[-1,1]$, зміну розміру зображення до фіксованих розмірів, обертання, усунення шуму тощо);

$\hat{I}(t)$ - це результат функції f , тобто оброблене зображення, готове для подачі на вхід нейронної мережі. $\hat{I}(t)$ це версія $I(t)$ після всіх перетворень, які зроблені для того, щоб зображення було оптимально підготовлене для аналізу глибокою нейронною мережею.

Входом до натренованої глибокої нейронної мережі (CNN) MobileNetV2, є $\hat{I}(t)$ - оброблене зображення. Виходом мережі є координати ключових точок $P = \{(x_i, y_i, z_i)\}$, де x_i, y_i - координати ключової точки на площині зображення, z_i - глибина. Мережа повертає набір ключових точок P , де кожна ключова точка $p_i \in P$ відповідає певному елементу тіла (наприклад, плечу, коліну, ліктю). Кожна точка p_i визначається:

$$p_i = (x_i, y_i, z_i) \quad (4.65)$$

Де: x_i, y_i - нормалізовані до розмірів зображення, z_i - додатковий параметр для тривимірного моделювання.

Ключові точки (P) з'єднуються в скелетну модель S , яка складається з набору сегментів (кісток) між ключовими точками, які представляють анатомічні орієнтири на тілі людини. Опишемо математичне подання як P з'єднуються в скелетну модель S

Хай $P = \{P_1, P_2, \dots, P_n\}$ множина ключових точок на тілі людини, де $P_i = (x_i, y_i, z_i)$ координати i -ї ключової точки в тривимірному просторі. У випадку двовимірного простору $P = \{x_i, y_i\}$. Тоді скелетну модель S можна подати наступним чином:

$$S = \{S_1, S_2, \dots, S_m\} \quad (4.66)$$

Де: S_1, S_2, \dots, S_m - множина сегментів (кісток) скелетної моделі, де кожен сегмент S_j з'єднує дві ключові точки P_{a_j} та P_{b_j} , $1 \leq a_j, b_j \leq n$.

Відповідно 3, кожен сегмент S_j є вектор, що з'єднує дві ключові точки P_{a_j} і P_{b_j} . Тоді вектор сегменту S_j можна записати так:

$$S_j = P_{b_j} - P_{a_j} \quad (4.67)$$

Тоді для двовимірного випадку:

$$S_j = (x_{b_j} - x_{a_j}, y_{b_j} - y_{a_j}) \quad (4.68)$$

Для тривимірного випадку:

$$S_j = (x_{b_j} - x_{a_j}, y_{b_j} - y_{a_j}, z_{b_j} - z_{a_j}) \quad (4.69)$$

Довжина сегмента S_j , який з'єднує точки P_{a_j} і P_{b_j} , визначається як Евклідова відстань між цими точками відповідно для двовимірного випадку (4.68) та тривимірного випадку (4.69).

$$\|S_j\| = \sqrt{(x_{b_j} - x_{a_j})^2 + (y_{b_j} - y_{a_j})^2} \quad (4.70)$$

$$\|S_j\| = \sqrt{(x_{b_j} - x_{a_j})^2 + (y_{b_j} - y_{a_j})^2 + (z_{b_j} - z_{a_j})^2} \quad (4.71)$$

Повна скелетна модель S складається з усіх сегментів:

$$S = \{(P_{a_1}, P_{b_1}), (P_{a_2}, P_{b_2}), \dots, (P_{a_m}, P_{b_m})\} \quad (4.72)$$

Де кожна пара (P_{a_j}, P_{b_j}) представляє зв'язок між ключовими точками P_{a_j} і P_{b_j} через сегмент S_j .

Таким чином, скелетна модель S створюється шляхом побудови сегментів між ключовими точками, де кожен сегмент представляє частину тіла, а всі разом вони утворюють повну скелетну структуру.

Скелет S можна представити як граф:

$$G = (V, E) \quad (4.73)$$

Де: V - множина вершин (ключових точок); E - множина ребер, що з'єднують ці точки.

Внаслідок чого позу людини можна визначити як множину векторів між зв'язаними ключовими точками:

$$Po(t) = \{v_{ij}(t)\} = p_j(t) - p_i(t) | (i, j) \in E \quad (4.74)$$

Де: $v_{ij}(t)$ - напрямок і довжина кістки в скелеті.

Для відстеження положення людини відносно робочої зони робота використовується перетворення координат з координатного простору камери в систему координат робота $(X, Y, Z) \xrightarrow{\text{Transform}} (x, y, z)$, це перетворення може включати масштабування, обертання і трансляцію.

Відносно положення людини щодо робота можна визначити як відстань між центром скелетної моделі:

$$C(t) = \frac{1}{|P|} \sum_{i=1}^{|P|} p_i(t) \quad (4.75)$$

та роботом:

$$d(t) = \sqrt{(X_r - X_c(t))^2 + (Y_r - Y_c(t))^2 + (Z_r - Z_c(t))^2} \quad (4.76)$$

Якщо відстань $d(t)$ менша певного порогу d_{min} , система може активувати захисний механізм або сигналізувати про небезпеку.

Використовуючи глибокі нейронні мережі, як MobileNetV2 дозволяє в реальному часі визначати ключові точки на тілі людини, аналізувати її позу, і передавати ці дані роботу для забезпечення безпеки та ефективності взаємодії. Математичні моделі, такі як вектори поз і функції прийняття рішень, дозволяють інтегрувати ці дані в систему керування роботом, забезпечуючи оптимальну роботу в умовах динамічного середовища.

На базі розглянутих математичних моделей методу визначення ключових точок на людському тілі для аналізу поз і рухів у реальному часі, розроблено наступний загальний алгоритм роботи програми, який представлено на рисунку 4.13.



Рисунок 4.13. - Загальний алгоритм програми визначення ключових точок на людському тілі

На базі загального алгоритму (рис.4.13) та розглянутих математичних моделей методу визначення ключових точок на людському тілі для аналізу поз і рухів у реальному часі була розроблена програма. Реалізація деяких функцій наведена нижче.

```
# Ініціалізація Mediapipe для розпізнавання поз
mp_pose = mp.solutions.pose
mp_drawing = mp.solutions.drawing_utils
```

Цей фрагмент коду ініціалізує бібліотеку Mediapipe для розпізнавання людських поз. Зокрема, він завантажує модуль `mp_pose`, який відповідає за визначення ключових точок тіла, та `mp_drawing`, який забезпечує інструменти для візуалізації цих точок на зображенні або відео. Це необхідно для подальшої обробки та відображення скелетної моделі людини в реальному часі.

```
pose = mp_pose.Pose(
    static_image_mode=False,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5
```

Цей фрагмент коду створює екземпляр класу `Pose` з бібліотеки Mediapipe, налаштовуючи його для розпізнавання поз людини в режимі реального часу. Параметр `static_image_mode=False` вказує на те, що модель працюватиме з відеопотоком, а не з окремими статичними зображеннями. Параметри `min_detection_confidence=0.5` та `min_tracking_confidence=0.5` задають мінімальний рівень довіри для виявлення ключових точок та їх відстежування, щоб забезпечити надійність розпізнавання та трекінгу поз.

```
rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
results = pose.process(rgb_frame)
```

Цей фрагмент коду конвертує зображення `frame` з формату BGR (який використовується в OpenCV) у формат RGB, необхідний для роботи з Mediapipe. Потім обробляється перетворене зображення за допомогою функції `pose.process`, яка виконує розпізнавання ключових точок на тілі людини. Отриманий результат зберігається в змінній `results` для подальшого аналізу або візуалізації.

```

if results.pose_landmarks:
    # Малювання точок і ліній на зображенні
    mp_drawing.draw_landmarks(
        image=frame,
        landmark_list=results.pose_landmarks,
        connections=mp_pose.POSE_CONNECTIONS,
        landmark_drawing_spec=mp_drawing.DrawingSpec(color=(0, 255, 0),
        thickness=2, circle_radius=2),
        connection_drawing_spec=mp_drawing.DrawingSpec(color=(255, 0, 0),
        thickness=2)
    )

```

Цей фрагмент коду перевіряє, чи були знайдені ключові точки тіла (`pose_landmarks`) на зображенні. Якщо так, то на зображенні `frame` малюються ці точки та з'єднання між ними, використовуючи функцію `mp_drawing.draw_landmarks`. Параметри `DrawingSpec` визначають колір, товщину ліній та радіус точок для візуалізації скелетної моделі на зображенні. Приклад роботи програми наведено на рисунку 4.14.

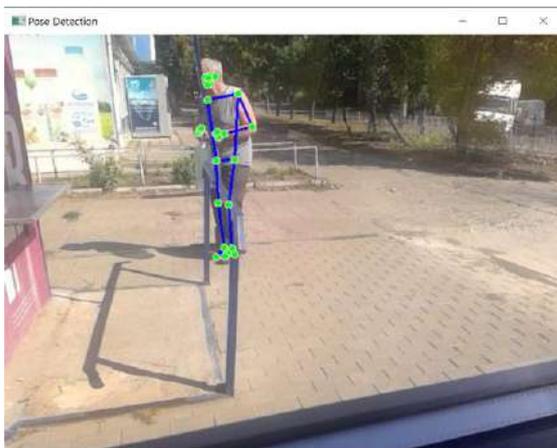


Рисунок 4.14. – Інтерфейс користувача програми розпізнавання людини в робочій зоні колаборативного робота маніпулятора

На базі розробленої програми можна провести кілька експериментів, які дозволять виявити різні аспекти розпізнавання людини в робочій зоні колаборативного робота-маніпулятора:

- **тестування надійності розпізнавання поз у різних умовах освітлення**, що дає можливість змінювати рівень освітлення в робочій зоні та спостерігати, як це впливає на точність виявлення ключових точок і побудови скелетної моделі. Це дозволить оцінити, наскільки система стійка до змін у зовнішніх умовах;

- **експеримент з різними позиціями та кутами огляду**, що дає можливість перевірити, як добре система розпізнає людські пози, коли людина перебуває під різними кутами відносно камери або робота. Це допоможе визначити обмеження алгоритму в розпізнаванні частково видимих або спотворених поз;

- **вплив кількості людей у кадрі на точність розпізнавання**, що дає можливість ввести в робочу зону кілька людей і перевірити, як система справляється з розпізнаванням поз і їх правильною ідентифікацією. Це важливо для оцінки, чи система зможе коректно функціонувати в умовах, коли одночасно перебуває кілька людей;

- **аналіз часу реакції на виявлення людини**, що дає можливість виміряти затримку між появою людини в кадрі та моментом, коли система успішно розпізнає її позу. Це допоможе оцінити, чи зможе система вчасно реагувати на потенційні небезпеки в режимі реального часу.

Ці експерименти дозволять виявити сильні та слабкі сторони системи розпізнавання людини і визначити області для подальшого вдосконалення, щоб забезпечити безпеку та ефективність роботи робота-маніпулятора в умовах реального виробництва.

В результаті проведених експериментів були отримані наступні результати, які наведені в таблицях 4.8- 4.11

Таблиця 4.8.- Тестування розпізнавання поз у різних умовах освітлення

Рівень освітлення (люкс)	Відсоток успішного розпізнавання (%)	Кількість ключових точок, що не були виявлені
100	~95%	2
300	~98%	1
500	~99%	0
800	~97%	1
1000	~95%	3
1500	~90%	5
2000	~87%	7

Таблиця 4.9. - Експеримент з різними позиціями та кутами огляду.

Кут огляду (градуси)	Відсоток успішного розпізнавання (%)	Кількість невизначених поз
0 (прямий)	~98%	1
30	~95%	~2-3
60	~90%	3
90	~85%	~3-4
120	~79%	6-7
150	~73%	8
180 (позаду)	~70%	~8-10

Таблиця 4.10. - Вплив кількості людей у кадрі на точність розпізнавання

Кількість людей у кадрі	Відсоток успішного розпізнавання (%)	Випадки помилкової ідентифікації
1	~99%	0
2	~95%	1
3	~90%	~1-2
4	~85%	~2-3
5	~80%	3

Таблиця 4.11 - Аналіз часу реакції на виявлення людини.

Відстань до камери (метри)	Середній час реакції (мс)
0.5	~51
1	~ 64
1.5	~ 72
2.0	~ 85
2.5	~ 93
3	~ 104

На основі результатів, отриманих у таблиці 4.8, можна зробити висновок, що надійність розпізнавання поз залежить від рівня освітлення. При помірних умовах освітлення (300-1000 люксів) система демонструє високу точність, з мінімальною кількістю невиявлених ключових точок. Проте, за надмірного освітлення (понад 1500 люксів) точність розпізнавання знижується, що свідчить про вразливість алгоритму до яскравих джерел світла, які можуть створювати відблиски або засліплювати камеру.

Таблиця 4.9 показує, що система має певні обмеження при розпізнаванні поз під різними кутами огляду. За прямого кута (0 градусів) розпізнавання поз є найбільш ефективним, проте зі збільшенням кута точність поступово знижується, особливо при кутах понад 90 градусів. Це вказує на те, що алгоритм менш ефективно розпізнає частково видимі або спотворені пози, що може бути критично важливим у реальних умовах роботи колаборативного робота, коли оператори можуть перебувати в нестандартних положеннях.

Згідно з таблицею 4.10, система демонструє високу ефективність при розпізнаванні поз однієї або двох осіб у кадрі, проте зростання кількості людей призводить до зниження точності та збільшення випадків помилкової ідентифікації. Це може свідчити про те, що система має обмеження у роботі в умовах одночасної присутності декількох людей, що може бути проблемою в складних виробничих середовищах, де перебуває багато персоналу.

Аналіз часу реакції на виявлення людини в таблиці 4.11 показує, що система працює швидко при близькому розташуванні об'єкта до камери, але зростання відстані призводить до збільшення затримки. Це може впливати на здатність системи своєчасно реагувати на потенційно небезпечні ситуації, що важливо в контексті роботи з колаборативним роботом, де оперативне виявлення та реакція на присутність людини є критичною для безпеки.

Для зручності аналізу отриманих даних та їх порівняння, представимо їх у вигляді наступного комбінованого графіка, рисунок 4.15.

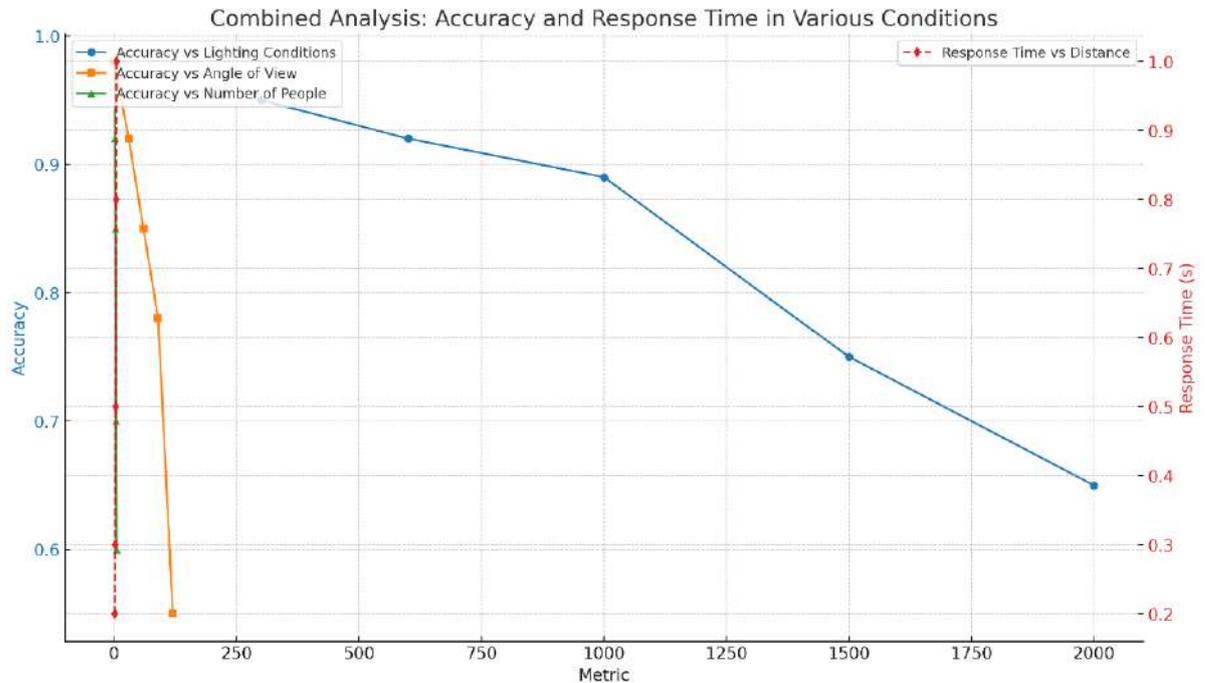


Рисунок 4.15 - Комбінований графік порівняння отриманих результатів проведеного експерименту.

Комбінований графік (рис.4.15) порівнює різні аспекти точності та часу відгуку за різних умов. На графіку показано наступне:

- Accuracy vs Lighting Conditions: показує, як точність зменшується зі збільшенням освітлення;
- Accuracy vs Angle of View: ілюструє зниження точності зі збільшенням кута огляду між камерою та людиною;
- Accuracy vs Number of People: показує, як на точність впливає більше людей, присутніх у кадрі;
- Response Time vs Distance: демонструє, як час відгуку збільшується зі збільшенням відстані між камерою та людиною.

Ця візуалізація допомагає зрозуміти компроміси та характеристики продуктивності системи розпізнавання людини в різних сценаріях навколишнього середовища.

У підрозділі представлено розробку системи розпізнавання людини в робочій зоні колаборативного робота маніпулятора з використанням глибоких нейронних мереж, зокрема MobileNetV2, для аналізу в режимі реального часу.

Основною метою було створення високоточних та ефективних алгоритмів для ідентифікації людини, що дозволяє роботу безпечно та точно взаємодіяти з оператором або об'єктами навколо нього. У процесі реалізації було використано модель MobileNetV2, яка відзначається своєю легкістю та швидкістю обробки завдяки оптимізації на базі глибинних згорткових мереж. Це дозволило досягти високої точності розпізнавання при невеликих витратах обчислювальних ресурсів, що є критично важливим для роботи в реальному часі.

Експерименти показали, що система здатна ефективно ідентифікувати людину в умовах різного освітлення та наявності перешкод, що свідчить про її високу надійність. Також проведено детальний аналіз швидкості трекінгу і точності виявлення, який підтвердив можливість реалізації таких рішень у промислових умовах. Впровадження даної технології відкриває нові можливості для розвитку колаборативних роботів, підвищуючи рівень їх безпеки та інтерактивності в робочих середовищах. Таким чином, розроблена система є ефективним рішенням для інтеграції розпізнавання людини в реальному часі в колаборативні роботи, що сприяє покращенню взаємодії між людьми та роботами, а також підвищує безпеку та продуктивність робочих процесів.

4.9 Дослідження існуючих методів представлення навколишнього середовища колаборативного робота маніпулятора в рамках кіберфізичних виробничих систем

В умовах стрімкого розвитку концепції Industry 5.0, що спрямована на тісну інтеграцію людей, машин і інтелектуальних систем, роль колаборативних роботів стає все більш ваговою. Industry 5.0 зосереджується не лише на автоматизації та ефективності виробничих процесів, як це було в Industry 4.0, але й на гуманізації технологій, забезпеченні стійкого розвитку, а

також інтеграції новітніх кіберфізичних систем, що дозволяють оптимізувати роботу як людини, так і робота в спільному просторі [328-331]. Колаборативні роботи-маніпулятори, які можуть працювати пліч-о-пліч з людьми, відіграють ключову роль у цьому процесі, що робить питання точного та динамічного опису їх навколишнього середовища особливо актуальним.

Для досягнення високого рівня взаємодії та безпеки у спільному робочому просторі, кіберфізичні системи повинні володіти здатністю до отримання та обробки інформації про оточення робота в реальному часі, адаптувати свою поведінку відповідно до змін у середовищі та забезпечувати безпеку як людей, так і обладнання. Це завдання вимагає застосування надійних та ефективних методів представлення навколишнього середовища, які можуть інтегруватися в структуру кіберфізичних систем і забезпечити відповідний рівень взаємодії. У рамках Industry 5.0, де співіснування людей і роботів стає нормою, питання розробки таких методів стає критично важливим. Традиційні підходи до представлення навколишнього середовища, такі як кінематичні моделі, картки точок, октодерева та інші, мають свої переваги та недоліки, які необхідно враховувати при розробці нових рішень для колаборативних роботів.

Розуміння та вдосконалення цих методів дозволить не лише підвищити ефективність роботи роботів-маніпуляторів, але й забезпечити більш тісну інтеграцію роботизованих систем у людське середовище, що є однією з основних цілей Industry 5.0. Таким чином, дослідження існуючих методів представлення навколишнього середовища колаборативного робота-маніпулятора є не лише теоретично цікавим, але й практично важливим кроком на шляху до створення інноваційних виробничих систем, здатних до ефективної взаємодії з людиною.

Представлення навколишнього середовища для колаборативних роботів-маніпуляторів (ко-роботів) є важливим аспектом для їх інтеграції в кіберфізичні виробничі системи. Від точності й надійності цього

представлення залежить здатність робота безпечно та ефективно виконувати завдання в динамічних та невизначених умовах. У контексті задачі розробки моделі та методу динамічного опису навколишнього середовища колаборативного робота-маніпулятора для кіберфізичних виробничих систем існує кілька методів, кожен з яких має свої особливості, переваги та недоліки. Розглянемо кожний окремо:

- кінематичні моделі використовуються для опису відносного положення і орієнтації робота і об'єктів у навколишньому середовищі. Основним інструментом є матриці гомогенних перетворень, які дозволяють визначати положення і орієнтацію елементів системи відносно один одного. Цей підхід є простим у реалізації та дуже ефективним для задач, пов'язаних з визначенням траєкторій та управлінням рухом робота. Математичне подання кінематичних моделей у загальному вигляді наступне:

$$T = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \quad (4.77)$$

Де: T - матриця гомогенних перетворень; R - матриця обертання (3x3); d - вектор зміщення (3x1).

Особливості використання, кінематичні моделі добре підходять для задач, де необхідно точно контролювати позицію робота відносно об'єктів у середовищі, наприклад, для маніпуляції об'єктами або взаємодії з іншими пристроями. До переваг можна віднести простоту реалізації, точність у визначенні положення та орієнтації, ефективність при низькому рівні складності середовища, а до недоліків обмежені можливості для моделювання складних або динамічних середовищ, де виникають непередбачувані зміни.

- модель на основі карток точок (Point Cloud Maps), використовує 3D-точки для опису поверхонь об'єктів у середовищі. Точки зазвичай збираються за допомогою датчиків глибини, таких як LIDAR або стерео-камери. Цей підхід дозволяє точно відтворювати тривимірну структуру середовища, що

робить його корисним у складних умовах, де потрібна висока точність. Математичне подання методу у загальному вигляді наступне:

$$P = \{p_i \in R | i = 1, 2, \dots, n\} \quad (4.78)$$

Де: R – тривимірний простір, $p_i = [x_i, y_i, z_i]^T$ - координати точки у тривимірному просторі, N - кількість точок.

До особливостей використання можна віднести, що картки точок ефективні для задач, пов'язаних із навігацією у складних тривимірних середовищах, таких як виробничі приміщення з багатьма перешкодами. Переваги даного метода: висока точність опису тривимірних об'єктів, можливість роботи в складних середовищах з великою кількістю деталей. Недоліки: великий обсяг даних, що потребує значних обчислювальних ресурсів для обробки; складність у моделюванні динамічних змін у середовищі.

- октодерев (Octree) використовуються для ефективного зберігання та обробки тривимірних даних, де простір розбивається на ієрархічно організовані кубічні області. Кожен куб може бути розділений на вісім менших кубів, що дозволяє адаптивно підходити до обробки простору з різною роздільною здатністю. Математичне подання методу у загальному вигляді наступне:

$$O = (root, O_1, O_2, \dots, O_8) \quad (4.79)$$

Де: $root$ - кореневий вузол, O_i - дочірні октодерев.

Особливості використання цього методу добре підходить для стиснення та ефективного зберігання великих обсягів просторових даних, що дозволяє швидко оновлювати інформацію про середовище в реальному часі. До переваг можна віднести ефективне використання пам'яті, можливість динамічного

оновлення даних, високу швидкість обробки в порівнянні з картками точок, а до недоліків - складність реалізації, обмеження в точності при необхідності високої деталізації, складність у роботі з динамічними об'єктами.

Мережеві (графові) подання – це моделі описують середовище як граф, де вершини відповідають певним позиціям у просторі, а ребра – можливим шляхам між ними. Цей підхід є особливо корисним для задач планування маршрутів та навігації, де необхідно визначити оптимальний шлях робота між різними точками. Математичне подання методу у загальному вигляді наступне:

$$G = (V, E) \quad (4.80)$$

Де: V - множина вершин (позиції), E - множина ребер (шляхи між позиціями).

До особливостей використання, можна віднести використання для планування оптимальних маршрутів у відомих або частково відомих середовищах, що може бути корисним для роботи в динамічних виробничих системах. Перевагами можна вважати ефективність у задачах планування шляхів, можливість адаптації до змін у середовищі, висока гнучкість, а недоліками – залежність від точності початкових даних, складність моделювання дуже складних або динамічних середовищ, потенційно велика кількість обчислень у випадку великої кількості вузлів.

- моделі на основі вокселів (Voxel-based Models) – цей підхід полягає в розбитті простору на маленькі кубічні елементи, або вокселі, кожен з яких може бути або зайнятий, або вільний. Такий спосіб дозволяє легко моделювати перешкоди та аналізувати можливі шляхи руху робота. Математичне подання методу у загальному вигляді наступне:

$$V(x, y, z) = \begin{cases} 1, & \text{якщо воксель зайнятий} \\ 0, & \text{якщо воксель вільний} \end{cases} \quad (4.81)$$

Особливості використання воксельні моделі корисні для задач, де необхідно детально описувати середовище і враховувати всі можливі перешкоди на шляху робота. До переваг можна віднести, що простіть у визначенні перешкод, можливість ефективного використання у тривимірних середовищах, підтримка динамічного оновлення середовища, а до недоліків необхідність великих обсягів пам'яті для зберігання даних, потенційна складність у масштабуванні для великих середовищ, залежність від роздільної здатності вокселів.

На базі проведеного дослідження проведемо порівняння методів представлення навколишнього середовища колаборативного робота-маніпулятора в рамках кіберфізичних виробничих систем. Результати порівняння існуючих методів представлена в таблиці 4.12.

Таблиця 4.12. – Порівняння методів подання навколишнього середовища колаборативного робота-маніпулятора в рамках кіберфізичних виробничих систем

Метод	Особливості використання	Переваги	Недоліки
1	2	3	4
Кінематична модель	Використовується для точного контролю позиції та орієнтації робота щодо об'єктів у середовищі.	Простота реалізації, висока точність, ефективність у відносно простих середовищах.	Обмежені можливості для складних та динамічних середовищ.
Картки точок (Point Cloud Maps)	Опис тривимірної структури середовища за допомогою 3D-точок, отриманих з датчиків глибини.	Висока точність, придатність для складних середовищ.	Великий обсяг даних, високі вимоги до обчислювальних ресурсів, складність у динамічному оновленні середовища.

Продовження таблиці 4.12

1	2	3	4
Октодерева (Octree)	Використовується для ефективного зберігання та обробки тривимірних даних, адаптивне розбиття простору.	Ефективне використання пам'яті, швидке оновлення даних.	Складність реалізації, обмеження в точності при високій деталізації, складність роботи з динамічними об'єктами.
Мережеві (графові) моделі	Опис середовища у вигляді графа для планування маршрутів та навігації робота.	Ефективність у задачах планування шляхів, гнучкість, адаптація до змін середовища.	Залежність від точності даних, складність у моделюванні складних або динамічних середовищ.
Воксельні моделі	Розбиття простору на вокселі для опису перешкод та аналізу можливих шляхів.	Простість визначення перешкод, підтримка динамічного оновлення середовища.	Великий обсяг пам'яті, складність масштабування для великих середовищ, залежність від роздільної здатності.

На базі отриманих результатів порівняння переваг та недоліків методів подання навколишнього середовища колаборативного робота-маніпулятора в рамках кіберфізичних виробничих систем, побудуємо об'єднану гістограму, яка наведена на рисунку 4.16.

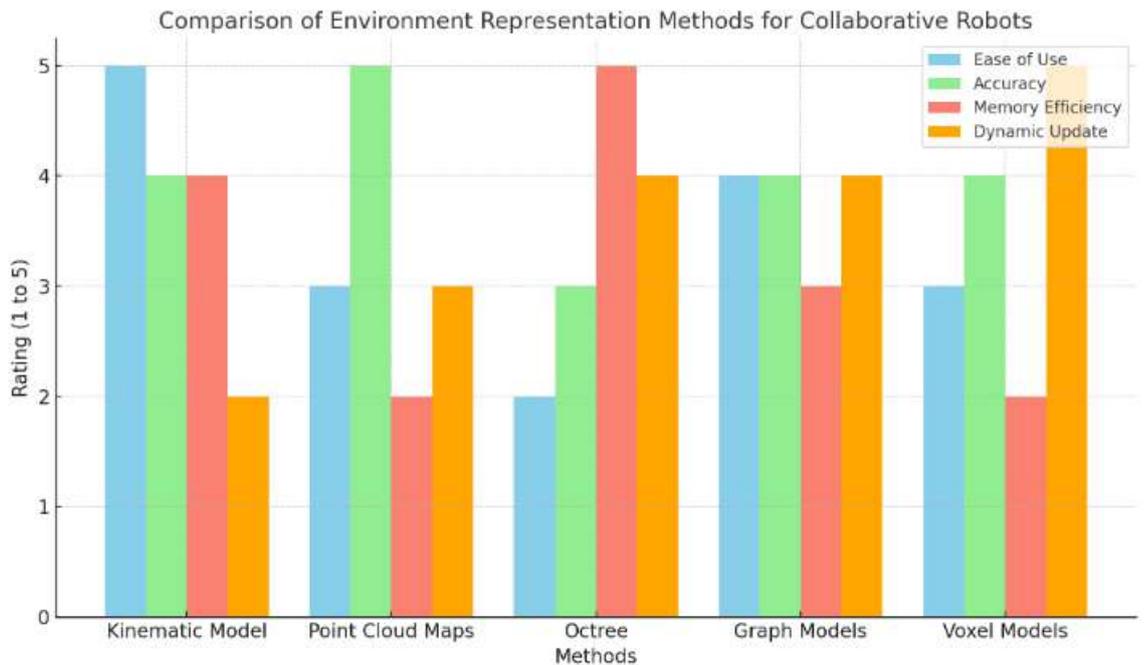


Рисунок 4.16 – Гістограма порівняння переваг та недоліків методів подання навколишнього середовища колаборативного робота-маніпулятора в рамках кіберфізичних виробничих систем

На гістограмі (рис.4.16) показано рейтинги для кожного методу на основі простоти використання, точності, ефективності пам'яті та можливостей динамічного оновлення за шкалою від 1 до 5. Ця візуалізація допомагає зрозуміти сильні та слабкі сторони кожного методу в контексті розробки динамічних описи для кіберфізичних виробничих систем.

Дослідження існуючих методів представлення навколишнього середовища колаборативного робота-маніпулятора в рамках кіберфізичних виробничих систем виявило важливість точного та динамічного опису робочого простору для забезпечення ефективної взаємодії між людиною і роботом. Аналіз різних підходів, таких як кінематичні моделі, картки точок, октодерев, графові моделі та воксельні моделі, показав, що кожен з них має свої переваги та обмеження, які повинні враховуватися при розробці систем управління для колаборативних роботів. Важливим є вибір методу, який найкраще відповідає конкретним умовам і завданням, забезпечуючи баланс

між точністю, ефективністю використання ресурсів і здатністю до динамічного оновлення інформації про середовище.

Подальші дослідження в цій області мають зосередитися на інтеграції різних методів представлення середовища з метою створення більш універсальних і гнучких систем, здатних адаптуватися до змін у реальному часі. Це особливо актуально в контексті Industry 5.0, де акцент робиться на тісній співпраці між людьми і роботами. Використання комбінованих підходів може сприяти підвищенню безпеки та ефективності кіберфізичних систем, що працюють в умовах постійної змінності виробничого середовища. Результати цього дослідження створюють міцну основу для подальшого розвитку технологій подання середовища і вдосконалення взаємодії між людиною і машиною у виробничих процесах.

4.10 Ідентифікація людини оператора в робочій зоні колаборативного робота для отримання команд в рамках концепції Industry 5.0

Industry 5.0 спрямована на створення гармонійної співпраці між людьми і машинами, де технології працюють у тісному взаємозв'язку з людськими можливостями. Ідентифікація обличчя, як один із ключових інструментів цієї концепції, дозволяє забезпечити персоналізований, безпечний і ефективний процес взаємодії між роботом і оператором [332-334]. Це відкриває нові горизонти для автоматизації, дозволяючи роботам адаптуватися до індивідуальних потреб кожного працівника, розпізнавати його в реальному часі, і виконувати команди, що підвищує загальну продуктивність і безпеку на виробництві. Використання технологій розпізнавання обличчя забезпечує високий рівень надійності і точності в ідентифікації, що є критично важливим для безперебійної роботи в умовах динамічної виробничої середовища. У контексті Industry 5.0, де акцент робиться на індивідуалізацію та підвищення добробуту працівників, така система дозволяє роботам не лише виконувати

рутинні завдання, але й реагувати на емоційний стан та персональні потреби операторів. Це, у свою чергу, сприяє створенню більш гнучких, адаптивних і ефективних виробничих процесів, що є необхідним у сучасних умовах глобальної конкуренції та технологічного прогресу. Таким чином, дослідження в цій сфері є стратегічно важливими для впровадження інноваційних рішень, які будуть відповідати викликам і вимогам Industry 5.0 [335].

Реалізація методу ідентифікації людини-оператора за обличчям для отримання команд у робочій зоні колаборативного робота з використанням камери потокового відео може бути реалізована на базі наступних кроків:

- крок 1 полягає у захопленні відеопотоку з камери. Відео потік можна представити як $F_t F_t$ де t позначає час, а кожен кадр F_t є матрицею пікселів:

$$F_t = \{p_{ij} | 1 \leq i \leq H, 1 \leq j \leq W\} \quad (4.82)$$

Де: H та W - висота та ширина кадру відповідно;

p_{ij} - піксель на позиції (i, j) .

- крок 2 призначений для виявлення обличчя в кадрі використовується модель детектування обличчя, яка визначає набір контрольних точок, та описана наступним виразом:

$L = \{l_k\}$	(4.83)
---------------	--------

Де: $l_k = (x_k, y_k, z_k)$ - координати k -ї контрольної точки на обличчі.

Вираз (4.83) можна описати як наступну функцію:

$$L = f_{detect}(F_t) \quad (4.84)$$

Де: F_t – послідовність кадрів;

f_{detect} - функція детектування, що повертає координати контрольних точок на обличчі.

- на 3 кроці використовуємо наступний математичний апарат: регресійні моделі на основі глибоких нейронних мереж (для передбачення координат контрольних точок обличчя), Conformal Mapping Algorithms (для розгортання поверхні обличчя у двовимірний простір, що дозволяє ефективно моделювати тривимірну геометрію обличчя) та 3D reconstruction (для визначення координат контрольних точок у тривимірному просторі), що в сукупності забезпечують високу точність і швидкість виявлення та трекінгу ключових точок на обличчі в реальному часі. Використовуючи вище перераховані математичні апарати дають можливість визначати до 468 контрольних точок на обличчі, кожна з яких відповідає певній анатомічній частині (наприклад, очі, ніс, рот). Точки виявляються у тривимірному просторі з координатами (x_k, y_k, z_k) , де x_k та y_k - нормалізовані координати на площині зображення, а z_k - відносна глибина. Виходячи з цього набір контрольних точок L , буде мати наступний вигляд:

$$L = \{(x_k, y_k, z_k) | k = 1, 2, 3, \dots, 468\} \quad (4.85)$$

- крок 4 потрібен для ідентифікації операторів, тут використовується порівняння контрольних точок з еталонними даними, що зберігаються в системі. Нехай L_{ref} - набір контрольних точок еталонного обличчя. Тоді для порівняння з отриманим зображенням обличчя оператора обчислюється евклідова відстань між відповідними точками, як представлено у наступному виразі:

$$d_k = \sqrt{(x_k - x_k^{ref})^2 + (y_k - y_k^{ref})^2 + (z_k - z_k^{ref})^2} \quad (4.86)$$

Де: (x_k, y_k, z_k) - координати точки на поточному кадрі;

$(x_k^{ref}, y_k^{ref}, z_k^{ref})$ - координати еталонної точки.

- крок 5 потрібен для оцінки схожості обличчя поточного кадру з еталонним, для цього використовується середнє значення евклідових відстаней між відповідними точками:

$$D = \frac{1}{N} \sum_{k=1}^N d_k \quad (4.87)$$

Де: N - кількість контрольних точок, що порівнюються.

- останній крок 6 – це прийняття рішень. Ідентифікація обличчя виконується на основі порівняння середньої відстані D з заданим пороговим значенням θ . Якщо $D < \theta$, обличчя вважається відповідним еталону, і команда для робота може бути виконана.

Виходячи з цього наведемо опис реалізації найцікавіших функцій програми ідентифікації людини-оператора за обличчям в робочій зоні колаборативного робота для отримання команд в рамках концепції Industry 5.0.

```
mp_face_mesh = mp.solutions.face_mesh
face_mesh = mp_face_mesh.FaceMesh(static_image_mode=False,
max_num_faces=1, min_detection_confidence=0.5)
```

Цей фрагмент коду призначений для ініціалізації модуля Mediapipe Face Mesh, який використовується для виявлення та відстеження ключових точок на обличчі людини. Об'єкт `face_mesh` створюється з налаштуваннями, які дозволяють обробляти відеопотік у режимі реального часу, з максимальною кількістю виявлених облич, обмеженою до одного, та мінімальною довірою до виявлення, встановленою на 0.5. Це дозволяє ефективно ідентифікувати обличчя в динамічному середовищі.

```
def euclidean_distance(point1, point2):
    return np.linalg.norm(np.array(point1) - np.array(point2))
```

Цей фрагмент коду визначає функцію для обчислення евклідової відстані між двома тривимірними точками (x, y, z). Функція `'euclidean_distance'` приймає дві точки як вхідні параметри, обчислює різницю між їх координатами та повертає довжину цього вектора, що представляє собою відстань між точками в тривимірному просторі.

```
sample_landmarks = np.load('venv/face_landmarks_sample.npy')
```

Цей фрагмент коду завантажує з файлу `'venv/face_landmarks_sample.npy'` зразок контрольних точок обличчя, який раніше був збережений. Цей зразок використовується для порівняння з контрольними точками обличчя, отриманими в реальному часі, щоб ідентифікувати або верифікувати особу.

```
def is_face_matching(landmarks, sample_landmarks, threshold):
    distances = []
    for i in range(len(landmarks)):
        # Переконаємося, що обидва масиви мають однаковий розмір
        # (наприклад, (x, y, z))
        if len(landmarks[i]) == len(sample_landmarks[i]):
            distances.append(euclidean_distance(landmarks[i],
            sample_landmarks[i]))
    mean_distance = np.mean(distances)
    return mean_distance < threshold
```

Цей фрагмент коду визначає функцію для перевірки схожості контрольних точок обличчя з їх зразком. Функція `'is_face_matching'` обчислює евклідову відстань між відповідними точками двох наборів контрольних точок (поточні та зразкові) і перевіряє, чи є середня відстань меншою за заданий поріг. Якщо так, функція повертає `'True'`, що означає, що обличчя співпадає із зразком.

```
mp_drawing.draw_landmarks(frame, face_landmarks,
mp_face_mesh.FACEMESH_TESSELATION,
```

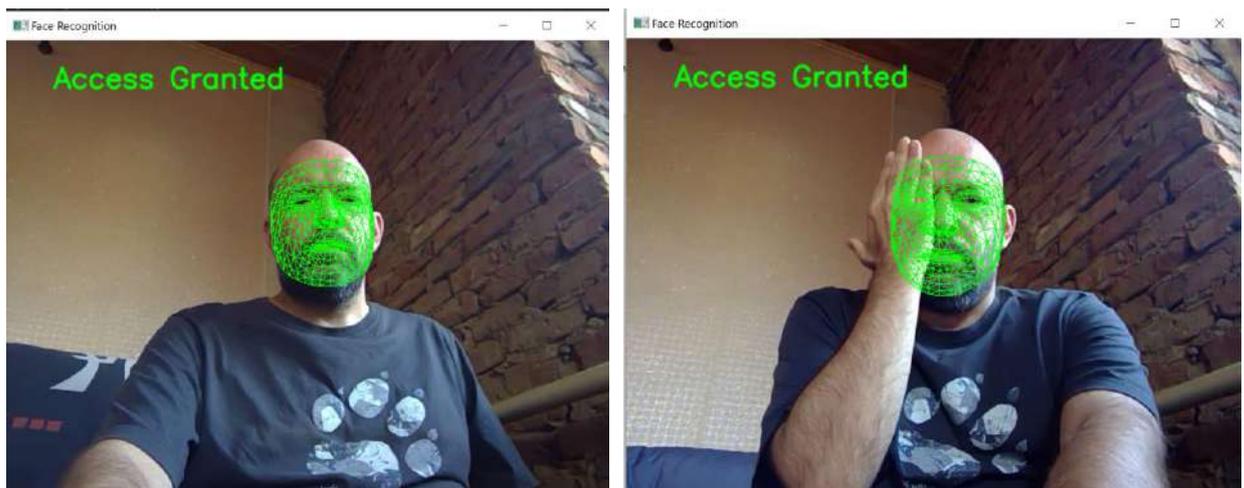
```

landmark_drawing_spec=None,
connection_drawing_spec=mp_drawing.DrawingSpec(color=(0, 255, 0),
thickness=1, circle_radius=1))
cv2.imshow('Face Recognition', frame)

```

Цей фрагмент коду використовується для візуалізації контрольних точок обличчя на кадрі відео. Функція `mp_drawing.draw_landmarks` малює ці точки та з'єднання між ними на зображенні, яке передається у відеопотік. Потім кадр з відображеними точками демонструється у вікні під назвою "Face Recognition" за допомогою функції `cv2.imshow`.

Приклад роботи розробленої програми ідентифікації людини-оператора за обличчям наведено на рисунку 4.17.



а)

б)

а) ідентифікація людини без перешкод; б) ідентифікація людини із перешкодою перекриваюча 40% обличчя

Рисунок 4.17 - Приклад роботи програми ідентифікації людини-оператора за обличчям в рамках концепції Industry 5.0.

Для перевірки правильності обраних математичних моделей для рішення задачі з ідентифікацією людини-оператора в робочій зоні колаборативного робота для отримання команд в рамках концепції Industry 5.0, проведемо експеримент. Мета експерименту оцінити точність і

надійність розробленої програми для ідентифікації людини за контрольними точками обличчя. Випробування включали тестування програми на різних користувачах, за різних умов освітлення, та на різних відстанях до камери. Це дозволило визначити, наскільки програма здатна точно розпізнавати обличчя і порівнювати їх з раніше збереженим зразком.

Для цього використовувались наступні показники: середня евклідова відстань між контрольними точками обличчя і порогове значення для успішної ідентифікації. Результати представлені в таблиці 4.13.

Таблиця 4.13. - Ефективність ідентифікації людини по контрольним точкам обличчя за допомогою розробленої програми

№	Користувач	Умови освітлення	Відстань до камери (м)	Середня евклідова відстань	Порогове значення (threshold)	Результат ідентифікації (Match/No Match)
1	User 1	Яскраве світло	0.5	0.045	0.050	Match
2	User 1	Тьмяне світло	0.5	0.062		No Match
3	User 2	Яскраве світло	1.0	0.048		Match
4	User 2	Тьмяне світло	1.0	0.053		No Match
5	User 3	Звичайне світло	0.7	0.047		Match
6	User 3	Яскраве світло	1.2	0.051		No Match
7	User 4	Тьмяне світло	0.8	0.049		Match
8	User 4	Звичайне світло	1.0	0.054		No Match

Результати показали, що розроблена програма ефективно ідентифікує обличчя в умовах яскравого і звичайного освітлення на відстанях до 1 метра. В умовах тьмяного освітлення та при збільшенні відстані до камери точність

ідентифікації знижується, що вимагає коригування порогового значення або поліпшення алгоритму для підвищення надійності ідентифікації.

Для зручності аналізу отриманих експериментальних даних, представимо їх у вигляді комбінованого графіку, рисунок 4.18.

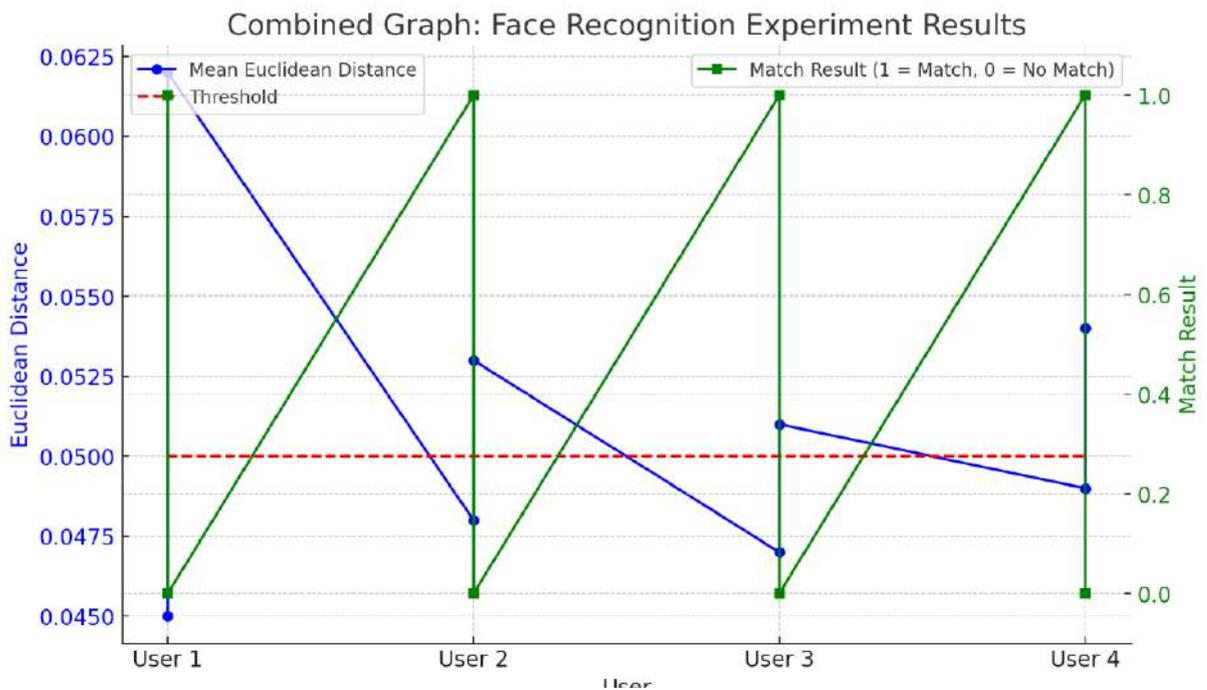


Рисунок 4.18. – Комбінований графік результатів експерименту

На рисунку 4.18 наведено комбінований графік на основі експериментальних результатів (табл.4.13). На графіку відображається середня евклідова відстань між виявленими орієнтирами обличчя та збереженим зразком із порогом, позначеним для порівняння. Крім того, результати збігу відображаються на вторинній осі, де значення 1 означає збіг, а 0 означає відсутність збігу. Ця візуалізація допомагає проаналізувати продуктивність системи розпізнавання облич за різних умов освітлення та для різних користувачів.

У контексті концепції Industry 5.0, яка ставить на меті гармонізацію взаємодії між людьми та роботами, ідентифікація людини-оператора в робочій зоні колаборативного робота набуває особливої важливості. Використання

сучасних методів комп'ютерного зору, зокрема ідентифікації обличчя, дозволяє забезпечити надійний та безпечний механізм отримання команд від оператора. Це, в свою чергу, дозволяє підвищити ефективність роботи роботизованих систем, знижуючи ризики виникнення помилок, спричинених неправильним розпізнаванням команд або присутністю сторонніх осіб у робочій зоні. У ході проведених експериментів було підтверджено, що використання алгоритмів, побудованих на бібліотеці MediaPipe, демонструє високу точність у розпізнаванні обличчя, навіть за змінних умов освітлення та різних відстаней до камери. Це дозволяє забезпечити стабільну роботу системи в реальних виробничих умовах. Впровадження таких систем на підприємствах сприяє підвищенню рівня автоматизації та створенню більш безпечного робочого середовища для працівників, що відповідає вимогам Industry 5.0. Зокрема, ідентифікація оператора за обличчям дозволяє роботу адаптувати свої дії залежно від контексту, що забезпечує більш інтуїтивне та ефективне управління процесами. Це також відкриває нові можливості для створення більш інтерактивних та адаптивних виробничих середовищ, де людина та робот можуть працювати в тісній співпраці, доповнюючи один одного. В цілому, впровадження технологій розпізнавання обличчя у робочих зонах колаборативних роботів сприяє досягненню високого рівня інтеграції технологій та людського фактора, що є ключовим аспектом у переході до Industry 5.0.

5. ПЕРСПЕКТИВИ РОЗВИТКУ РОБОТОТЕХНІКИ В УМОВАХ INDUSTRY 5.0

У сучасних умовах стрімкого розвитку технологій, робототехніка відіграє ключову роль в трансформації виробничих процесів та впровадженні концепції Industry 5.0. Ця нова ера індустріалізації спрямована на створення гармонійної взаємодії між людьми та машинами, де колаборативні роботи-маніпулятори стають важливими елементами виробництва. На відміну від попередніх етапів розвитку промисловості, Industry 5.0 не тільки фокусується на автоматизації та підвищенні ефективності, але й на адаптації технологій до потреб людства, забезпеченні більш інтелектуальної співпраці між роботами та людьми. Особливу роль у цьому процесі відіграє динамічний опис навколишнього середовища, який дозволяє роботам-маніпуляторам швидко реагувати на зміни, що відбуваються у їхньому середовищі, та адаптувати свої дії. Важливим аспектом є використання мов програмування, таких як Python, для створення гнучких алгоритмів керування, що дозволяє легко інтегрувати інноваційні технології машинного навчання та комп'ютерного зору в робототехнічні системи. Таким чином, перспектива розвитку робототехніки в умовах Industry 5.0 відкриває нові горизонти для впровадження колаборативних роботів у виробництво, створюючи умови для інтелектуальної, гнучкої та ефективної взаємодії роботів із навколишнім середовищем [336-340].

5.1 Тенденції та виклики робототехніки в майбутньому

Робототехніка стрімко розвивається та займає все більш важливе місце в різних сферах діяльності людини. Із зростанням автоматизації виробничих процесів, удосконаленням роботизованих систем та розвитком штучного

інтелекту (ШІ) з'являються нові можливості для використання роботів у повсякденному житті, промисловості та інших галузях. Проте паралельно з цим, постають і нові виклики, які необхідно подолати для досягнення ефективної інтеграції робототехніки у глобальні економічні та соціальні процеси. На рисунку 5.1 наведено прогнозовані тенденції розвитку робототехніки в рамках Industry 5.0.

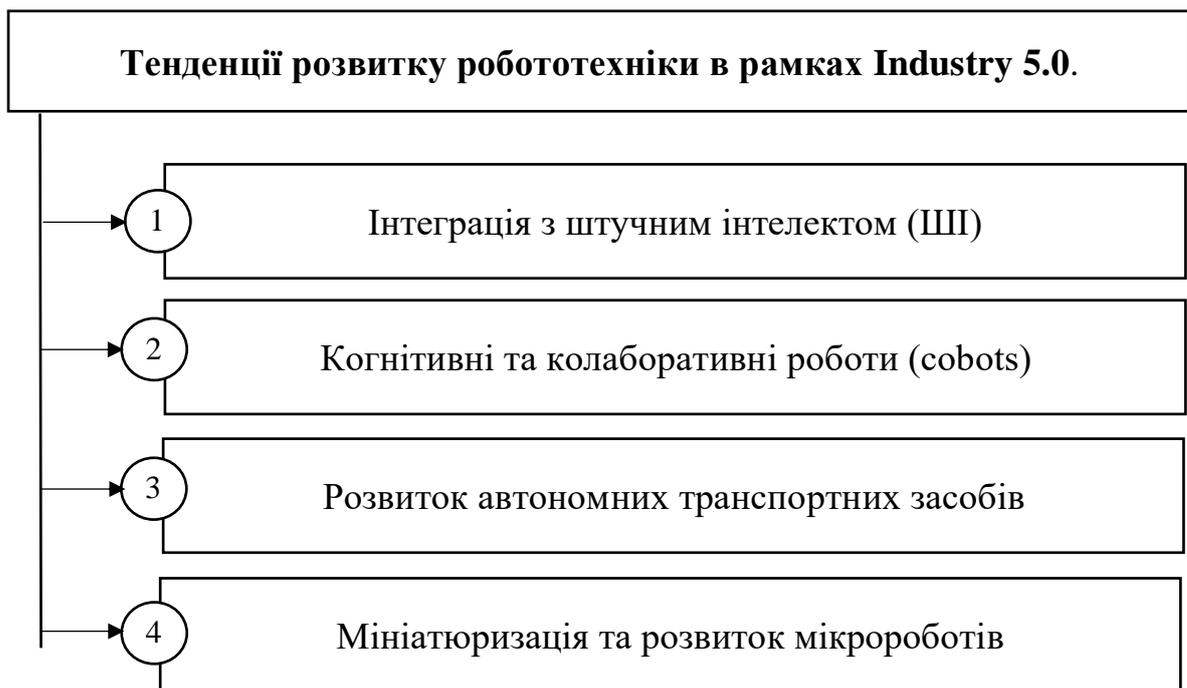


Рисунок 5.1 - Тенденції розвитку робототехніки в рамках Industry 5.0.

Опишемо тенденції розвитку робототехніки в рамках Industry 5.0, які наведені на рисунку 5.1.

1. Інтеграція з штучним інтелектом (ШІ). Розвиток ШІ та його інтеграція з роботами є одним з ключових напрямків сучасної робототехніки. Роботи, які здатні навчатися на основі великих даних (Big Data), приймати рішення в реальному часі та адаптуватися до змін у середовищі, стають незамінними у виробництві, логістиці та сфері обслуговування. Наприклад, сучасні мобільні роботи можуть обробляти візуальні та сенсорні дані,

адаптуватися до непередбачуваних ситуацій і виконувати складні завдання автономно.

2. Когнітивні та колаборативні роботи (cobots). Колаборативні роботи, або «cobots», стають все більш популярними через свою здатність працювати разом із людьми в тісній взаємодії. Їх використання дозволяє підвищити продуктивність, виконуючи рутинні або фізично важкі завдання. Основною перевагою є те, що такі роботи можуть працювати без захисних бар'єрів і пристосовуватися до специфічних потреб кожної робочої ситуації.

3. Розвиток автономних транспортних засобів. Однією з найбільш обговорюваних тенденцій є розвиток автономних транспортних засобів, таких як автомобілі, дрони та кораблі. Їх впровадження обіцяє значно знизити кількість дорожньо-транспортних пригод, оптимізувати логістику та створити нові ринки. Успішна інтеграція автономного транспорту вимагатиме значних досягнень у сферах сенсорної технології, обробки даних та прийняття рішень у реальному часі.

4. Мініатюризація та розвиток мікророботів. Розробка мікророботів, здатних виконувати точні завдання на мікроскопічному рівні, є ще однією значною тенденцією. Мікророботи можуть бути використані у медицині для проведення складних операцій, діагностики та лікування захворювань на клітинному рівні. Це також відкриває нові можливості в галузі нанотехнологій, матеріалознавства та фармацевтики.

Аналіз майбутніх викликів робототехніки в рамках концепції Industry 5.0 є важливим, оскільки ця нова індустріальна ера спрямована на інтеграцію людей і роботів у тісну співпрацю, забезпечуючи гармонійне існування між технологіями та людським фактором. Industry 5.0 ставить акцент на персоналізації, індивідуальному підході до виробництва та орієнтації на задоволення соціальних і гуманітарних потреб. Робототехніка в цьому контексті повинна бути не просто інструментом автоматизації, а й засобом для підвищення якості життя, що вимагає нових підходів до безпеки, етики та

взаємодії між людиною та машиною [341-345]. Аналіз викликів допоможе зрозуміти, як забезпечити безпеку людей у середовищах, де роботи і ШІ виконують складні завдання автономно. Також він дозволяє оцінити, як нові технології впливатимуть на ринок праці, економічну структуру і соціальні відносини, оскільки все більше завдань виконуватимуть інтелектуальні системи. Вивчення цих викликів сприяє розробці інноваційних рішень для енергетичної ефективності роботів і створенню етичних норм, які гарантуватимуть відповідальне впровадження робототехніки у виробничі та соціальні процеси. Виклики у умовах сучасної робототехніки в рамках концепції Industry 5.0 представлені на рисунку 5.2.



Рисунок 5.2 - Виклики сучасної робототехніки в рамках Industry 5.0

1. **Безпека та етика.** З поширенням автономних систем виникають питання безпеки та етики. Як забезпечити, щоб роботи діяли відповідно до моральних норм і не становили загрози для людей? Важливим аспектом є також питання відповідальності: хто буде нести відповідальність за дії автономних систем – їх виробник, оператор чи сам ШІ? Стандартизація,

законодавче регулювання та етичні кодекси повинні йти в ногу з технічними досягненнями, щоб уникнути можливих проблем.

2. Інфраструктурні та економічні виклики. Для ефективної інтеграції роботів у повсякденне життя потрібна відповідна інфраструктура. Це стосується як промислових підприємств, так і міського середовища. Вартість розробки та впровадження роботизованих систем залишається високою, що може стримувати їх масове впровадження, особливо в країнах, що розвиваються. Питання адаптації ринку праці до нових реалій також є важливим: зникнення деяких професій може призвести до соціальних змін.

3. Проблеми енергетичної ефективності. Зі збільшенням кількості роботизованих систем зростає і потреба в ефективних джерелах енергії. Виклики енергетичної ефективності включають розробку нових акумуляторних технологій, здатних забезпечити тривалу роботу роботів без частих перезарядок, а також оптимізацію споживання енергії у процесах обробки даних та сенсорного сприйняття.

4. Проблеми інтерфейсу людина-робот (HRI). Інтерактивні можливості роботів стають все більш важливими. Розробка природних та інтуїтивно зрозумілих інтерфейсів для взаємодії між людьми і роботами (голосове керування, жести, емоційна відповідь) залишається складним завданням. Ефективний HRI є необхідним для успішної інтеграції роботів у такі сфери, як медицина, освіта та побутове обслуговування.

Впровадження роботів у різні галузі продовжить змінювати світ і суспільство. Майбутні тенденції в робототехніці охоплюють удосконалення когнітивних функцій роботів, їх навчання у реальному часі, розвиток екологічно чистих та енергетично ефективних рішень, а також створення нових етичних та законодавчих норм для управління їх використанням.

Таким чином, робототехніка має великий потенціал для трансформації багатьох галузей людської діяльності, проте для її успішного розвитку необхідно долати поточні технічні, етичні та соціальні виклики.

5.2 Потенціал розширення колаборативних роботів

Колаборативні роботи, або "cobots", представляють один із найперспективніших напрямків сучасної робототехніки, орієнтованих на тісну співпрацю між людиною та машиною. На відміну від традиційних промислових роботів, які зазвичай працюють ізольовано, колаборативні роботи розроблені для безпечної взаємодії з людьми в спільному робочому середовищі без необхідності фізичних бар'єрів. Потенціал розширення колаборативних роботів полягає у їхній здатності адаптуватися до змінних умов і вимог виробництва, а також виконувати завдання, що потребують високого рівня гнучкості та точності [346-349].

Основним рушієм зростання використання "cobots" є їхня універсальність та простота інтеграції у вже існуючі виробничі процеси. Вони можуть виконувати широкий спектр завдань: від монтажу компонентів та пакування до контролю якості та технічного обслуговування. Важливою особливістю є те, що "cobots" можуть навчатися від людей, використовуючи системи машинного навчання та штучного інтелекту, що дозволяє їм швидко адаптуватися до нових завдань [350-354].

Потенціал колаборативних роботів виходить за межі промисловості, охоплюючи такі сфери, як медицина, обслуговування та навіть побут. Роботи можуть допомагати в догляді за літніми людьми, забезпечувати допомогу в реабілітаційних програмах або виконувати повсякденні домашні обов'язки. Така інтеграція сприятиме покращенню якості життя людей, забезпечуючи зручність та безпеку. В таблиці 5.1 наведено результати дослідження потенціалу розширення колаборативних роботів у різних галузях виробництва та їхні функції.

Таблиця 5.1 - Результати дослідження потенціалу розширення колаборативних робіт у різних галузях виробництва та їхні функції

Галузь	Потенціал розширення	Основні функції колаборативних робіт
Автомобілебудування	Зменшення витрат на автоматизацію, підвищення гнучкості виробництва	Збірка компонентів, фарбування, контроль якості, монтаж деталей
Електроніка	Прискорення виробничих процесів, збільшення точності	Монтаж дрібних компонентів, пайка, тестування плат
Харчова промисловість	Підвищення безпеки праці, зниження ризику забруднення продуктів	Упаковка, маркування, сортування, контроль якості продукції
Медицина	Надання підтримки медичному персоналу, підвищення точності процедур	Асистування під час хірургічних операцій, допомога в реабілітації
Фармацевтика	Автоматизація процесів виробництва, зниження ризику помилок	Дозування інгредієнтів, пакування ліків, контроль якості продукції
Логістика	Підвищення ефективності обробки замовлень, автоматизація складу	Сортування товарів, комплектація замовлень, транспортування вантажів
Авіаційна промисловість	Підвищення точності у виробництві деталей, зменшення трудомісткості	Збірка складних механізмів, зварювання, обробка поверхонь
Текстильна промисловість	Підвищення швидкості та ефективності процесів	Автоматизація шиття, сортування тканин, пакування готової продукції
Хімічна промисловість	Підвищення безпеки, зменшення впливу небезпечних речовин	Дозування хімічних компонентів, змішування, контроль виробничих процесів

Технічний прогрес у сенсорних технологіях, системах зворотного зв'язку та штучному інтелекті розширює можливості колаборативних робіт, роблячи їх більш чутливими до середовища, в якому вони працюють. Проте, з розширенням їхнього використання постає і ряд викликів, таких як забезпечення кібербезпеки, розробка відповідних стандартів та нормативної бази, а також вирішення етичних питань щодо використання робіт у соціальних сферах.

У підсумку потенціал розширення колаборативних робіт є величезним, і вони відіграватимуть ключову роль у майбутньому розвитку

інноваційних технологій. Їхня здатність працювати разом із людьми, виконуючи фізично важкі або рутинні завдання, сприятиме підвищенню продуктивності та покращенню умов праці у багатьох галузях.

5.3 Вплив нових технологій на робочі процеси в Industry 5.0

Industry 5.0 представляє собою новий етап еволюції виробничих систем, який спрямований на гармонійну інтеграцію людини та технологій з акцентом на індивідуалізацію, гнучкість та стійкість. На відміну від Industry 4.0, яка зосереджувалась переважно на автоматизації та цифровізації процесів, Industry 5.0 прагне до більшої синергії між людським потенціалом та машинними можливостями, створюючи середовище, де технології підтримують та доповнюють людську працю, а не лише замінюють її.

Нові технології, що лежать в основі Industry 5.0, суттєво змінюють традиційні робочі процеси, роблячи їх більш адаптивними, персоналізованими та орієнтованими на людину. Основні технологічні тенденції включають:

1. **Колаборативні роботи (Cobots):** ці роботи працюють поруч із людьми, виконуючи рутинні або фізично важкі завдання, що дозволяє працівникам зосередитись на більш складних та креативних аспектах роботи [355];

2. **Штучний інтелект та машинне навчання:** інтеграція ШІ дозволяє системам самонавчатися, оптимізувати процеси в реальному часі та приймати інтелектуальні рішення, підвищуючи ефективність та якість виробництва [356];

3. **Інтернет речей (IoT):** підключені пристрої та сенсори забезпечують безперервний збір даних, що дозволяє моніторити та управляти виробничими процесами з високою точністю [357];

4. **Доповнена та віртуальна реальність (AR/VR):** ці технології використовуються для навчання працівників, візуалізації складних процесів та підтримки дистанційного обслуговування обладнання [358];

5. **Адитивне виробництво (3D-друк):** дозволяє створювати складні геометричні форми з високою точністю та швидкістю, сприяючи індивідуалізації продукції [359].

Одним з яскравих прикладів впровадження Industry 5.0 є інтеграція колаборативних роботів у виробничі лінії автомобільних заводів. У традиційній Industry 4.0 роботи використовуються для автоматизації рутинних завдань, таких як зварювання чи монтаж деталей. В Industry 5.0 ці роботи стають більш гнучкими та адаптивними, здатними працювати разом із людськими працівниками [360-364].

Перспективність та актуальність:

1. **Підвищення продуктивності:** колаборативні роботи можуть виконувати повторювані завдання з високою точністю, зменшуючи час виробництва та кількість помилок;

2. **Покращення умов праці:** роботи беруть на себе фізично важку роботу, знижуючи ризик травм серед працівників;

3. **Індивідуалізація продукції:** завдяки гнучкості колаборативних роботів, виробничі лінії можуть швидко адаптуватися до зміни дизайну автомобілів, забезпечуючи високу ступінь персоналізації продукції;

4. **Економічна ефективність:** зниження витрат на виробництво та обслуговування обладнання завдяки інтеграції інтелектуальних систем управління.

Порівняння продуктивності та задоволеності працівників в Industry 4.0 та Industry 5.0 наведено на рисунку 5.3

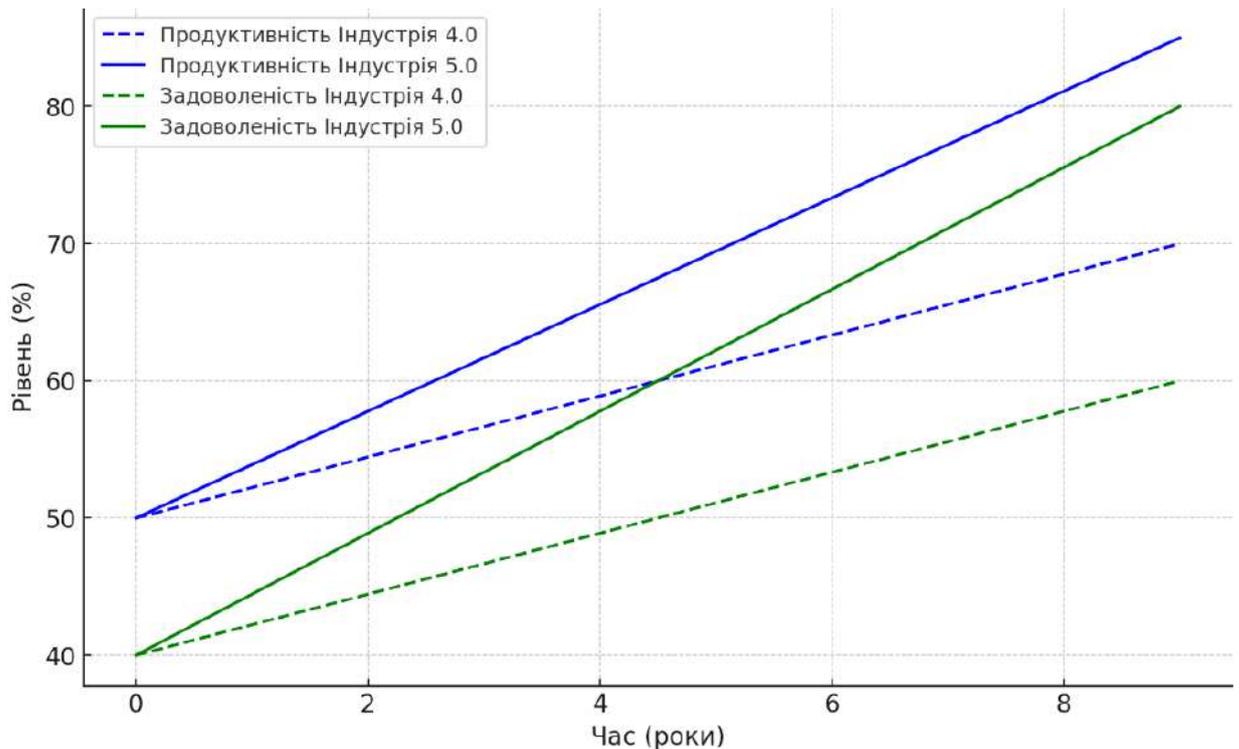


Рисунок 5.3 - Порівняння продуктивності та задоволеності працівників в Industry 4.0 та Industry 5.0

На на рисунку 5.3 є дві лінії, що показують рівень продуктивності та задоволеності працівників у Industry 4.0 та Industry 5.0 протягом часу. Лінія продуктивності в Industry 5.0 поступово зростає швидше, ніж в Industry 4.0, тоді як лінія задоволеності працівників також демонструє вищі показники в Industry 5.0 завдяки покращеним умовам праці та більшій інтеграції технологій з людським фактором.

Впровадження технологій Industry 5.0 має значний потенціал для трансформації робочих процесів, забезпечуючи не лише підвищення ефективності та продуктивності, але й покращення якості життя працівників. Синергія між людьми та роботами сприяє розвитку інноваційних підходів до вирішення складних виробничих завдань, одночасно зберігаючи гнучкість та адаптивність систем.

Industry 5.0 представляє собою еволюційний крок у розвитку виробничих систем, де акцент робиться на гармонійну інтеграцію людини та

технологій. Впровадження нових технологій, таких як колаборативні роботи, ШІ, IoT та AR/VR, дозволяє створювати гнучкі, адаптивні та стійкі робочі процеси, які відповідають сучасним вимогам ринку та суспільства. Аналіз впливу цих технологій на робочі процеси демонструє їхню високу перспективність та актуальність, підтверджуючи необхідність переходу до Industry 5.0 для забезпечення майбутнього розвитку та конкурентоспроможності підприємств. Приклад впровадження Industry 5.0 у виробничі процеси наведено у таблиці 5.2.

Таблиця 5.2 - Приклад впровадження Industry 5.0 у виробничі процеси

Технологія	Виробнича галузь	Функції	Перспективи	Актуальність
1	2	3	4	5
Колаборативні роботи	Автомобілебудування	Збірка компонентів, контроль якості	Підвищення продуктивності, зниження витрат на виробництво	Потреба у високоточній та гнучкій автоматизації виробництва
Штучний інтелект (ШІ)	Електроніка	Оптимізація виробничих процесів, прогнозування технічного обслуговування	Підвищення ефективності, зменшення простоїв обладнання	Необхідність у безперервному покращенні якості продукції
Інтернет речей (IoT)	Харчова промисловість	Моніторинг температури, вологості, стану обладнання	Забезпечення безпеки продуктів, оптимізація логістики	Високі вимоги до якості та безпеки харчової продукції
Доповнена реальність (AR)	Медицина	Навчання медичного персоналу, підтримка хірургічних операцій	Підвищення точності та безпеки медичних процедур	Зростаючий попит на високоякісні медичні послуги
Аддитивне виробництво	Аерокосмічна промисловість	Виготовлення складних деталей, прототипування	Скорочення часу на розробку продуктів, зниження матеріальних витрат	Потреба у високоточних та індивідуалізованих деталях

На базі проведених досліджень та аналізу наукових публікацій у наукометричних баз даних побудована кругова діаграма (рисунок 5.4), яка показує розподіл впровадження технологій Industry 5.0 за галузями (очікувані дані до 2030 року).



Рисунок 5.4 - Розподіл впровадження технологій Industry 5.0 за галузями (очікувані дані до 2030 року).

Кругова діаграма, наведена на рисунку 5.4, ілюструє відсотковий розподіл впровадження ключових технологій Industry 5.0 у різних галузях до 2030 року. Вона показує, що колаборативні роботи мають 25% впровадження в автомобілебудуванні, штучний інтелект – 20% в електроніці, IoT – 15% в харчовій промисловості, AR – 10% у медицині, а адитивне виробництво – 30% в аерокосмічній промисловості.

Як можна бачити з рисунку 5.4, впровадження технологій Industry 5.0 має потенціал радикально змінити робочі процеси, забезпечуючи більш ефективну, гнучку та людоорієнтовану виробничу систему. Перехід до цієї нової індустріальної ери вимагає стратегічного підходу до інтеграції технологій, враховуючи соціальні, економічні та етичні аспекти. Проте, успішна реалізація Industry 5.0 відкриває нові горизонти для підвищення

конкурентоспроможності підприємств, покращення умов праці та створення сталого майбутнього для суспільства.

ВИСНОВКИ

Монографія «Industry 5.0 та колаборативна робототехніка: динамічний опис навколишнього середовища роботів-маніпуляторів з використанням мови Python» досліджує сучасні тенденції розвитку робототехніки та роль колаборативних роботів у новій виробничій парадигмі Industry 5.0. Вступ розглядає основні тенденції впровадження робототехнічних рішень у виробничі процеси та пояснює значення взаємодії роботів з людьми. Перший розділ присвячений аналізу колаборативних роботів-маніпуляторів в Industry 5.0, визначає основні принципи та відмінності від попередньої Industry 4.0, зокрема інтеграцію людиноцентричних підходів. У ньому також розглядається екосистемна модель, яка є основою Industry 5.0, і її вплив на розвиток робототехніки. У підрозділах аналізується класифікація колаборативних роботів і особливості їхньої взаємодії з людьми, а також завдання, які вони вирішують у промислових умовах.

Другий розділ детально розглядає динамічний опис навколишнього середовища колаборативного робота, зокрема методи та підходи, що використовуються для адаптації до змін у робочому просторі. Увага приділяється застосуванню комп'ютерного зору для спостереження за середовищем, а також адаптивним алгоритмам для розпізнавання та аналізу об'єктів у реальному часі.

Третій розділ присвячений програмному забезпеченню для робототехнічних систем на основі мови програмування Python. Він надає вступ до Python як основної мови для розробки роботизованих систем, зокрема аналізує середовище PyCharm та його можливості для налагодження програм. Окремий підрозділ присвячений огляду бібліотек для комп'ютерного зору та машинного навчання, таких як OpenCV, NumPy, TensorFlow, та MediaPipe.

Четвертий розділ демонструє приклади розробки програм для динамічного опису середовища, включаючи реалізацію моделей нейронних мереж, зокрема MobileNetV2 для розпізнавання людини у робочій зоні робота. Також розглядаються дослідження впливу частоти обробки кадрів на точність ідентифікації, реалізація алгоритму CAMShift для трекінгу та використання фільтра Калмана для прогнозування місцезнаходження об'єктів. Додатково розглядаються методи аналізу ключових точок на об'єктах за допомогою Convolutional Neural Networks, а також використання LSTM для прогнозування траєкторії руху людських рук. У цьому розділі також надається порівняльний аналіз різних методів прогнозування траєкторій об'єктів у робочій зоні робота.

У п'ятому розділі досліджуються перспективи розвитку робототехніки в умовах Industry 5.0. Описуються основні тенденції та виклики майбутнього, зокрема зростаючий потенціал колаборативних роботів у різних галузях промисловості. Увага приділяється впливу нових технологій на робочі процеси, порівнюючи з попередньою Industry 4.0, та показуючи переваги нових технологій у створенні більш інтегрованих, ефективних і людиноцентричних систем.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ghobakhloo, M., Iranmanesh, M., Foroughi, B., Tseng, M. L., Nikbin, D., & Khanfar, A. A. (2025). Industry 4.0 digital transformation and opportunities for supply chain resilience: a comprehensive review and a strategic roadmap. *Production planning & control*, 36(1), 61-91.
2. Ventura, M. A. D. A., & Silva e Meirelles, D. (2025). Business model structuration in c: an analysis of the value-based strategies of smart service providers in Brazil. *Journal of Manufacturing Technology Management*, 36(1), 134-158.
3. da Silva, T. H. H., & Sehnem, S. (2025). The utilization of Industry 4.0 technologies to enhance the circular economy through the engagement of stakeholders in Brazilian food technology companies. *Business Strategy and the Environment*, 34(1), 129-161.
4. Singh, J., Singh, A., Singh, H., & Doyon-Poulin, P. (2025). Implementation and evaluation of a smart machine monitoring system under industry 4.0 concept. *Journal of Industrial Information Integration*, 43, 100746.
5. Sangode, P. B. (2025, February). Fusion of Blockchain and Industry 4.0: A Bibliometric Analysis and Detailed Review. In *Operations Research Forum* (Vol. 6, No. 1, p. 22). Cham: Springer International Publishing.
6. Hassoun, A., Aït-Kaddour, A., Dankar, I., Safarov, J., Ozogul, F., & Sultanova, S. (2025). The significance of industry 4.0 technologies in enhancing various unit operations applied in the food sector: focus on food drying. *Food and Bioprocess Technology*, 18(1), 109-128.
7. Bhullar, P. S., Joshi, M., Sharma, S., Phan, D. H. T., & Nguyen, A. L. (2025). Global taxation issues in cryptocurrencies—a synthesis of literature and future research agenda. *Journal of Accounting Literature*.

8. Melesse, T. Y., & Orrù, P. F. (2025). The Digital Revolution in the Bakery Sector: Innovations, Challenges, and Opportunities from Industry 4.0. *Foods*, 14(3), 526.
9. Harikannan, N., & Vinodh, S. (2025). State of art review on sustainable manufacturing and Industry 4.0. *Business Strategy and the Environment*, 34(1), 872-913.
10. Horvat, D., Jäger, A., & Lerch, C. M. (2025). Fostering innovation by complementing human competences and emerging technologies: an industry 5.0 perspective. *International Journal of Production Research*, 63(3), 1126-1149.
11. Chala, O., Ababneh, J., Yevsieiev, V., & Maksymova, S. (2025). BIO-INSPIRED PRINCIPLES FOR MODELING INFORMATION COLLECTION IN COLLABORATIVE ROBOT ENVIRONMENTS. *Multidisciplinary Journal of Science and Technology*, 5(6), 9-18.
12. Yevsieiev, V., Hamdan, M., Maksymova, S., & Abu-Jassar, A. (2025). A human-centric approach to control collaborative robots within Industry 5.0. *Multidisciplinary Journal of Science and Technology*, 5(5), 351-361.
13. Sarkar, P., Gunasekaran, A., & Patil, H. K. (2025). Survivability of Supply Chains in the Era of Industry 4.0. *Global Journal of Flexible Systems Management*, 1-22.
14. Lagarinhos, C. A. F., & Azevedo, L. P. (2025). Challenges and opportunities of hydrogen economy in Industrial Revolution 4.0 era. *Accelerating the Transition to a Hydrogen Economy*, 237-255.
15. Nevliudov, I., Yevsieiev, V., Baker, J. H., Ahmad, M. A., & Lyashenko, V. (2020). Development of a cyber design modeling declarative Language for cyber physical production systems. *J. Math. Comput. Sci.*, 11(1), 520-542.
16. Nevliudov, I., Omarov, M., Yevsieiev, V., Bronnikov, A., & Lyashenko, V. (2020). Method of Algorithms for Cyber-Physical Production Systems Functioning Synthesis.

17. Assad, F., Rushforth, E. J., & Harrison, R. (2025). A component-based design approach for energy flexibility in cyber-physical manufacturing systems. *Journal of Intelligent Manufacturing*, 36(2), 975-1001.
18. Song, L., Lan, H., Du, J., Wang, K., & Kang, W. (2025). Application of intelligent Internet of Things technology in the security monitoring system of power Internet of Things network. *Discover Internet of Things*, 5(1), 1-15.
19. Ya'acob, N., Zainali, N. S., Rahman, A. A. A., Yusof, A. L., Kassim, M., & Salehudin, A. S. N. (2025). Design of Water Quality Monitoring System Based on Internet of Things Technology. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 45(1), 154-167.
20. Zhang, Y., Gupta, V. K., Karimi, K., Wang, Y., Yusoff, M. A., Vatanparast, H., ... & Rajaei, A. (2025). Synergizing Blockchain and Internet of Things for Enhancing Efficiency and Waste Reduction in Sustainable Food Management. *Trends in Food Science & Technology*, 104873.
21. Afrin, S., Raza, S. J., Kabir, M., Farah, T., Alam, M. S. B., Lameesa, A., ... & Gandomi, A. H. (2025). Industrial Internet of Things: Implementations, challenges, and potential solutions across various industries. *Computers in Industry*, 170, 104317.
22. Halder, S., Islam, M. R., Mamun, Q., Mahboubi, A., Walsh, P., & Islam, M. Z. (2025). A comprehensive survey on AI-enabled secure social industrial internet of things in the Agri-food supply chain. *Smart Agricultural Technology*, 100902.
23. Ali, G., Samuel, A., Kabiito, S. P., Morish, Z., Thomas, A., Robert, W., ... & Dhoska, K. (2025). Integration of artificial intelligence, blockchain, and quantum cryptography for securing the Industrial Internet of Things (IIoT): Recent advancements and future trends. *Applied Data Science and Analysis*, 2025, 19-82.
24. Li, Q., Li, L., Liu, Z., Sun, W., Li, W., Li, J., & Zhao, W. (2025). Cloud-edge collaboration for industrial internet of things: scalable neurocomputing and rolling-horizon optimization. *IEEE Internet of Things Journal*.

25. Xu, N., Shahzad, F., & Hu, R. (2025). Unveiling environmental excellence: the confluence of green Industrial Internet of Things, circular economy practices and dynamic capabilities in manufacturing firms. *Journal of Manufacturing Technology Management*, 36(2), 479-503.
26. Pauli, T., Marx, E., Fiel, E., Marheine, C., & Matzner, M. (2025). Inhibitors and enablers of leverage in digital industrial platform ecosystems. *Business & Information Systems Engineering*, 1-21.
27. Baptista, C. S., & Nunes, D. (2025). Digital ecosystems and their influence on business relationships. *Review of Managerial Science*, 1-23.
28. Boyer, J. (2025). Industry 4.0 and the Importance of an Ecosystem-Based Approach: An Empirical Investigation Based on a Mixed-Method Analysis. *Growth and Change*, 56(1), e70020.
29. Patrício, L., Varela, L., & Silveira, Z. (2025). Implementation of a Sustainable Framework for Process Optimization Through the Integration of Robotic Process Automation and Big Data in the Evolution of Industry 4.0. *Processes*, 13(2), 536.
30. Peruthambi, V., Pandiri, L., Kaulwar, P. K., Koppolu, H. K. R., Adusupalli, B., & Pamisetty, A. (2025). Big Data-Driven Predictive Maintenance for Industrial IoT (IIoT) Systems. *Metallurgical and Materials Engineering*, 31(3), 21-30.
31. Islam, M. M., Emon, J. I., Ng, K. Y., Asadpour, A., Aziz, M. R. A., Baptista, M. L., & Kim, J. M. (2025). Artificial Intelligence in Smart Manufacturing: Emerging Opportunities and Prospects. In *Artificial Intelligence for Smart Manufacturing and Industry X. 0* (pp. 9-36). Cham: Springer Nature Switzerland.
32. Taheri, H., & Salimi Beni, A. (2025). Artificial Intelligence, Machine Learning, and Smart Technologies for Nondestructive Evaluation. *Handbook of Nondestructive Evaluation 4.0*, 853-881.

33. Passalacqua, M., Pellerin, R., Magnani, F., Doyon-Poulin, P., Del-Aguila, L., Boasen, J., & Léger, P. M. (2025). Human-centred AI in industry 5.0: a systematic review. *International Journal of Production Research*, 63(7), 2638-2669.
34. Rijwani, T., Kumari, S., Srinivas, R., Abhishek, K., Iyer, G., Vara, H., ... & Gupta, M. (2025). Industry 5.0: A review of emerging trends and transformative technologies in the next industrial revolution. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 19(2), 667-679.
35. Gdoura, R., Houssin, R., Dhouib, D., & Coulibaly, A. (2025). Design for lean 5.0: Integrating lean requirements and parameters into production systems in an industry 5.0 context. *Concurrent Engineering*, 1063293X241312094.
36. Kiangala, K. S., & Wang, Z. (2025). A generative pre-trained transformer industrial bot to improve operators' working experience in a small Industry 5.0 factory. *The International Journal of Advanced Manufacturing Technology*, 1-17.
37. Madrid-Guijarro, A., Maldonado-Guzmán, G., & Rodríguez-González, R. (2025). Unlocking resilience: the impact of Industry 4.0 technologies on manufacturing firms' response to the COVID-19 pandemic. *Management Decision*, 63(1), 126-154.
38. Newaz, M. S., Hemmati, M., Rahman, M. K., Appolloni, A., Zailani, S., Martins, F. P., & Botelho Junior, A. B. (2025). Impact of Industry 4.0 on sustainability of Malaysia's manufacturing industry in post-COVID era. foresight.
39. Rijwani, T., Kumari, S., Srinivas, R., Abhishek, K., Iyer, G., Vara, H., ... & Gupta, M. (2025). Industry 5.0: A review of emerging trends and transformative technologies in the next industrial revolution. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 19(2), 667-679.
40. Зубкова, А. Б., Майгурова, Д., & Місюня, Р. М. (2023). Управління проєктами цифрової трансформації міжнародних підприємств: ключові відмінності Індустрії 4.0 та 5.0. *Scientific journal "Modeling The Development Of The Economic Systems"*. С.121-130

41. Rijwani, T., Kumari, S., Srinivas, R., Abhishek, K., Iyer, G., Vara, H., ... & Gupta, M. (2025). Industry 5.0: A review of emerging trends and transformative technologies in the next industrial revolution. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 19(2), 667-679.
42. Ricci, A., Ronca, V., Capotorto, R., Giorgi, A., Vozzi, A., Germano, D., ... & Aricò, P. (2025). Understanding the Unexplored: A Review on the Gap in Human Factors Characterization for Industry 5.0. *Applied Sciences*, 15(4), 1822.
43. Olsson, A. K., Eriksson, K. M., & Carlsson, L. (2025). Management toward Industry 5.0: a co-workership approach on digital transformation for future innovative manufacturing. *European Journal of Innovation Management*, 28(1), 65-84.
44. Kulkarni, P., & Pandiya, B. (2025). Human-Machine Interface in the Industry 5.0 Workplace. In *Expanding Operations Through Agile Principles and Sustainable Practices* (pp. 341-366). IGI Global Scientific Publishing.
45. Islam, M. T., Sepanloo, K., Woo, S., Woo, S. H., & Son, Y. J. (2025). A review of the industry 4.0 to 5.0 transition: exploring the intersection, challenges, and opportunities of technology and Human–Machine collaboration. *Machines*, 13(4), 267.
46. Ghugar, U., Nayak, R., & Awotunde, J. B. Enabling Technologies for Seamless Human-Machine Interaction in Industry 5.0. In *Computational Intelligence for Analysis of Trends in Industry 4.0 and 5.0* (pp. 215-240). Auerbach Publications.
47. Narkhede, G. B., Pasi, B. N., Rajhans, N., & Kulkarni, A. (2025). Industry 5.0 and sustainable manufacturing: a systematic literature review. *Benchmarking: An International Journal*, 32(2), 608-635.
48. Daoud, A. O., Kineber, A. F., Ali, A. H., & Elseknidy, M. (2025). Empowering Sustainable Infrastructure and Sustainable Development Goals Through Industry 5.0 Implementation. *Sustainable Development*.

49. Ghai, A. S., Sharma, R. C., Jasola, S., & Zamfiroiu, A. (2025). Intelligent sustainable development in online learning: Insights from higher education and Industry 5.0. In *Impacts of Generative AI on the Future of Research and Education* (pp. 417-444). IGI Global.

50. Veeramachaneni, V. (2025). Edge Computing: Architecture, Applications, and Future Challenges in a Decentralized Era. *Recent Trends in Computer Graphics and Multimedia Technology*, 7(1), 8-23.

51. Langås, E. F., Zafar, M. H., & Sanfilippo, F. (2025). Exploring the synergy of human-robot teaming, digital twins, and machine learning in industry 5.0: A step towards sustainable manufacturing. *Journal of Intelligent Manufacturing*, 1-24.

52. Mon, B. F., Hayajneh, M., Ali, N. A., Ullah, F., Ullah, H., & Alkobaisi, S. (2025). Digital Twins in the IIoT: Current Practices and Future Directions Toward Industry 5.0. *Computers, Materials & Continua*, 83(3).

53. Rathod, S., Parate, P., Rathod, N., & Wagh, S. (2025). Big Data Analytics (BDA) for Industry 5.0. In *Industry 4.0, Smart Manufacturing, and Industrial Engineering* (pp. 36-64). CRC Press.

54. Demska, N., Yevsieiev, V., Maksymova, S., & Ababneh, J. (2025). DECISION-MAKING MODEL FOR CONTROLLING A COLLABORATIVE ROBOT-MANIPULATOR BASED ON THE SENSOR FUSION METHOD AND CNN APPROACH TO RULE FORMATION. *Multidisciplinary Journal of Science and Technology*, 5(6), 846-859.

55. Haque, S., & Singh, S. K. (2025). Impact of 6G technologies on Industry 5.0. In *6G Impacts on Natural Habitats and Human Life* (pp. 109-134). IGI Global Scientific Publishing.

56. Thiruvankadam, K., & Pandi, V. Trustworthiness of Blockchain Technology in Healthcare 5.0 and Industry 5.0. In *Edge AI for Industry 5.0 and Healthcare 5.0 Applications* (pp. 244-268). Auerbach Publications.

57. Rijwani, T., Kumari, S., Srinivas, R., Abhishek, K., Iyer, G., Vara, H., ... & Gupta, M. (2025). Industry 5.0: A review of emerging trends and transformative

technologies in the next industrial revolution. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 19(2), 667-679.

58. Pant, R., Singh, R., Gehlot, A., & Thakur, A. K. (2025, February). A Systematic Review of Emerging Industry 5.0 Technologies: Enhancing Agile Manufacturing for Sustainability. In *Operations Research Forum* (Vol. 6, No. 1, p. 29). Cham: Springer International Publishing.

59. Neri, A., Butturi, M. A., Tomasin da Silva, L., Lolli, F., Gamberini, R., & Sellitto, M. A. (2025). Fostering lithium-ion battery remanufacturing through Industry 5.0. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 1-18.

60. Salierno, G., Leonardi, L., & Cabri, G. (2025). Generative AI and Large Language Models in Industry 5.0: Shaping Smarter Sustainable Cities. *Encyclopedia*, 5(1), 30.

61. Safavi Jahromi, G., & Ghazinoory, S. (2025). Clothing industry in transition from Industry 4.0 to Industry 5.0. *The Journal of The Textile Institute*, 116(3), 365-379.

62. Rijwani, T., Kumari, S., Srinivas, R., Abhishek, K., Iyer, G., Vara, H., ... & Gupta, M. (2025). Industry 5.0: A review of emerging trends and transformative technologies in the next industrial revolution. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 19(2), 667-679.

63. Olsson, A. K., Eriksson, K. M., & Carlsson, L. (2025). Management toward Industry 5.0: a co-workship approach on digital transformation for future innovative manufacturing. *European Journal of Innovation Management*, 28(1), 65-84.

64. Kaswan, M. S., Chaudhary, R., Garza-Reyes, J. A., & Singh, A. (2025). A review of Industry 5.0: from key facets to a conceptual implementation framework. *International Journal of Quality & Reliability Management*, 42(4), 1196-1223.

65. Safavi Jahromi, G., & Ghazinoory, S. (2025). Clothing industry in transition from Industry 4.0 to Industry 5.0. *The Journal of The Textile Institute*, 116(3), 365-379.
66. Yevsieiev V. Ecosystem Model of the Concept of Industry 5.0 / V. Yevsieiev // *Digital innovation & sustainable development 2024 : Proceedings of I-st International Conference*, November 15, 2024. - Kharkiv, 2024. - P. 12-13.
67. Sarkar, B. D., Shardeo, V., Dwivedi, A., & Pamucar, D. (2024). Digital transition from industry 4.0 to industry 5.0 in smart manufacturing: A framework for sustainable future. *Technology in Society*, 78, 102649.
68. Agrawal, S., Agrawal, R., Kumar, A., Luthra, S., & Garza-Reyes, J. A. (2024). Can industry 5.0 technologies overcome supply chain disruptions?—a perspective study on pandemics, war, and climate change issues. *Operations Management Research*, 17(2), 453-468.
69. Sigov, A., Ratkin, L., Ivanov, L. A., & Xu, L. D. (2024). Emerging enabling technologies for industry 4.0 and beyond. *Information Systems Frontiers*, 26(5), 1585-1595.
70. Leng, J., Zhu, X., Huang, Z., Li, X., Zheng, P., Zhou, X., ... & Liu, Q. (2024). Unlocking the power of industrial artificial intelligence towards Industry 5.0: Insights, pathways, and challenges. *Journal of Manufacturing Systems*, 73, 349-363.
71. Lu, Y. (2025). The current status and developing trends of industry 4.0: A review. *Information Systems Frontiers*, 27(1), 215-234.
72. Molina, R. I. R., Ruiz, M. J. S., Molina, R. J. R., Raby, N. D. L., & Severino-González, P. (2024). Innovation systems in industry 5.0: theoretical and methodological bases. *Procedia Computer Science*, 231, 595-600.
73. Martínez-Gutiérrez, A., Díez-González, J., Perez, H., & Araújo, M. (2024). Towards industry 5.0 through metaverse. *Robotics and Computer-Integrated Manufacturing*, 89, 102764.

74. Guelida, O., Jai Andaloussi, S., & Ouchetto, O. (2024). Smart Contracts in Finance and Banking Systems in the Era of Industry 5.0: A Systematic Review. *Industry 5.0 and Emerging Technologies: Transformation Through Technology and Innovations*, 317-346.

75. Saveetha, D., Ponnusamy, V., Zdravković, N., & Nandini, S. M. (2024). Blockchain Application in Industry 5.0. Artificial Intelligence-Enabled Digital Twin for Smart Manufacturing, 421-442.

76. Rohit, K. C., Sridhar, S. K., & Gurupriya, M. (2024, August). Information Processing of Electronic Medical Records Through Intelligent Smart Contracting and its Future Landscape in Industry 5.0. In *2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 1560-1564). IEEE.

77. Sizan, N. S., Dey, D., Layek, M. A., Uddin, M. A., & Huh, E. N. (2025). Evaluating blockchain platforms for iot applications in industry 5.0: A comprehensive review. *Blockchain: Research and Applications*, 100276.

78. Parry, G. M. (2024). Blockchain Technology: Enabling Decentralized and Secure Industrial Processes in Industry 5.0. In *Industry 5.0 and Emerging Technologies: Transformation Through Technology and Innovations* (pp. 259-296). Cham: Springer Nature Switzerland.

79. Kannan, D., Amiri, A. S., Shaayesteh, M. T., Nasr, A. K., & Mina, H. (2024). Unveiling barriers to the integration of blockchain-based circular economy and Industry 5.0 in manufacturing industries: A strategic prioritization approach. *Business Strategy and the Environment*, 33(8), 7855-7886.

80. Zhen, Z., & Yao, Y. (2024). The confluence of digital twin and blockchain technologies in Industry 5.0: Transforming supply chain management for innovation and sustainability. *Journal of the Knowledge Economy*, 1-27.

81. Butera, A., & Gatteschi, V. (2024, October). EU Battery Regulation and the Role of Blockchain in Creating Digital Battery Passports. In *2024 6th*

Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS) (pp. 1-4). IEEE.

82. Spigarelli, F., Compagnucci, L., & Lepore, D. (2024). Blockchain unlocking collaborative opportunities for environmental sustainability through innovation intermediaries. *The Journal of Technology Transfer*, 1-36.

83. Liu, S., Han, W., Zhang, Z., & Chan, F. T. (2024). An analysis of performance, pricing, and coordination in a supply chain with cloud services: The impact of data security. *Computers & Industrial Engineering*, 192, 110237.

84. Surucu-Balci, E., Iris, Ç., & Balci, G. (2024). Digital information in maritime supply chains with blockchain and cloud platforms: Supply chain capabilities, barriers, and research opportunities. *Technological Forecasting and Social Change*, 198, 122978.

85. Yang, D., Li, R., & Liu, S. (2025). Exploring the Influence of Cloud Computing on Supply Chain Performance: The Mediating Role of Supply Chain Governance. *Journal of Theoretical and Applied Electronic Commerce Research*, 20(2), 70.

86. Haval, A. M. (2025). Deploying cloud computing and data warehousing to optimize supply chain management and retail analytics. In *Applications of Mathematics in Science and Technology* (pp. 810-816). CRC Press.

87. Soomro, M. A., Khan, S., & Shah, A. (2025). Cloud-Enabled Supply Chain Integration for Sustainable Firms: An Empirical Study on Technology and the Circular Economy Relationship. In *Innovation and Sustainability Through Circular Economy in Businesses* (pp. 65-90). IGI Global Scientific Publishing.

88. Rashid, A., Rasheed, R., Ngah, A. H., & Amirah, N. A. (2024). Unleashing the power of cloud adoption and artificial intelligence in optimizing resilience and sustainable manufacturing supply chain in the USA. *Journal of Manufacturing Technology Management*, (ahead-of-print).

89. Haval, A. M. (2025). Deploying cloud computing and data warehousing to optimize supply chain management and retail analytics. In *Applications of Mathematics in Science and Technology* (pp. 810-816). CRC Press.
90. Zhang, Y., Niu, Z., Liu, C., Liu, C., & Hou, K. (2024). SYSTEM DYNAMICS ANALYSIS FOR IMPACT OF CLOUD-BASED INFORMATION-SHARING ON SUPPLY CHAIN PERFORMANCE. *International Journal of Industrial Engineering*, 31(2).
91. Wang, B., Zhou, H., Li, X., Yang, G., Zheng, P., Song, C., ... & Wang, L. (2024). Human Digital Twin in the context of Industry 5.0. *Robotics and Computer-Integrated Manufacturing*, 85, 102626.
92. Bongomin, O., Mwape, M. C., Mpofu, N. S., Bahunde, B. K., Kidega, R., Mpungu, I. L., ... & Ngulube, G. (2025). Digital Twin technology advancing Industry 4.0 and Industry 5.0 across Sectors. *Results in Engineering*, 105583.
93. Korepin, V., Mohamed, T. I., Zhaksylyk, A., & Liu, J. (2024). Implementation of digital twins as a tool for increasing the efficiency of business operations. *Economics of Innovation and New Technology*, 1-16.
94. Rathod, S. B., Vyawahare, H. R., Khan, R. A. H., & Ponnusamy, S. (2024). Twinning for Success: A Guide to Strategic Business Integration With AI and Digital Twins. In *Harnessing AI and Digital Twin Technologies in Businesses* (pp. 411-431). IGI Global.
95. Ponnusamy, S., Assaf, M., Antari, J., Singh, S., & Kalyanaraman, S. (Eds.). (2024). Digital twin technology and AI implementations in future-focused businesses. IGI Global.
96. Stjepandić, J., Lützenberger, J., & Kremer, P. (2024). Business Case for Digital Twin of a Process Plant. In *Generation and Update of a Digital Twin in a Process Plant* (pp. 89-109). Cham: Springer International Publishing.
97. Shi, J. (2024). Adaptive change: Emerging economy enterprises respond to the international business environment challenge. *Technovation*, 133, 102998.

98. Rasshyvalov, D., Portnov, Y., Sigaieva, T., Alboshchii, O., & Rozumnyi, O. (2024). Navigating geopolitical risks: Implications for global
99. Bentahar, O., & Belhadi, A. (2025, January). Integrating project management and supply chain management for resilient and sustainable operations in a VUCA world. In *Supply Chain Forum: An International Journal* (Vol. 26, No. 1, pp. 1-6). Taylor & Francis.
100. Holloway, S. (2025). Adapting Marketing Strategies to Evolving Consumer Preferences: The Role of Supply Chain Flexibility. Available at SSRN 5135313.
101. Farazi, M. Z. R. (2025). Enhancing supply chain resilience with multi-agent systems and machine learning: a framework for adaptive decision-making. *The American Journal of Engineering and Technology*, 7(03), 6-20.
102. Holloway, S. (2025). Leveraging Collaboration for Supply Chain Resilience: Insights from Emerging Technology Integration. Available at SSRN 5121813.
103. Нагара, М. (2022). Прогресивні бізнес-моделі: домінування цінностей Індустрії 5.0. *Економіка та суспільство*, (45). <https://doi.org/10.32782/2524-0072/2022-45-38>.
104. Xu, X., Lu, Y., Vogel-Heuser, B., & Wang, L. (2021). Industry 4.0 and Industry 5.0–Inception, conception and perception. *Journal of manufacturing systems*, 61, 530-535.
105. Akundi, A., Euresti, D., Luna, S., Ankobiah, W., Lopes, A., & Edinbarough, I. (2022). State of Industry 5.0–Analysis and identification of current research trends. *Applied System Innovation*, 5(1), 27.
106. Babkin, A., Tashenova, L., Mamrayeva, D., & Shkarupeta, E. (2022, April). Industry 5.0 and digital ecosystems: scientometric research of development trends. In *International Scientific Conference “Digital Transformation on Manufacturing, Infrastructure & Service”* (pp. 544-564). Cham: Springer Nature Switzerland.

107. Javed, S., Javed, S., van Deventer, J., Mokayed, H., & Delsing, J. (2023, May). A smart manufacturing ecosystem for industry 5.0 using cloud-based collaborative learning at the edge. In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium* (pp. 1-6). IEEE.

108. Gorodetsky, V., Larukchin, V., & Skobelev, P. (2020). Conceptual model of digital platform for enterprises of industry 5.0. In *Intelligent Distributed Computing XIII* (pp. 35-40). Springer International Publishing.

109. Singh, T., Singh, D., Singh, C. D., & Singh, K. (2023). *Industry 5.0. Factories of the Future: Technological Advancements in the Manufacturing Industry*, 21-45.

110. Li, J., Sun, B., & Bo, Y. U. (2024). Suitable innovation in Industry 5.0: The moderating mechanism of innovation transmission types in innovation ecosystem. *IEEE Transactions on Engineering Management*.

111. Sangaiah, A. K., Wang, X., Obaidat, M. S., Huang, P. C., & Govindan, K. (2024). Guest Editorial Data-Driven Innovation and Adversarial Learning Models for Industry 5.0 Toward Consumer Digital Ecosystems. *IEEE Transactions on Consumer Electronics*, 70(2), 4878-4881.

112. Maddikunta, P. K. R., Pham, Q. V., Prabadevi, B., Deepa, N., Dev, K., Gadekallu, T. R., ... & Liyanage, M. (2022). Industry 5.0: A survey on enabling technologies and potential applications. *Journal of industrial information integration*, 26, 100257.

113. Wagh, M. S., Sowjanya, S., Nath, P. C., Chakraborty, A., Amrit, R., Mishra, B., ... & Mohanta, Y. K. (2024). Valorisation of agro-industrial wastes: circular bioeconomy and biorefinery process—a sustainable symphony. *Process Safety and Environmental Protection*, 183, 708-725.

114. Herrmann, L. W., Letti, L. A. J., de Oliveira Penha, R., Soccol, V. T., Rodrigues, C., & Soccol, C. R. (2024). *Bacillus* genus industrial applications and innovation: First steps towards a circular bioeconomy. *Biotechnology Advances*, 70, 108300.

115. Proestou, M., Schulz, N., & Feindt, P. H. (2024). A global analysis of bioeconomy visions in governmental bioeconomy strategies. *Ambio*, 53(3), 376-388.
116. Zafar, M. H., Langås, E. F., & Sanfilippo, F. (2024). Exploring the synergies between collaborative robotics, digital twins, augmentation, and industry 5.0 for smart manufacturing: A state-of-the-art review. *Robotics and Computer-Integrated Manufacturing*, 89, 102769.
117. Watkins, E. A., Moss, E., Manuvinakurike, R., Shi, M., Beckwith, R., & Raffa, G. (2025). ACE, Action and Control via Explanations: A Proposal for LLMs to Provide Human-Centered Explainability for Multimodal AI Assistants. *arXiv preprint arXiv:2503.16466*.
118. Wang, Y., Wang, L., & Siau, K. L. (2025). Human-centered interaction in virtual worlds: A new era of generative artificial intelligence and metaverse. *International Journal of Human-Computer Interaction*, 41(2), 1459-1501.
119. Xu, J., Sun, Q., Han, Q. L., & Tang, Y. (2025). When embodied AI meets Industry 5.0: Human-centered smart manufacturing. *IEEE/CAA Journal of Automatica Sinica*, 12(3), 485-501.
120. Chen, C., Zhang, Z., Khalilov, I., Guo, B., Gebreegziabher, S. A., Ye, Y., ... & Li, T. J. J. (2025). Toward a human-centered evaluation framework for trustworthy llm-powered gui agents. *arXiv preprint arXiv:2504.17934*.
121. OYESANYA, O. S., & FA-YUSUF, H. S. (2025). Islamic Finance as a Catalyst for Inclusive Growth in the Era of Industry 5.0. *African Development Finance Journal*, 8(4), 60-80.
122. Sharma, R., & Gupta, H. (2024). Leveraging cognitive digital twins in industry 5.0 for achieving sustainable development goal 9: An exploration of inclusive and sustainable industrialization strategies. *Journal of Cleaner Production*, 448, 141364.
123. Ghobakhloo, M., Iranmanesh, M., Foroughi, B., Tirkolaei, E. B., Asadi, S., & Amran, A. (2023). Industry 5.0 implications for inclusive sustainable

manufacturing: An evidence-knowledge-based strategic roadmap. *Journal of Cleaner Production*, 417, 138023.

124. Grabowska, S. (2022). Key components of the business model in an Industry 5.0 environment. *Zeszyty Naukowe. Organizacja i Zarządzanie/Politechnika Śląska*, (158), 191-199.

125. Botti, A., & Baldi, G. (2025). Business model innovation and Industry 5.0: A possible integration in GLAM institutions. *European Journal of Innovation Management*, 28(1), 27-49.

126. Tyagi, A. K., Lakshmi Priya, R., Mishra, A. K., & Balamurugan, G. (2023). Industry 5.0: potentials, issues, opportunities, and challenges for society 5.0. *Privacy Preservation of Genomic and Medical Data*, 409-432.

127. Paschek, D., Luminosu, C. T., & Ocakci, E. (2022). Industry 5.0 challenges and perspectives for manufacturing systems in the society 5.0. *Sustainability and Innovation in Manufacturing Enterprises: Indicators, Models and Assessment for Industry 5.0*, 17-63.

128. Strange, R., Chen, L., & Fleury, M. T. L. (2022). Digital transformation and international strategies. *Journal of International Management*, 28(4), 100968.

129. Bhaumik, S. K., Estrin, S., & Narula, R. (2024). Integrating host-country political heterogeneity into MNE–state bargaining: insights from international political economy. *Journal of International Business Studies*, 55(2), 157-171.

130. Fogaça, D. R., Grijalvo, M., & Sacomano Neto, M. (2025). What Are Industry 4.0 and Industry 5.0 All About? An Integrative Institutional Model for the New Industrial Paradigms. *Administrative Sciences*, 15(4), 118.

131. Mouazen, A. M., Hernández-Lara, A. B., Chahine, J., & Halawi, A. (2025). Triple bottom line sustainability and Innovation 5.0 management through the lens of Industry 5.0, Society 5.0 and Digitized Value Chain 5.0. *European Journal of Innovation Management*.

132. Agrawal, A., Sharma, A., & Sarkar, B. D. (2025). Business Strategies for Sustainable Development: Leveraging Industry 5.0 and Circular Pharmaceutical

Supply Chains to Overcome Medicine Waste. *Business Strategy and the Environment*.

133. Jain, A.; Jain, P.; Chan, F.T.; Singh, S. A review on manufacturing flexibility. *Int. J. Prod. Res.* 2013, 51, 5946–5970.

134. Gustavsson, S.O. Flexibility and productivity in complex production processes. *Int. J. Prod. Res.* 1984, 22, 801–808.

135. Jovane, F.; Koren, Y.; Boër, C. Present and Future of Flexible Automation: Towards New Paradigms. *CIRP Ann.* 2003, 52, 543–560.

136. Urbani, A.; Molinari-Tosatti, L.; Pierpaoli, F. New frontiers for manufacturing in mass customization. In *Proceedings of the 5th Biannual World Automation Congress, WAC-02, Orlando, FL, USA, 9–13 June 2002*.

137. Mourtzis, D.; Doukas, M. The Evolution of Manufacturing Systems: From Craftsmanship to the Era of Customisation. In *Advances in Logistics, Operations, and Management Science*; IGI Global: Hershey, PA, USA, 2014; pp. 1–29.

138. Chammas, A.; Quaresma, M.; Mont’Alvão, C. A Closer Look on the User Centred Design. *Procedia Manuf.* 2015, 3, 5397–5404.

139. Bragança, S.; Costa, E.; Castellucci, I.; Arezes, P.M. A Brief Overview of the Use of Collaborative Robots in Industry 4.0: Human Role and Safety. In *Occupational and Environmental Safety and Health*; Springer International Publishing: Cham, Switzerland, 2019; pp. 641–650.

140 Fager, P.; Sgarbossa, F.; Calzavara, M. Cost modelling of onboard cobot-supported item sorting in a picking system. *Int. J. Prod. Res.* 2020, 59, 3269–3284.

141. El Makrini, I.; Elprama, S.A.; Van den Bergh, J.; Vanderborgh, B.; Knevels, A.J.; Jewell, C.I.; Stals, F.; De Coppel, G.; Ravysse, I.; Potargent, J.; et al. Working with Walt: How a Cobot Was Developed and Inserted on an Auto Assembly Line. *IEEE Robot. Autom. Mag.* 2018, 25, 51–58.

142. Safeea, M.; Neto, P.; Béarée, R. The Third Hand, Cobots Assisted Precise Assembly. In *Lecture Notes in Computer Science*; Springer International Publishing: Cham, Switzerland, 2019; pp. 454–457.

143. Barravecchia, F.; Mastrogiacomo, L.; Franceschini, F. A general cost model to assess the implementation of collaborative robots in assembly processes. *Int. J. Adv. Manuf. Technol.* 2023, 125, 5247–5266.

144. Heredia, J.; Schlette, C.; Kjærgaard, M.B. Breaking Down the Energy Consumption of Industrial and Collaborative Robots: A Comparative Study. In *Proceedings of the 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sinaia, Romania, 12–15 September 2023.

145. Keshvarparast, A.; Battini, D.; Battaia, O.; Pirayesh, A. Collaborative robots in manufacturing and assembly systems: Literature review and future research agenda. *J. Intell. Manuf.* 2023.

146. Giberti, H.; Abbattista, T.; Carnevale, M.; Giagu, L.; Cristini, F. A Methodology for Flexible Implementation of Collaborative Robots in Smart Manufacturing Systems. *Robotics* 2022, 11, 9.

146. Lee, E.; Barthelmey, A.; Reckelkamm, T.; Kang, H.; Son, J. A Study on Human-Robot Collaboration based Hybrid Assembly System for Flexible Manufacturing. In *Proceedings of the IECON 2019–45th Annual Conference of the IEEE Industrial Electronics Society*, Lisbon, Portugal, 14–17 October 2019.

148. Sherwani, F.; Asad, M.M.; Ibrahim, B. Collaborative Robots and Industrial Revolution 4.0 (IR 4.0). In *Proceedings of the 2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, Karachi, Pakistan, 26–27 March 2020.

149. Strassmair, C.; Taylor, N. Human Robot Collaboration in Production Environments. In *Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication 2014: Towards a Framework for Joint Action Workshop*, IEEE RO-MAN 2014, Edinburgh, UK, 25–29 August 2014.

150. Othman, U.; Yang, E. Human–Robot Collaborations in Smart Manufacturing Environments: Review and Outlook. *Sensors* 2023, 23, 5663.

151. Berry, S., Bouygues, A., Drant, J., Gonzalez-Moreau, A., Servouin, C., Madec, C., & Madur, A. (2024, August). Cobotisation for SRF cryomodules at CEA: focus on ESS and future prospects. In 32nd Linear Accelerator Conference (LINAC2024), Chicago, USA.

152. Climent Giménez, E. M. (2025). Developing Real-Time Digital Twins for Industrial Cobots.

153. Shehata, O. R. (2023). Design and Implementation of a User-Friendly Platform Interface (UFPI) for Controlling the KUKA LBR iiwa Robot Via Real-Time Hand Guiding (Doctoral dissertation, Bochum University of Applied Sciences).

154. Fernandez-Vega, M., Alfaro-Viquez, D., Zamora-Hernandez, M., Garcia-Rodriguez, J., & Azorin-Lopez, J. (2025). Transforming Robots into Cobots: A Sustainable Approach to Industrial Automation. *Electronics*, 14(11), 2275.

155. Кирилович, В. А., Мельничук, П. П., Кравчук, А. Р., & Яновський, В. А. (2022). Термінологічний та змістовний аспекти колаборативної робототехніки: аналіз та рекомендації. Державний університет «Житомирська політехніка» №2(90) С. 13-21. <http://eztuir.ztu.edu.ua/123456789/8113>.

156. Inoue T. Standardization of Assistive Products with Robotic Technology – From a Perspective of ISO/TC173 / T.Inoue, S.Yamauchi, K.-E.Westman // *Harnessing the Power of Technology to Improve Lives : Studies in Health Technology and Informatics*. – 2017. – Vol. 242.

157. Villaronga Fosch Eduard ISO 13482:2014 and Its Confusing Categories. Building a Bridge Between Law and Robotics / Eduard Fosch Villaronga // *Mechanisms and Machine Science*. – 2016. – Vol. 39.

159. Robots and Robotics Devices – Safety Requirements for Personal Care Robots : ISO 13482:2014 [Electronic resource]. – Access mode : <https://www.iso.org/standard/53820.html>.

160. Application of dynamically scaled safety zones based on the ISO/TS 15066:2016 for collaborative robotics / L.Scalera, A.Giusti, V. di Cosmo, M.Riedl // International Journal of Mechanics and Control. – 2020. – № 21 (1), June. – P. 41–50

161. Collaborative Robotics Safety Control Application Using Dynamic Safety Zones Based on the ISO/TS 15066:2016 / Vincenzo Di Cosmo and other // Advances in Intelligent Systems and Computing. – 2019. – Vol. 980

162. Chemwenoa P. Orienting safety assurance with outcomes of hazard analysis and risk assessment : A review of the ISO 15066 standard for collaborative robot systems / P.Chemwenoa, L.Pintelonb, W.Decreb // Safety Science. – 2020. – Vol. 129, September

163. Robots and robotic devices – Collaborative robots : ISO/TS 15066:2016 / International Organisation of Standardisation [Electronic resource]. – Access mode : <https://www.iso.org/standard/62996.html>.

164. Risk management : ISO 31000:2018 [Electronic resource]. – Access mode : <https://www.iso.org/ru/iso-31000-riskmanagement.html>.

165. McGarry L. Assessment of ISO Standardisation to Identify an Industrial Robot's Base Frame / L.McGarry, J.Butterfield, A.Murphy // Robotics and Computer-Integrated Manufacturing. – 2022. – Vol. 74.

166. BS EN ISO 9283:1998 [Electronic resource]. – Access mode : <https://cutt.ly/nNEhk8J>.

167. ISO/TR 13309:1995 [Electronic resource]. – Access mode : <https://www.iso.org/standard/21679.html>.

168. Safe Human-Robot Cooperation in an Industrial Environment / N.Pedrocchi, F.Vicentini, M.Matteo, L.Molinari Tosatti // International Journal of Advanced Robotic Systems. – 2013. – Vol. 10, Issue 1

170. Robots and robotic devices. Safety requirements for industrial robots. Part 1: Robots (ISO 10218-1:2011) / International Organisation of Standardisation, 2011.

171. Dalle Mura M. Designing assembly lines with humans and collaborative robots: A genetic approach / M. Dalle Mura, G.Dini // *CIRP Annals*. – 2019. – Vol. 68, Issue 1.
172. Malik A.A. Collaborative robots in assembly: A practical approach for tasks distribution / A.A. Malik, A.Bilberg // *Procedia CIRP*. – 2019. – Vol. 81.
173. Accorsi R. An application of collaborative robots in a food production facility / R.Accorsi // *Procedia Manufacturing*. – 2019. – Vol. 38.
174. Collision detection algorithm for collaborative robots considering joint friction / J.Xiao and other // *International Journal of Advanced Robotic Systems*. – 2018. – Vol. 15, Issue 4
175. Aaltonena I. Experiences and expectations of collaborative robots in industry and academia: barriers and development needs / I.Aaltonena, T.Salmi // *Procedia Manufacturing*. – 2019. – Vol. 38.
176. Vicentini F. Terminology in safety of collaborative robotics / F.Vicentini // *Robotics and Computer-Integrated Manufacturing*. – 2020. – Vol. 63.
177. Demska, N., Yevsieiev, V., Maksymova, S., & Alkhalailah, A. (2025). Analysis of Methods, Models and Algorithms for a Collaborative Robots Group Decentralized Control. *ACUMEN: International journal of multidisciplinary research*, 2(2), 235-249.
178. Невлюдов, І. ІІ., Євсєєв, В. В., & Гурін, Д. В. (2025). MODEL DEVELOPMENT OF DYNAMIC REPRESENTATION A MODEL DESCRIPTION PARAMETERS FOR THE ENVIRONMENT OF A COLLABORATIVE ROBOT MANIPULATOR WITHIN THE INDUSTRY 5.0 FRAMEWORK. Системи управління, навігації та зв'язку. Збірник наукових праць, 1(79), 42-48.
179. Sun, J., Zhang, Q., Lu, Y., Huang, B., & Li, Q. (2025). A Review of Touching-Based Underwater Robotic Perception and Manipulation. *Machines*, 13(1), 41.

180. Hearn, G., Channa, N. A., & Casali, G. L. (2025). When socio-technical imaginaries meet the reality of adoption: Cobots in manufacturing. *Futures*, 171, 103596.

181. Puttero, S., Verna, E., Genta, G., & Galetto, M. (2025). Collaborative robots for quality control: an overview of recent studies and emerging trends. *Journal of Intelligent Manufacturing*, 1-37.

182. Stewart, J. (2025). Using Cobot Welding to Reduce the Welder Workforce Shortage: A Basic Qualitative Study of the Perceptions of Manufacturing Managers in Small and Medium Size Enterprises in the United States (Doctoral dissertation, American College of Education).

183. Crivelli, D., & Balconi, M. (2025). Industrial Collaborative Robotics, Humanoid Cobots, and Social Embodiment: A Critical Look on Human–Robot Interaction. *International Journal of Humanoid Robotics*, 22(01), 1-16.

184. Yevsieiev, V., Hamdan, M., Maksymova, S., & Abu-Jassar, A. (2025). A human-centric approach to control collaborative robots within Industry 5.0. *Multidisciplinary Journal of Science and Technology*, 5(5), 351-361.

185. Невлюдов, І. ІІ., Євсєєв, В. В., & Гурін, Д. В. (2025). MODEL DEVELOPMENT OF DYNAMIC REPRESENTATION A MODEL DESCRIPTION PARAMETERS FOR THE ENVIRONMENT OF A COLLABORATIVE ROBOT MANIPULATOR WITHIN THE INDUSTRY 5.0 FRAMEWORK. Системи управління, навігації та зв'язку. Збірник наукових праць, 1(79), 42-48.

186. Rehman, A., & Umar, T. (2025). Literature review: Industry 5.0. Leveraging technologies for environmental, social and governance advancement in corporate settings. *Corporate Governance: The International Journal of Business in Society*, 25(2), 229-251.

187. Narkhede, G. B., Pasi, B. N., Rajhans, N., & Kulkarni, A. (2025). Industry 5.0 and sustainable manufacturing: a systematic literature review. *Benchmarking: An International Journal*, 32(2), 608-635.

188. Sarkar, B. D., Sharma, I., & Shardeo, V. (2025). A multi-method examination of barriers to traceability in Industry 5.0-enabled digital food supply chains. *The International Journal of Logistics Management*, 36(2), 354-380.
189. Safavi Jahromi, G., & Ghazinoory, S. (2025). Clothing industry in transition from Industry 4.0 to Industry 5.0. *The Journal of The Textile Institute*, 116(3), 365-379.
190. Kermenov, R., Foix, S., Borràs, J., Castorani, V., Longhi, S., & Bonci, A. (2025). Automating the hand layup process: On the removal of protective films with collaborative robots. *Robotics and Computer-Integrated Manufacturing*, 93, 102899.
191. Zia, A., & Haleem, M. (2025). Bridging Research Gaps in Industry 5.0: Synergizing Federated Learning, Collaborative Robotics, and Autonomous Systems for Enhanced Operational Efficiency and Sustainability. *IEEE Access*.
192. Gao, Z., Chen, C., Peng, F., Zhang, Y., Liu, H., Zhou, W., ... & Tang, X. (2025). Adaptive safety-critical control using a variable task energy tank for collaborative robot tasks under dynamic environments. *Robotics and Computer-Integrated Manufacturing*, 94, 102964.
193. Zheng, W., Guo, D., Yang, W., & Liu, H. (2025). A Large-Area Robotic Skin for Intelligent Tactile Interaction of Collaborative Robots. *IEEE/ASME Transactions on Mechatronics*.
194. Zheng, C., Li, Z., Zhang, Z., Zhang, L., & Tang, Q. (2025). Multi-objective restarted simulated annealing algorithm for assembly line balancing problem with collaborative robots considering ergonomics risks. *Flexible Services and Manufacturing Journal*, 1-39.
195. Zheng, C., Li, Z., Janardhanan, M., Zhang, Z., & Zhang, L. (2025). Improved swarm-based metaheuristics for optimizing human–robot collaborative assembly lines with multi-type collaborative robots. *Flexible Services and Manufacturing Journal*, 1-63.

196. Demska, N., Yevsieiev, V., Maksymova, S., & Alkhalailah, A. (2025). Analysis of Methods, Models and Algorithms for a Collaborative Robots Group Decentralized Control. *ACUMEN: International journal of multidisciplinary research*, 2(2), 235-249.
197. Zheng, C., Du, Y., Xiao, J., Sun, T., Wang, Z., Eynard, B., & Zhang, Y. (2025). Semantic map construction approach for human-robot collaborative manufacturing. *Robotics and Computer-Integrated Manufacturing*, 91, 102845.
198. Liu, J., Yang, J., Mao, J., Zhu, T., Xie, Q., Li, Y., ... & Li, S. (2025). Flexible active safety motion control for robotic obstacle avoidance: A cbf-guided mpc approach. *IEEE Robotics and Automation Letters*.
199. Yevsieiev, V., Ababneh, J., Maksymova, S., & Abu-Jassar, A. (2025). DEVELOPMENT OF A MATHEMATICAL MODEL FOR SIMULATING A DECENTRALIZED CONTROL SYSTEM FOR COLLABORATIVE ROBOT NETWORKS. *Multidisciplinary Journal of Science and Technology*, 5(5), 1187-1202.
200. Chala, O., Yevsieiev, V., Maksymova, S., & Abu-Jassar, A. (2025). Mathematical Model Based on Multi-Agent Reinforcement Learning (MARL) and Partially Observable Markov Decision Process (POMDP) for Modeling Cargo Movement for a Mobile Robots Group. *Multidisciplinary Journal of Science and Technology*, 5(4), 480-489.
201. Yevsieiev, V., & Gurin, D. (2024). Study of Methods of Dynamic Description of The Environment for Collaborative Robots-Manipulators in the Concepts of Industry 5.0 (Doctoral dissertation, Collection of scientific papers «SCIENTIA»).
202. Meziane, R., Otis, M. J. D., & Ezzaidi, H. (2017). Human-robot collaboration while sharing production activities in dynamic environment: SPADER system. *Robotics and Computer-Integrated Manufacturing*, 48, 243-253.

203. Hameed, A., Ordys, A., Możaryn, J., & Sibilska-Mroziewicz, A. (2023). Control system design and methods for collaborative robots. *Applied Sciences*, 13(1), 675.
204. Yevsieiev, V., Alkhalaileh, A., Maksymova, S., & Gurin, D. (2024). Research of Existing Methods of Representing a Collaborative Robot-Manipulator Environment within the Framework of Cyber-Physical Production Systems. *Multidisciplinary Journal of Science and Technology*, 4(9), 112-120.
205. Tiwari, R., Srinivaas, A., & Velamati, R. K. (2025). Adaptive Navigation in Collaborative Robots: A Reinforcement Learning and Sensor Fusion Approach. *Applied System Innovation*, 8(1), 9.
206. Zhang, Q., Hu, S., Duan, J., Qin, J., & Zhou, Y. (2025, January). A SAC-Bi-RRT Two-Layer Real-Time Motion Planning Approach for Robot Assembly Tasks in Unstructured Environments. In *Actuators* (Vol. 14, No. 2, p. 59). MDPI.
207. Brian, W., Gutiérrez, S., & Gil, J. J. (2025). Using Franka Emika collaborative robot as a haptic device. *Mechatronics*, 106, 103287.
208. Zhang, F., Wang, Y., Wang, W., Xin, J., Peng, J., & Wang, Y. (2025). An Improved Dual Neural Network Method Based on Levy Flight for Multi-Robot Cooperative Area Coverage Search in 3D Unknown Environments. *IEEE Transactions on Cognitive and Developmental Systems*.
209. Qiang, S. (2025). Differential Dynamics Modeling and Simulation Analysis of Multi-Agent Cooperative Motion. *Journal of Advances in Engineering and Technology*, 2(1).
210. Yan, M., Zhu, Z., Zhao, Y., Chen, B., Ji, Y., Xu, K., & Li, S. Goal-Oriented Multi-Robot Collaborative Source Search with Dynamic Exploration-Exploitation Balance in Large-Scale Constrained Areas. Available at SSRN 5099827.
211. Yevsieiev, V., Alkhalaileh, A., Maksymova, S., & Gurin, D. (2024). Research of Existing Methods of Representing a Collaborative Robot-Manipulator

Environment within the Framework of Cyber-Physical Production Systems. *Multidisciplinary Journal of Science and Technology*, 4(9), 112-120.

212. Maksymova, S., Abu-Jassar, A., Gurin, D., & Yevsieiev, V. (2024). Comparative Analysis of methods for Predicting the Trajectory of Object Movement in a Collaborative Robot-Manipulator Working Area. *Multidisciplinary Journal of Science and Technology*, 4(10), 38-48.

213. Li, Z., Wei, H., Liu, C., He, Y., Liu, G., Zhang, H., & Li, W. (2024). An improved iterative approach with a comprehensive friction model for identifying dynamic parameters of collaborative robots. *Robotica*, 42(5), 1500-1522.

214. Saleem, Z., Gustafsson, F., Furey, E., McAfee, M., & Huq, S. (2025). A review of external sensors for human detection in a human robot collaborative environment. *Journal of Intelligent Manufacturing*, 36(4), 2255-2279.

215. Zhou, M., Wang, Z., Wang, J., & Cao, Z. (2024). Multi-robot collaborative hunting in cluttered environments with obstacle-avoiding Voronoi cells. *IEEE/CAA Journal of Automatica Sinica*, 11(7), 1643-1655.

216. Fang, S., & Li, H. (2024). Multi-vehicle cooperative simultaneous LiDAR SLAM and object tracking in dynamic environments. *IEEE Transactions on Intelligent Transportation Systems*, 25(9), 11411-11421.

217. Mozafari, B., O'Shea, N., Fenelon, M., Li, R., Daly, D. F., & Villing, R. (2024). An automated platform for measuring infant formula powder rehydration quality using a collaborative robot integrated with computer vision. *Journal of Food Engineering*, 383, 112229.

218. Yevsieiev, V., Abu-Jassar, A., Maksymova, S., & Gurin, D. (2024). Human Operator Identification in a Collaborative Robot Workspace within the Industry 5.0 Concept. *Multidisciplinary Journal of Science and Technology*, 4(9), 95-105.

219. Yevsieiv V. A Model of Using Computer Vision to Monitor the Environment of a Collaborative Manipulator Robot / V. Yevsieiv, N. Demska //

Manufacturing & Mechatronic Systems 2024 : Proceedings of VIII st International Conference, October 25-26, 2024. - Kharkiv, 2024. - P. 102-104.

220. Yuan, P., Ma, A., Yao, Y., Yao, H., Tomizuka, M., & Ding, M. (2025). Remac: Self-reflective and self-evolving multi-agent collaboration for long-horizon robot manipulation. arXiv preprint arXiv:2503.22122.

221. Lv, Q., Li, H., Deng, X., Shao, R., Li, Y., Hao, J., ... & Nie, L. (2025). Spatial-temporal graph diffusion policy with kinematic modeling for bimanual robotic manipulation. In Proceedings of the Computer Vision and Pattern Recognition Conference (pp. 17394-17404).

222. Todorova, M. (2025). Integrating Computer Vision and Mechanical Engineering for Intelligent and Adaptive Robot Control. *Global Research Review*, 1(1), 25-33.

223. Yumbla, F., Fajardo-Pruna, M., Piguave, A., Ronquillo, D., Ortiz, R., Choi, J., ... & Moon, H. (2025). An Open-Source Multi-Robot Framework System for collaborative environments based on ROS2. *IEEE Access*.

224. Papavasileiou, A., Michalos, G., & Makris, S. (2025). Quality control in manufacturing—review and challenges on robotic applications. *International Journal of Computer Integrated Manufacturing*, 38(1), 79-115.

225. Sajwan, M., & Singh, S. (2023). A review on the effectiveness of machine learning and deep learning algorithms for collaborative robot. *Archives of Computational Methods in Engineering*, 30(6), 3489-3508.

226. Borboni, A., Reddy, K. V. V., Elamvazuthi, I., AL-Quraishi, M. S., Natarajan, E., & Azhar Ali, S. S. (2023). The expanding role of artificial intelligence in collaborative robots for industrial applications: a systematic review of recent works. *Machines*, 11(1), 111.

227. Tan, S. C., Lee, A. V. Y., & Lee, M. (2022). A systematic review of artificial intelligence techniques for collaborative learning over the past two decades. *Computers and Education: Artificial Intelligence*, 3, 100097.

228. Chemweno, P., & Torn, R. J. (2022). Innovative safety zoning for collaborative robots utilizing Kinect and LiDAR sensory approaches. *Procedia CIRP*, 106, 209-214.
229. Haddeler, G., Aybakan, A., Akay, M. C., & Temeltas, H. (2020). Evaluation of 3D LiDAR sensor setup for heterogeneous robot team. *Journal of Intelligent & Robotic Systems*, 100(2), 689-709.
230. Wang, Z., Li, P., Zhang, Q., Zhu, L., & Tian, W. (2025). A LiDAR-depth camera information fusion method for human robot collaboration environment. *Information Fusion*, 114, 102717.
231. Lima, R. K. D., Heckler, W. F., Francisco, R., & Barbosa, J. L. V. (2025). Integrating collaborative learning and advanced technology in industry 5.0: A systematic mapping study and taxonomy. *International Journal of Human-Computer Interaction*, 41(1), 707-722.
232. Zia, A., & Haleem, M. (2025). Bridging Research Gaps in Industry 5.0: Synergizing Federated Learning, Collaborative Robotics, and Autonomous Systems for Enhanced Operational Efficiency and Sustainability. *IEEE Access*.
233. Kaswan, M. S., Chaudhary, R., Garza-Reyes, J. A., & Singh, A. (2025). A review of Industry 5.0: from key facets to a conceptual implementation framework. *International Journal of Quality & Reliability Management*, 42(4), 1196-1223.
234. Merchán-Cruz, E. A., Gabelaia, I., Savrasovs, M., Hansen, M. F., Soe, S., Rodriguez-Cañizo, R. G., & Aragón-Camarasa, G. (2025). Trust by Design: An Ethical Framework for Collaborative Intelligence Systems in Industry 5.0. *Electronics*, 14(10), 1952.
235. Ganesh Kumar, G., & Kanakaprabha, S. (2025). Intelligent Robots and Cobots: Concepts and Applications for Industry 5.0 Transformation. *Intelligent Robots and Cobots: Industry 5.0 Applications*, 131-157.
236. Zafar, M. H., Langås, E. F., & Sanfilippo, F. (2024). Exploring the synergies between collaborative robotics, digital twins, augmentation, and industry

5.0 for smart manufacturing: A state-of-the-art review. *Robotics and Computer-Integrated Manufacturing*, 89, 102769.

237. Barros, D., Fraga-Lamas, P., Fernández-Caramés, T. M., & Lopes, S. I. (2022, November). A cost-effective thermal imaging safety sensor for industry 5.0 and collaborative robotics. In *International Conference on Intelligent Edge Processing in the IoT era* (pp. 3-15). Cham: Springer Nature Switzerland.

238. Gouthaman, P., Shanthi, R. R., Ponnusamy, V., & Arivazhagan, N. (2024). *Computer Vision and Robotics for Industry 5.0. Industry 4.0, Smart Manufacturing, and Industrial Engineering: Challenges and Opportunities*, 196.

239. Raffik, R., Sathya, R. R., Vaishali, V., & Balavedhaa, S. (2023, June). Industry 5.0: Enhancing human-robot collaboration through collaborative robots—A review. In *2023 2nd international conference on advancements in electrical, electronics, communication, computing and automation (ICAECA)* (pp. 1-6). IEEE.

240. Yevsieiev, V., Abu-Jassar, A., Maksymova, S., & Gurin, D. (2024). Human Operator Identification in a Collaborative Robot Workspace within the Industry 5.0 Concept. *Multidisciplinary Journal of Science and Technology*, 4(9), 95-105.

241. Munasinghe, C., Amin, F. M., Scaramuzza, D., & van de Venn, H. W. (2022, September). Covered, collaborative robot environment dataset for 3d semantic segmentation. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFa)* (pp. 1-4). IEEE.

242. Kaplan, K. E., Nichols, K. A., & Okamura, A. M. (2016, May). Toward human-robot collaboration in surgery: Performance assessment of human and robotic agents in an inclusion segmentation task. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 723-729). IEEE.

243. Sun, R., Wu, C., Zhao, X., Zhao, B., & Jiang, Y. (2023). Object recognition and grasping for collaborative robots based on vision. *Sensors*, 24(1), 195.

244. Martinez-Martin, E., & Del Pobil, A. P. (2017). Object detection and recognition for assistive robots: Experimentation and implementation. *IEEE Robotics & Automation Magazine*, 24(3), 123-138.
245. Ahmed, I., Din, S., Jeon, G., Piccialli, F., & Fortino, G. (2021). Towards collaborative robotics in top view surveillance: A framework for multiple object tracking by detection using deep learning. *IEEE/CAA Journal of Automatica Sinica*, 8(7), 1253-1270.
246. Borboni, A., Reddy, K. V. V., Elamvazuthi, I., AL-Quraishi, M. S., Natarajan, E., & Azhar Ali, S. S. (2023). The expanding role of artificial intelligence in collaborative robots for industrial applications: a systematic review of recent works. *Machines*, 11(1), 111.
247. Shah, M., & Sureja, N. (2025). A comprehensive review of bias in deep learning models: Methods, impacts, and future directions. *Archives of Computational Methods in Engineering*, 32(1), 255-267.
248. Sheng, H., Wu, X., Si, X., Li, J., Zhang, S., & Duan, X. (2025). Seismic foundation model: A next generation deep-learning model in geophysics. *Geophysics*, 90(2), IM59-IM79.
249. Liu, J., & Li, B. (2025). A deep learning model for predicting mechanical behaviors of dynamic power cable of offshore floating wind turbine. *Marine Structures*, 99, 103705.
250. Javed, H., El-Sappagh, S., & Abuhmed, T. (2025). Robustness in deep learning models for medical diagnostics: security and adversarial challenges towards robust AI applications. *Artificial Intelligence Review*, 58(1), 1-107.
251. Yuan, M., Meng, H., Wu, J., & Cai, S. (2025). Global Recurrent Mask R-CNN: Marine ship instance segmentation. *Computers & Graphics*, 126, 104112.
252. Alhwaiti, Y., Khan, M., Asim, M., Siddiqi, M. H., Ishaq, M., & Alruwaili, M. (2025). Leveraging YOLO deep learning models to enhance plant disease identification. *Scientific Reports*, 15(1), 7969.

253. Zheng, W., Yuan, X., Bao, X., & Dong, Y. (2025). Adaptive support vector machine for time-variant failure probability function estimation. *Reliability Engineering & System Safety*, 253, 110510.

254. Liu, Y., Shi, H., Cao, K., Wu, S., Ye, H., Wang, X., ... & Xiong, W. (2025). kMetha-Mamba: K-means clustering mamba for methane plumes segmentation. *International Journal of Applied Earth Observation and Geoinformation*, 142, 104664.

255. Vidal-Silva, C., Pizarro, R., Castillo-Soto, M., Ingram, B., de la Fuente, C., Duarte, V., ... & Ibañez, A. (2025). A Comparative Study of a Deep Reinforcement Learning Solution and Alternative Deep Learning Models for Wildfire Prediction. *Applied Sciences* (2076-3417), 15(7).

256. Gnanasekar, S., & Sriharipriya, K. C. (2025). Enhancing Power Allocation in DAS: A Hybrid Machine Learning and Reinforcement Learning Model. *IEEE Access*.

257. Wang, Y., Yang, Q., Zeng, Z., Ren, L., Liu, L., Peng, B., ... & Shen, Y. (2025). Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*.

258. Chu, X., Huang, H., Zhang, X., Wei, F., & Wang, Y. (2025). Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*.

259. Zeng, W., Huang, Y., Liu, Q., Liu, W., He, K., Ma, Z., & He, J. (2025). Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

260. Calzada-Garcia, A., Victores, J. G., Naranjo-Campos, F. J., & Balaguer, C. (2025). A Review on Inverse Kinematics, Control and Planning for Robotic Manipulators With and Without Obstacles via Deep Neural Networks. *Algorithms*, 18(1), 23.

261. Amin, J., Gul, N., & Sharif, M. (2025). Dual-method for semantic and instance brain tumor segmentation based on UNet and mask R-CNN using MRI. *Neural Computing and Applications*, 1-19.
262. Kang, S., Hu, Z., Liu, L., Zhang, K., & Cao, Z. (2025). Object detection YOLO algorithms and their industrial applications: Overview and comparative analysis. *Electronics*, 14(6), 1104.
263. Mary, T., & Claral, A. (2025). Hybrid Deep Learning Model to Predict Students' Sentiments in Higher Educational Institutions. *International Journal of Interactive Mobile Technologies*, 19(1).
264. Icton, J. B., Henkin, S., Kadian-Dodov, D., & Vlazny, D. (2025). A day in the life of a vascular medicine specialist and highlights from the 2025 SVM Fellows and APP course. *Vascular Medicine*, 30(3), 395-399.
265. Vats, S., Sharma, V., Rawat, P., & Ratra, A. (2025). K-means clustering over distributed environment: A review. *Uncertainty in Computational Intelligence-Based Decision Making*, 173-185.
266. Surya, S., Chhetri, A., & Rao, V. (2025). Kalman Filter–Machine learning fusion for core temperature estimation in Li-ion batteries. *Journal of Energy Storage*, 113, 115656.
267. Banerjee, C., Nguyen, K., Fookes, C., & Raissi, M. (2025). A survey on physics informed reinforcement learning: Review and open problems. *Expert Systems with Applications*, 128166.
268. Yao, Y., Zhan, H., Noorian, A., & Hazratifard, M. (2025). Enhancing POI recommendations on social media: a sequential approach incorporating LSTM and user feedback. *Computing*, 107(1), 1-35.
269. Xin, X., Liu, Y., Zhang, Z., Zheng, H., & Zhou, Y. (2025). A day-ahead operational regulation method for solar district heating systems based on model predictive control. *Applied Energy*, 377, 124619.

270. Rocha, M. M. D., Favela, J., & Muchaluat-Saade, D. C. (2025). Design and Evaluation of an XML-Based Language for Programming Socially Assistive Robots. *International Journal of Social Robotics*, 1-29.

271. Assante, D., Krnjajic, Z., Pirrone, D., Radigovic, J., & Spina, A. (2025). Lego-Based Online Laboratories: Case-Study of Industrial Robotics. In *Online Laboratories in Engineering and Technology Education: State of the Art and Trends for the Future* (pp. 257-275). Cham: Springer Nature Switzerland.

272. Micheli, A., Bit-Monnot, A., Röger, G., Scala, E., Valentini, A., Framba, L., ... & Stock, S. (2025). Unified Planning: Modeling, manipulating and solving AI planning problems in Python. *SoftwareX*, 29, 102012.

273. Rao, T., Rawat, S., Gupta, A., Mehra, N., Dhondiyal, S. A., & Kapil, D. (2025). Comprehensive study on automatic number plate recognition with python using OpenCV and EasyOCR. In *Challenges in Information, Communication and Computing Technology* (pp. 501-505). CRC Press.

274. Kensert, A., Desmet, G., & Cabooter, D. (2025). MolGraph: a Python package for the implementation of molecular graphs and graph neural networks with TensorFlow and Keras. *Journal of Computer-Aided Molecular Design*, 39(1), 1-11.

275. Osborne, C., Daneshyan, F., He, R., Ye, H., Zhang, Y., & Zhou, M. (2025). Characterising open source co-opetition in company-hosted open source software projects: the cases of PyTorch, TensorFlow, and transformers. *Proceedings of the ACM on Human-Computer Interaction*, 9(2), 1-30.

276. Diandra, D. P. S. A. (2025). MONITORING LAHAN PARKIR MENGGUNAKAN PYTHON, CV2, CVZONE DAN NUMPY. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(1), 157-161.

277. Garg, A., Singh, A. P., Saraswat, A., Kumar, A., & Taluja, A. (2025, February). Optimisation Techniques in Investment Risk Management: Scipy, SLSQP and Modern Portfolio Theory. In *2025 3rd International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC)* (pp. 1329-1334). IEEE.

278. Aljamal, M., Patel, S., & Mahmood, A. (2025). Comprehensive Review of Robotics Operating System-Based Reinforcement Learning in Robotics. *Applied Sciences*, 15(4), 1840.

279. Damjanović, D., Biočić, P., Prkljačić, S., Činčurak, D., & Balen, J. (2025). A comprehensive survey on SLAM and machine learning approaches for indoor autonomous navigation of mobile robots. *Machine vision and applications*, 36(3), 55.

280. Joice, A., Tufaique, T., Tazeen, H., Igathinathane, C., Zhang, Z., Whippo, C., ... & Archer, D. (2025). Applications of Raspberry Pi for Precision Agriculture—A Systematic Review. *Agriculture*, 15(3), 227.

281. PyCharm: The only Python IDE you need. URL: <https://www.jetbrains.com/pycharm/>

282. Setiadi, A., Fadhliasis, F., Ahadi, A. H., & Gustina, G. (2025). IMPLEMENTASI PENGOLAHAN CITRA UNTUK PEMBUATAN FILTER GAMBAR BERBASIS WEB MENGGUNAKAN OPENCV DAN FLASK: Pengolahan citra digital, filter gambar, OpenCV, Flask, Python, web. *JURNAL AKADEMIKA*, 17(2), 30-34.

283. Rao, T., Rawat, S., Gupta, A., Mehra, N., Dhondiyal, S. A., & Kapil, D. (2025). Comprehensive study on automatic number plate recognition with python using OpenCV and EasyOCR. In *Challenges in Information, Communication and Computing Technology* (pp. 501-505). CRC Press.

284. Kamble, V. B., Mundhe, O. N., Walunjkar, C. M., & Kale, G. A. (2025). AI-Driven Smart Traffic Management System: An Adaptive Approach Using YOLO and OpenCV. *International Journal of Multidisciplinary on Science and Management IJMSM*, 2(2).

285. Baharuddin, A. D., & Basri, M. A. M. (2025, February). A Color-Based Real-Time Target Detection Using OpenCV for DJI Tello Drone. In *2025 21st IEEE International Colloquium on Signal Processing & Its Applications (CSPA)* (pp. 77-82). IEEE.

286. Qisthiano, M. R., & Pratiwi, A. O. (2025). Deteksi Tepi Pada Citra Objek Benda Menggunakan Algoritma Sobel Dan Prewitt Dengan Python. *Jurnal Informatika dan Teknik Elektro Terapan*, 13(2).

287. Rafidianto, M., Fatah, D. A., & Negara, Y. D. P. (2025). Analisis Sentimen Aplikasi Mpstore Menggunakan Metode Support Vector Machine Multi Kernel. *Innovative: Journal Of Social Science Research*, 5(1), 6253-6263.

288. Ulandari, I. G. A. A. M., Wiguna, I. K. A. G., & Setiawan, I. M. D. (2025). Sentiment Analysis of Twitter Comments (X) on the Cyanide Coffee Case Using Comparison of Naïve Bayes and K-NN Method Results. *TECHNOVATE: Journal of Information Technology and Strategic Innovation Management*, 2(1), 59-69.

289. Cheng, K., Shen, X., Yang, Y., Wang, T., Cao, Y., Ali, M. A., ... & Wang, D. (2025). Codemenv: Benchmarking large language models on code migration. *arXiv preprint arXiv:2506.00894*.

290. Lafler, R. P., Premier Analytics Consulting, L. L. C., & Wade, A. (2025, January). Developing Artificial and Convolutional Neural Networks with Python's Keras API for TensorFlow. In *105th AMS Annual Meeting*. AMS.

291. Kumar, S., Midya, K., Ghosh, S., Kumar, P., & Mishra, V. N. (2025). Land Use Change Analysis and Prediction of Urban Growth using Multi-Layer Perceptron Neural Network Markov Chain Model in Faridabad-A data-scarce region of Northwestern India. *Physics and Chemistry of the Earth, Parts A/B/C*, 103884.

292. Wahed, M. A., Alqaraleh, M., Alzboon, M. S., & Al-Batah, M. S. (2025). Evaluating AI and Machine Learning Models in Breast Cancer Detection: A Review of Convolutional Neural Networks (CNN) and Global Research Trends. *LatIA*, 3, 117-117.

293. Madaoui, L., Amira, A., Talha, M. K., Kerdjij, O., & Himeur, Y. (2025). Image encoding and wearable sensors-based locomotion mode recognition using convolutional recurrent neural networks. *Biomedical Signal Processing and Control*, 100, 107068.

294. Palisetti, J. R., Srilikhitha, D., Bhargava, K., Devi, B. B., & Kumar, M. B. (2025, April). Touchless Mouse Control Via Hand Gestures Using Resnet and Mediapipe. In 2025 International Conference on Inventive Computation Technologies (ICICT) (pp. 185-191). IEEE.
295. Gurin, D., Yevsieiev, V., Maksymova, S., & Alkhalaileh, A. (2024). MobileNetv2 Neural Network Model for Human Recognition and Identification in the Working Area of a Collaborative Robot. *Multidisciplinary Journal of Science and Technology*, 4(8), 5-12.
296. Gurin, D., Yevsieiev, V., Maksymova, S., & Abu-Jassar, A. (2024). Effect of Frame Processing Frequency on Object Identification Using MobileNetV2 Neural Network for a Mobile Robot. *Multidisciplinary Journal of Science and Technology*, 4(8), 36-44.
297. Hui, L., Ibrahim, A., & Hindi, R. (2025). Computer Vision-Based Concrete Crack Identification Using MobileNetV2 Neural Network and Adaptive Thresholding. *Infrastructures*, 10(2), 42.
298. Bharadwaj, S. K., Jha, R., Kumar, J., Mishra, D. K., Shinde, V., & Jadon, V. K. (2024). COMPARATIVE STUDY OF MOBILENETV2, SIMPLE CNN AND VGG19 FOR IMAGE CLASSIFICATION. *Journal of Data Acquisition and Processing*, 39(1).
299. Nasra, P., & Gupta, S. (2024, December). Efficient Orange Disease Detection Using MobileNetV2: A Deep Learning Approach. In 2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS) (pp. 900-905). IEEE.
300. Mittal, K., Gill, K. S., Aggarwal, P., Rawat, R. S., & Aluvala, S. (2024, April). Using MobileNetV2 Deep Convolutional Neural Networks and Transfer Learning for Shell and Pebble Classification. In 2024 IEEE 9th International Conference for Convergence in Technology (I2CT) (pp. 1-5). IEEE.

301. CAMShift Algorithm for Human Tracking in the Collaborative Robot Working Area / D. Gurin, V. Yevsieiev, S. Maksymova., Ahmad Alkhalaileh // Journal of Universal Science Research. – 2024. – Vol. 2(8). – P. 87–101.

302. Gurin, D. (2025). Intelligent Tracking Algorithms in Collaborative Robotic Systems: Application of Camshift and Kalman Filter. In I International Conference «Sustainable smart cities and communities: business and innovation solutions»(Сталі розумні міста та спільноти: бізнес та інноваційні рішення) (p. 17).

303. See, A. K. B., & Xing, J. (2024). Enhanced CAMSHIFT with Perceptual Grouping, Weighted Histogram, Selective Adaptation and Kalman Filtering: Improving Stability and Accuracy in Face Detection and Tracking. *Malaysian Journal of Science and Advanced Technology*, 266-289.

304. Qin, L., Tang, X., & Gao, R. (2025, April). Research on face tracking method based on improved CAMSHIFT and Kalman filter. In International Conference on Mechatronic Engineering and Artificial Intelligence (MEAI 2024) (Vol. 13555, pp. 865-869). SPIE.

305. Qin, L., Tang, X., & Gao, R. (2025, April). Research on face tracking method based on improved CAMSHIFT and Kalman filter. In International Conference on Mechatronic Engineering and Artificial Intelligence (MEAI 2024) (Vol. 13555, pp. 865-869). SPIE.

306. Yerlikaya-Özkurt, F., & Taylan, P. (2025). Enhancing classification modeling through feature selection and smoothness: A conic-fused lasso approach integrated with mean shift outlier modelling. *Journal of Dynamics and Games*, 12(1), 1-23.

307. Liu, H., Shi, H., Yuan, T., Fu, S., & Ran, B. (2025). Bus travel feature inference with small samples based on multi-clustering topic model over Internet of Things. *Future Generation Computer Systems*, 163, 107525.

308. Hajihosseini, M., Maghsoudi, A., & Ghezelbash, R. (2024). Intelligent mapping of geochemical anomalies: Adaptation of DBSCAN and mean-shift clustering approaches. *Journal of Geochemical Exploration*, 258, 107393.

309. Liu, N., Xie, Y., Su, Z., Zhao, Z., & Wang, W. (2025). Adaptive Kalman Filter-Integrated navigation measurement using inertial sensor for vehicle motion state recognition. *Measurement*, 116907.

310. Wei, X., Fan, S., Zhang, Y., Gao, W., Shen, F., Ming, X., & Yang, J. (2025). A robust adaptive error state Kalman filter for MEMS IMU attitude estimation under dynamic acceleration. *Measurement*, 242, 116097.

311. Prakash, S., Bhattacharjee, R., Bitragunta, S., Bhatia, A., & Tiwari, K. (2025). Quantum Computing-Accelerated Kalman Filtering for Satellite Clusters: Algorithms and Comparative Analysis. *IEEE Open Journal of the Computer Society*.

312. Maulana, D. A., Sofro, A. Y., Ariyanto, D., Romadhonia, R. W., Oktaviarina, A., & Purnama, M. D. (2025). STOCK PRICE PREDICTION AND SIMULATION USING GEOMETRIC BROWNIAN MOTION-KALMAN FILTER: A COMPARISON BETWEEN KALMAN FILTER ALGORITHMS. *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, 19(1), 97-106.

313. Mienye, I. D., Swart, T. G., Obaido, G., Jordan, M., & Ilono, P. (2025). Deep convolutional neural networks in medical image analysis: A review. *Information*, 16(3), 195.

314. Chen, C., Isa, N. A. M., & Liu, X. (2025). A review of convolutional neural network based methods for medical image classification. *Computers in Biology and Medicine*, 185, 109507.

315. Cao, C., Wang, X., Yang, F., Xie, M., Liu, B., Kong, Y., ... & Zhou, Z. (2025). Attention-driven graph convolutional neural networks for mineral prospectivity mapping. *Ore Geology Reviews*, 106554.

316. Ghimire, S., Deo, R. C., Casillas-Pérez, D., Salcedo-Sanz, S., Acharya, R., & Dinh, T. (2025). Electricity demand uncertainty modeling with Temporal

Convolution Neural Network models. *Renewable and Sustainable Energy Reviews*, 209, 115097.

317. Alam, N., Hasan, S., Mashud, G. A., & Bhujel, S. (2025, January). Neural Network for Enhancing Robot-Assisted Rehabilitation: A Systematic Review. In *Actuators* (Vol. 14, No. 1, p. 16). MDPI.

318. Waqas, M., & Humphries, U. W. (2024). A critical review of RNN and LSTM variants in hydrological time series predictions. *MethodsX*, 102946.

319. Al-Selwi, S. M., Hassan, M. F., Abdulkadir, S. J., Muneer, A., Sumiea, E. H., Alqushaibi, A., & Ragab, M. G. (2024). RNN-LSTM: From applications to modeling techniques and beyond—Systematic review. *Journal of King Saud University-Computer and Information Sciences*, 102068.

320. Alam, N., Hasan, S., Mashud, G. A., & Bhujel, S. (2025, January). Neural Network for Enhancing Robot-Assisted Rehabilitation: A Systematic Review. In *Actuators* (Vol. 14, No. 1, p. 16). MDPI.

321. Shobayo, O., & Saatchi, R. (2025). Developments in Deep Learning Artificial Neural Network Techniques for Medical Image Analysis and Interpretation. *Diagnostics*, 15(9), 1072.

322. Ding, H., Hou, H., Wang, L., Cui, X., Yu, W., & Wilson, D. I. (2025). Application of Convolutional Neural Networks and Recurrent Neural Networks in Food Safety. *Foods*, 14(2), 247.

323. Ghimire, S., Deo, R. C., Casillas-Pérez, D., Salcedo-Sanz, S., Acharya, R., & Dinh, T. (2025). Electricity demand uncertainty modeling with Temporal Convolution Neural Network models. *Renewable and Sustainable Energy Reviews*, 209, 115097.

324. Xiong, R., He, Y., Sun, Y., Jia, Y., & Shen, W. (2025). Enhanced electrode-level diagnostics for lithium-ion battery degradation using physics-informed neural networks. *Journal of Energy Chemistry*.

325. Chala, O., Yevsieiev, V., Maksymova, S., & Abu-Jassar, A. (2025). Using the Human Face Recognition Method Based on the MobileNetV2 Neural

Network in Authentication Systems. *Multidisciplinary Journal of Science and Technology*, 5(3), 882-895.

326. Maksymova, S., Yevsieiev, V., Chala, O., & Ababneh, J. (2025). DECISION-MAKING MODEL FOR CONTROLLING A COLLABORATIVE ROBOT-MANIPULATOR BASED ON THE SENSOR FUSION METHOD AND THE RULES OF RULE-BASED SYSTEMS. *Multidisciplinary Journal of Science and Technology*, 5(6), 526-538.

327. Yevsieiev, V., Abu-Jassar, A., Maksymova, S., & Demska, N. (2025). Development of a model for recognizing various objects and tools in a collaborative robot workspace. *ACUMEN: International journal of multidisciplinary research*, 2(1), 224-239.

328. Nevliudov, I., Yevsieiev, V., Baker, J. H., Ahmad, M. A., & Lyashenko, V. (2020). Development of a cyber design modeling declarative Language for cyber physical production systems. *J. Math. Comput. Sci.*, 11(1), 520-542.

329. Khalid, M. S., Yevsieiev, V., Nevliudov, I. S., Lyashenko, V., & Wahid, R. (2022). HMI development automation with GUI elements for object-Oriented programming Languages implementation.

330. GUI Elements and Windows Form Formalization Parameters and Events Method to Automate the Process of Additive Cyber-Design CPPS Development / I. Nevliudov, V. Yevsieiev, V. Lyashenko, M. Ayaz. Ahmad // *Advances in Dynamical Systems and Applications*. – 2021. – Vol. 16(2). – pp. 441-455.

331. Yevsieiev, V. Comparative Analysis of the Characteristics of Mobile Robots and Collaboration Robots Within INDUSTRY 5.0. / V. Yevsieiev, D. Gurin // *Sectoral research XXI : characteristics and features : collection of scientific papers "SCIENTIA" with proceedings of the VI International Scientific and Theoretical Conference, September 8, 2023. - Chicago : European Scientific Platform, 2023. - P. 92-94.*

332. Gurin, D., Yevsieiev, V., Maksymova, S., & Alkhalaileh, A. (2024). Using Convolutional Neural Networks to Analyze and Detect Key Points of Objects in Image. *Multidisciplinary Journal of Science and Technology*, 4(9), 5-15.

333. Yevsieiev, V., Abu-Jassar, A., Maksymova, S., & Gurin, D. (2024). Human Operator Identification in a Collaborative Robot Workspace within the Industry 5.0 Concept. *Multidisciplinary Journal of Science and Technology*, 4(9), 95-105.

334. Yevsieiev, V. USING MULTI-AGENT SYSTEMS IN THE MANAGEMENT OF COLLABORATIVE ROBOTS Vladyslav Yevsieiev. Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки 2025: матеріали II-ої Всеукраїнської конференції, Харків, 16-17 травня 2025.: тези доповідей/[редкол. III Невлюдов (відповідальний редактор)].- Харків:[електронний друк], 2025.–132 с. У збірник включені тези доповідей, які присвячені сучасним автоматизованим технологіям Industry 4.0 та їх впровадження; інформаційні управляючі системи технологічного.

335. Yevsieiev, V., Abu-Jassar, A., Maksymova, S., & Gurin, D. (2024). Human Operator Identification in a Collaborative Robot Workspace within the Industry 5.0 Concept. *Multidisciplinary Journal of Science and Technology*, 4(9), 95-105.

336. Rijwani, T., Kumari, S., Srinivas, R., Abhishek, K., Iyer, G., Vara, H., ... & Gupta, M. (2025). Industry 5.0: A review of emerging trends and transformative technologies in the next industrial revolution. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 19(2), 667-679.

337. Kaswan, M. S., Chaudhary, R., Garza-Reyes, J. A., & Singh, A. (2025). A review of Industry 5.0: from key facets to a conceptual implementation framework. *International Journal of Quality & Reliability Management*, 42(4), 1196-1223.

338. Monferdini, L., Tebaldi, L., & Bottani, E. (2025). From Industry 4.0 to Industry 5.0: Opportunities, Challenges, and Future Perspectives in Logistics. *Procedia Computer Science*, 253, 2941-2950.
339. Passalacqua, M., Pellerin, R., Magnani, F., Doyon-Poulin, P., Del-Aguila, L., Boasen, J., & Léger, P. M. (2025). Human-centred AI in industry 5.0: a systematic review. *International Journal of Production Research*, 63(7), 2638-2669.
340. Rehman, A., & Umar, T. (2025). Literature review: Industry 5.0. Leveraging technologies for environmental, social and governance advancement in corporate settings. *Corporate Governance: The International Journal of Business in Society*, 25(2), 229-251.
341. Narkhede, G. B., Pasi, B. N., Rajhans, N., & Kulkarni, A. (2025). Industry 5.0 and sustainable manufacturing: a systematic literature review. *Benchmarking: An International Journal*, 32(2), 608-635.
342. Sarkar, B. D., Sharma, I., & Shardeo, V. (2025). A multi-method examination of barriers to traceability in Industry 5.0-enabled digital food supply chains. *The International Journal of Logistics Management*, 36(2), 354-380.
343. Safavi Jahromi, G., & Ghazinoory, S. (2025). Clothing industry in transition from Industry 4.0 to Industry 5.0. *The Journal of The Textile Institute*, 116(3), 365-379.
344. Rejeb, A., Rejeb, K., Simske, S., & Süle, E. (2025). Industry 5.0 research: an approach using co-word analysis and BERTopic modeling. *Discover Sustainability*, 6(1), 1-24.
345. Salierno, G., Leonardi, L., & Cabri, G. (2025). Generative AI and Large Language Models in Industry 5.0: Shaping Smarter Sustainable Cities. *Encyclopedia*, 5(1), 30.
346. Puttero, S., Verna, E., Genta, G., & Galetto, M. (2025). Collaborative robots for quality control: an overview of recent studies and emerging trends. *Journal of Intelligent Manufacturing*, 1-37.

347. Figliolini, G., Lanni, C., & Tomassi, L. (2025). Cobot Kinematic Model for Industrial Applications. *Inventions*, 10(3), 37.
348. Berx, N., Decré, W., De Schutter, J., & Pintelon, L. (2025). A harmonious synergy between robotic performance and well-being in human-robot collaboration: A vision and key recommendations. *Annual Reviews in Control*, 59, 100984.
349. Chala, O., Ababneh, J., Yevsieiev, V., & Maksymova, S. (2025). BIO-INSPIRED PRINCIPLES FOR MODELING INFORMATION COLLECTION IN COLLABORATIVE ROBOT ENVIRONMENTS. *Multidisciplinary Journal of Science and Technology*, 5(6), 9-18.
350. Maksymova, S., Yevsieiev, V., Chala, O., & Ababneh, J. (2025). DECISION-MAKING MODEL FOR CONTROLLING A COLLABORATIVE ROBOT-MANIPULATOR BASED ON THE SENSOR FUSION METHOD AND THE RULES OF RULE-BASED SYSTEMS. *Multidisciplinary Journal of Science and Technology*, 5(6), 526-538.
351. Demska, N., Yevsieiev, V., Maksymova, S., & Ababneh, J. (2025). DECISION-MAKING MODEL FOR CONTROLLING A COLLABORATIVE ROBOT-MANIPULATOR BASED ON THE SENSOR FUSION METHOD AND CNN APPROACH TO RULE FORMATION. *Multidisciplinary Journal of Science and Technology*, 5(6), 846-859.
352. Yevsieiev, V., Hamdan, M., Maksymova, S., & Abu-Jassar, A. (2025). A human-centric approach to control collaborative robots within Industry 5.0. *Multidisciplinary Journal of Science and Technology*, 5(5), 351-361.
353. Nevliudov I. Sh. Mathematical Model of Block Process Planning in Systems of Allocation of Task Between People and Collaborative Robots in the Framework of Industries 5.0 / I. Sh. Nevliudov, V. V. Yevsieiev, D. V. Gurin // *Visnyk of Kherson National Technical University*. – 2025. - Vol. 1, № 1(92). - P. 157-163.

354. Babalola, G. T., Gaston, J. M., Trombetta, J., & Tulk Jesso, S. (2024). A systematic review of collaborative robots for nurses: where are we now, and where is the evidence?. *Frontiers in Robotics and AI*, 11, 1398140.

355. Sánchez-Calleja, I., Martínez-Gutiérrez, A., Ferrero-Guillén, R., Díez-González, J., & Perez, H. (2024). Contact system method for the precise interaction between cobots and mobile robots in smart manufacturing. *International Journal of Precision Engineering and Manufacturing*, 25(2), 303-318.

356. Datta, S. D., Islam, M., Sobuz, M. H. R., Ahmed, S., & Kar, M. (2024). Artificial intelligence and machine learning applications in the project lifecycle of the construction industry: A comprehensive review. *Heliyon*.

357. Liu, Z., Yu, X., Liu, N., Liu, C., Jiang, A., & Chen, L. (2025). Integrating AI with detection methods, IoT, and blockchain to achieve food authenticity and traceability from farm-to-table. *Trends in Food Science & Technology*, 104925.

358. Ren, Z., & Li, Z. (2025). The historical achievements and future prospects of AI/AR/VR/XR/robot art under the current development situation of the metaverse. *Mediterranean Archaeology and Archaeometry*, 25(3).

359. Tanwar, R. S., & Jhavar, S. (2025). Multi-wire additive manufacturing: A comprehensive review on materials, microstructure, methodological advances, and applications. *Results in Engineering*, 104814.

360. Horvat, D., Jäger, A., & Lerch, C. M. (2025). Fostering innovation by complementing human competences and emerging technologies: an industry 5.0 perspective. *International Journal of Production Research*, 63(3), 1126-1149.

361. Kaswan, M. S., Chaudhary, R., Garza-Reyes, J. A., & Singh, A. (2025). A review of Industry 5.0: from key facets to a conceptual implementation framework. *International Journal of Quality & Reliability Management*, 42(4), 1196-1223.

362. Monferdini, L., Tebaldi, L., & Bottani, E. (2025). From Industry 4.0 to Industry 5.0: Opportunities, Challenges, and Future Perspectives in Logistics. *Procedia Computer Science*, 253, 2941-2950.

363. Li, L., & Duan, L. (2025). Human centric innovation at the heart of industry 5.0—exploring research challenges and opportunities. *International Journal of Production Research*, 1-33.

364. Rehman, A., & Umar, T. (2025). Literature review: Industry 5.0. Leveraging technologies for environmental, social and governance advancement in corporate settings. *Corporate Governance: The International Journal of Business in Society*, 25(2), 229-251.

ДОДАТОК А

Основні функції бібліотеки cv2

1. `cv2.imread()` - завантажує зображення з файлу.

- синтаксис:

```
cv2.imread(filename, flags)
```

параметри:

- `filename` (str): шлях до файлу зображення;
- `flags` (int): прапори для завантаження зображення.

Можливі значення:

- `cv2.IMREAD_COLOR`: завантажує кольорове зображення (за замовчуванням).
- `cv2.IMREAD_GRAYSCALE`: завантажує чорно-біле зображення.
- `cv2.IMREAD_UNCHANGED`: завантажує зображення з усіма каналами (включно з альфа-каналом).

2. `cv2.imshow()` - відображає зображення у вікні.

- синтаксис:

```
cv2.imshow(window_name, image)
```

параметри:

- `window_name` (str): назва вікна для відображення.
- `image` (numpy.ndarray): зображення для відображення.

3. `cv2.waitKey()` - очікує натискання клавіші.

- синтаксис:

```
cv2.waitKey(delay)
```

параметри:

- `delay` (int): час очікування в мілісекундах. Якщо встановлено 0, очікує нескінченно.

4. `cv2.destroyAllWindows()` - закриває всі відкриті вікна, створені функцією `cv2.imshow()`.

- синтаксис:

```
cv2.destroyAllWindows()
```

5. `cv2.imwrite()` - зберігає зображення на диск.

- синтаксис**:

`cv2.imwrite(filename, image)``

параметри:

- `filename`` (str): шлях до файлу для збереження зображення.
- `image`` (numpy.ndarray): зображення для збереження.

6. `cv2.cvtColor()`` - перетворює зображення з одного кольорового простору в інший.

- синтаксис:

`cv2.cvtColor(src, code)``

параметри:

- `src`` (numpy.ndarray): вхідне зображення.
- `code`` (int): тип перетворення кольору (наприклад, `cv2.COLOR_BGR2GRAY``, `cv2.COLOR_BGR2RGB``).

7. `cv2.resize()`` - змінює розмір зображення.

- синтаксис:

`cv2.resize(src, dsize, fx=0, fy=0, interpolation=cv2.INTER_LINEAR)``

параметри:

- `src`` (numpy.ndarray): вхідне зображення.
- `dsize`` (tuple): новий розмір зображення (ширина, висота). Якщо це значення задано, `fx`` та `fy`` ігноруються.
- `fx`` (float): масштабний фактор по ширині.
- `fy`` (float): масштабний фактор по висоті.
- `interpolation`` (int): метод інтерполяції (наприклад, `cv2.INTER_LINEAR``, `cv2.INTER_NEAREST``).

8. `cv2.GaussianBlur()`` - застосовує гауссове розмиття до зображення.

- синтаксис:

`cv2.GaussianBlur(src, ksize, sigmaX, sigmaY=0)``

параметри:

- `src`` (numpy.ndarray): вхідне зображення.
- `ksize`` (tuple): розмір ядра фільтра (ширина, висота), має бути непарним числом.
- `sigmaX`` (float): стандартне відхилення по осі X.
- `sigmaY`` (float): стандартне відхилення по осі Y (якщо дорівнює 0, то використовується значення `sigmaX``).

9. `cv2.Canny()`` - здійснює виявлення контурів на зображенні за допомогою методу Кенні.

- синтаксис:

```
`cv2.Canny(image, threshold1, threshold2)`
```

параметри:

- ``image`` (numpy.ndarray): вхідне зображення (сірий масштаб).
- ``threshold1`` (float): нижній поріг для виявлення границь.
- ``threshold2`` (float): верхній поріг для виявлення границь.

10. ``cv2.findContours()`` - знаходить контури на бінарному зображенні.

синтаксис:

```
`cv2.findContours(image, mode, method)`
```

параметри:

- ``image`` (numpy.ndarray): вхідне бінарне зображення.
- ``mode`` (int): режим пошуку контурів (наприклад, ``cv2.RETR_EXTERNAL``, ``cv2.RETR_TREE``).
- ``method`` (int): метод апроксимації контурів (наприклад, ``cv2.CHAIN_APPROX_SIMPLE``, ``cv2.CHAIN_APPROX_NONE``).

ДОДАТОК Б

Основні функції бібліотеки NumPy

1. `numpy.array()` - створює масив на основі вхідних даних.

- синтаксис:

```
numpy.array(object, dtype=None, copy=True, order='K',
            subok=False, ndmin=0)
```

параметри:

- `object` (array-like): вхідні дані (список, кортеж, інший масив).
- `dtype` (data-type, optional): бажаний тип даних елементів масиву (наприклад, `numpy.float32`, `numpy.int32`).
- `copy` (bool, optional): чи потрібно копіювати вхідні дані (за замовчуванням `True`).
- `order` (optional): спосіб розташування даних у пам'яті ('C' – рядковий порядок, 'F' – стовпцевий порядок, 'A' або 'K' – будь-який, в залежності від порядку вхідних даних).
- `ndmin` (int, optional): мінімальна кількість вимірів у масиві (за замовчуванням 0).

2. `numpy.zeros()` - створює масив, заповнений нулями.

- синтаксис:

```
numpy.zeros(shape, dtype=float, order='C')
```

параметри:

- `shape` (int або tuple): форма масиву, наприклад (2, 3) для матриці 2x3.
- `dtype` (data-type, optional): тип елементів масиву (за замовчуванням `float`).
- `order` (optional): порядок зберігання даних у пам'яті ('C' – рядковий, 'F' – стовпцевий).

3. `numpy.ones()` - створює масив, заповнений одиницями.

- синтаксис:

```
numpy.ones(shape, dtype=float, order='C')
```

- параметри:

- `shape` (int або tuple): форма масиву.
- `dtype` (data-type, optional): тип елементів масиву.
- `order` (optional): порядок зберігання даних.

4. `numpy.eye()` - створює одиничну матрицю (діагональна матриця з одиницями на головній діагоналі).

- синтаксис:

```
numpy.eye(N, M=None, k=0, dtype=float, order='C')
```

- параметри:

- `'N'` (int): кількість рядків.
- `'M'` (int, optional): кількість стовпців. Якщо не вказано, то `'M=N'`.
- `'k'` (int, optional): індекс діагоналі (0 - головна діагональ, 1 - діагональ над головною і т.д.).
- `'dtype'` (data-type, optional): тип елементів масиву.
- `'order'` (optional): порядок зберігання даних.

5. `numpy.arange()` - створює послідовність чисел з певним кроком.

- синтаксис:

```
numpy.arange([start, ]stop, [step, ]dtype=None)
```

- параметри:

- `'start'` (optional): початкове значення (за замовчуванням 0).
- `'stop'` (int): кінцеве значення (не включається в результат).
- `'step'` (optional): крок між значеннями (за замовчуванням 1).
- `'dtype'` (optional): тип елементів.

6. `numpy.linspace()` - створює послідовність рівновіддалених чисел у заданому діапазоні.

- синтаксис:

```
numpy.linspace(start, stop, num=50, endpoint=True, retstep=False, dtype=None, axis=0)
```

- параметри:

- `'start'` (float): початкове значення.
- `'stop'` (float): кінцеве значення.
- `'num'` (int, optional): кількість елементів у послідовності (за замовчуванням 50).
- `'endpoint'` (bool, optional): чи включати кінцеве значення (за замовчуванням `'True'`).
- `'retstep'` (bool, optional): чи повертати крок між значеннями (за замовчуванням `'False'`).
- `'dtype'` (optional): тип елементів.
- `'axis'` (int, optional): вісь, вздовж якої створюється масив.

7. `numpy.random.rand()` - створює масив із випадковими числами з рівномірного розподілу в діапазоні від 0 до 1.

- синтаксис:

```
`numpy.random.rand(d0, d1, ..., dn)`
```

параметри:

- ``d0, d1, ..., dn``: розміри масиву.

8. ``numpy.reshape()`` - змінює форму масиву без зміни його даних.

- синтаксис:

```
`numpy.reshape(a, newshape, order='C')`
```

параметри:

- ``a`` (`array_like`): вхідний масив.

- ``newshape`` (`int` або `tuple`): нова форма масиву.

- ``order`` (`optional`): порядок зберігання даних ('C' - рядковий порядок, 'F' - стовпцевий).

9. ``numpy.transpose()`` - транспонує масив (мінє місцями осі).

- синтаксис:

```
`numpy.transpose(a, axes=None)`
```

- параметри:

- ``a`` (`array_like`): вхідний масив.

- ``axes`` (`tuple, optional`): порядок осей (якщо не вказано, осі міняються місцями).

10. ``numpy.dot()`` - обчислює скалярний добуток двох масивів.

- синтаксис:

```
`numpy.dot(a, b, out=None)`
```

- параметри:

- ``a``, ``b`` (`array_like`): вхідні масиви для множення.

- ``out`` (`optional`): місце для збереження результату.

11. ``numpy.sum()`` - обчислює суму елементів масиву.

- синтаксис:

```
`numpy.sum(a, axis=None, dtype=None, out=None, keepdims=False)`
```

- параметри:

- ``a`` (`array_like`): вхідний масив.

- ``axis`` (`optional`): вісь або осі, по яких обчислюється сума.

- ``dtype`` (`optional`): тип елементів результату.

- ``out`` (`optional`): масив для збереження результату.

- ``keepdims`` (`optional`): чи зберігати початкові розміри масиву.

12. `numpy.mean()` - обчислює середнє значення елементів масиву.

- синтаксис:

```
numpy.mean(a, axis=None, dtype=None, out=None, keepdims=False)
```

- параметри:

- `a` (`array_like`): вхідний масив.
- `axis` (optional): вісь або осі, по яких обчислюється середнє значення.
- `dtype` (optional): тип результату.
- `out` (optional): масив для збереження результату.
- `keepdims` (optional): чи зберігати початкові розміри масиву.

13. `numpy.std()` - обчислює стандартне відхилення елементів масиву.

- синтаксис:

```
numpy.std(a, axis=None, dtype=None, out=None, ddof=0, keepdims=False)
```

- параметри:

- `a` (`array_like`): вхідний масив.
- `axis` (optional): вісь або осі.
- `dtype` (optional): тип результату.
- `ddof` (optional): градуси свободи (за замовчуванням 0).
- `out` (optional): масив для збереження результату.

ДОДАТОК В

Основні функції бібліотеки TensorFlow

1. `tf.constant()` - створює тензор, значення якого фіксоване та незмінне.

- синтаксис:

```
tf.constant(value, dtype=None, shape=None, name='Const')
```

- параметри:

- `value`: список, масив або скаляр, з якого створюється тензор.
- `dtype` (optional): тип елементів тензора (за замовчуванням обирається автоматично на основі `value`).
- `shape` (optional): форма тензора (якщо не вказана, то визначається автоматично).
- `name` (optional): ім'я операції.

2. `tf.Variable()` - створює змінний тензор, значення якого можна змінювати під час обчислень.

- синтаксис:

```
tf.Variable(initial_value, trainable=True, validate_shape=True, name=None, dtype=None)
```

- параметри:

- `initial_value`: початкове значення тензора.
- `trainable` (optional): чи буде змінна тренуватись у процесі навчання (за замовчуванням `True`).
- `validate_shape` (optional): чи перевіряти форму змінної.
- `name` (optional): ім'я змінної.
- `dtype` (optional): тип елементів.

3. `tf.placeholder()` (для TensorFlow 1.x) - створює тензор-заповнювач, який очікує на вхідні дані під час виконання графу. (У TensorFlow 2.x більше не використовується, замість нього використовуються тензори та функції Python).

- синтаксис:

```
tf.placeholder(dtype, shape=None, name=None)
```

- параметри:

- `dtype`: тип даних, які будуть передані в заповнювач.
- `shape` (optional): форма тензора.
- `name` (optional): ім'я заповнювача.

4. `tf.matmul()` - обчислює матричний добуток двох тензорів.

- синтаксис:

```
tf.matmul(a, b, transpose_a=False, transpose_b=False, adjoint_a=False,
adjoint_b=False, a_is_sparse=False, b_is_sparse=False, name=None)
```

- параметри:

- `a`, `b`: тензори для множення.
- `transpose_a` (optional): чи транспонувати тензор `a` перед множенням.
- `transpose_b` (optional): чи транспонувати тензор `b`.
- `adjoint_a` (optional): чи транспонувати і виконати комплексно-спряжену операцію над `a`.
- `adjoint_b` (optional): аналогічна операція для `b`.
- `a_is_sparse`, `b_is_sparse` (optional): вказує, чи є тензори розрідженими.
- `name` (optional): ім'я операції.

5. `tf.reshape()` - змінює форму тензора без зміни його даних.

- синтаксис:

```
tf.reshape(tensor, shape, name=None)
```

- параметри:

- `tensor`: вхідний тензор.
- `shape`: нова форма тензора.
- `name` (optional): ім'я операції.

6. `tf.reduce_mean()` - обчислює середнє значення елементів тензора по заданій осі або по всіх елементах.

- синтаксис:

```
tf.reduce_mean(input_tensor, axis=None, keepdims=False, name=None)
```

- параметри:

- `input_tensor`: вхідний тензор.
- `axis` (optional): вісь або осі, по яких буде обчислюватись середнє значення.
- `keepdims` (optional): чи зберігати початкові розміри тензора.
- `name` (optional): ім'я операції.

7. `tf.reduce_sum()` - обчислює суму елементів тензора по заданій осі або по всіх елементах.

- синтаксис:

```
tf.reduce_sum(input_tensor, axis=None, keepdims=False, name=None)
```

- параметри:

- `input_tensor`: вхідний тензор.

- ``axis`` (optional): вісь або осі, по яких буде обчислюватись сума.
- ``keepdims`` (optional): чи зберігати початкові розміри тензора.
- ``name`` (optional): ім'я операції.

8. ``tf.nn.relu()`` - застосовує функцію активації ReLU (Rectified Linear Unit), яка обнуляє всі від'ємні значення тензора.

- синтаксис:
``tf.nn.relu(features, name=None)``
- параметри:
 - ``features``: вхідний тензор.
 - ``name`` (optional): ім'я операції.

9. ``tf.nn.softmax()`` - застосовує функцію активації Softmax до останнього виміру тензора, нормуючи його так, щоб сума елементів дорівнювала 1.

- синтаксис:
``tf.nn.softmax(logits, axis=-1, name=None)``
- параметри:
 - ``logits``: вхідний тензор.
 - ``axis`` (optional): вісь, по якій застосовується Softmax (за замовчуванням -1, тобто остання).
 - ``name`` (optional): ім'я операції.

10. ``tf.keras.Model()`` - створює модель нейронної мережі в стилі Keras, яка дозволяє створювати, компілювати та тренувати моделі.

- синтаксис:
``tf.keras.Model(inputs, outputs, name=None)``
- параметри:
 - ``inputs``: вхідні тензори моделі (може бути кілька).
 - ``outputs``: вихідні тензори моделі (також може бути кілька).
 - ``name`` (optional): ім'я моделі.

11. ``model.compile()`` - компілює модель перед її тренуванням. Це включає вибір оптимізатора, функції втрат і метрик для оцінювання.

- синтаксис:
``model.compile(optimizer, loss, metrics=None, loss_weights=None, weighted_metrics=None, run_eagerly=None)``
- параметри:
 - ``optimizer``: оптимізатор для навчання (наприклад, ``adam``, ``sgd``).
 - ``loss``: функція втрат (наприклад, ``mse``, ``binary_crossentropy``).

- ``metrics`` (optional): список метрик для оцінки роботи моделі.
- ``loss_weights`` (optional): ваги для кожної функції втрат (якщо модель має кілька виходів).
- ``weighted_metrics`` (optional): метрики для оцінки з вагами.
- ``run_eagerly`` (optional): чи виконувати модель в режимі `eager` (без попередньої компіляції графу).

12. ``model.fit()`` - навчає модель на основі тренувальних даних.

- синтаксис:

```
`model.fit(x, y, batch_size=None, epochs=1, verbose=1, validation_data=None)`
```

- параметри:

- ``x``: вхідні дані (може бути масив або тензор).
- ``y``: цільові значення (масив або тензор).
- ``batch_size`` (optional): розмір батчу.
- ``epochs`` (optional): кількість епох тренування.
- ``verbose`` (optional): рівень деталізації під час навчання (0, 1 або 2).
- ``validation_data`` (optional): валідаційні дані.

13. ``model.evaluate()`` - оцінює модель на тестових даних.

- синтаксис:

```
`model.evaluate(x, y, batch_size=None, verbose=1)`
```

- параметри:

- ``x``: вхідні дані для оцінки.
- ``y``: цільові дані.
- ``batch_size`` (optional): розмір батчу.
- ``verbose`` (optional): рівень деталізації.

ДОДАТОК Г

Основні функції бібліотеки MediaPipe

1. `mp.solutions.hands` - цей модуль використовується для розпізнавання та відстеження рук у відеопотоці.

- клас:

`mp.solutions.hands.Hands()`

- синтаксис:

`mp.solutions.hands.Hands(static_image_mode=False, max_num_hands=2, min_detection_confidence=0.5, min_tracking_confidence=0.5)`

- параметри:

- `static_image_mode`: якщо `True`, модель працює в статичному режимі, обробляючи кожне зображення окремо. Якщо `False`, модель використовує попередні кадри для покращення продуктивності у відео.

- `max_num_hands`: максимальна кількість рук, які можна розпізнати.

- `min_detection_confidence`: мінімальна ймовірність для розпізнавання рук.

- `min_tracking_confidence`: мінімальна ймовірність для успішного трекінгу (слідкування за рухом рук).

2. `mp.solutions.face_mesh` - цей модуль використовується для побудови сітки (mesh) на обличчі, що дозволяє виявляти ключові точки на ньому.

- клас:

`mp.solutions.face_mesh.FaceMesh()`

- синтаксис:

`mp.solutions.face_mesh.FaceMesh(static_image_mode=False, max_num_faces=1, min_detection_confidence=0.5, min_tracking_confidence=0.5)`

- параметри:

- `static_image_mode`: обробка кожного зображення як статичного, якщо `True`.

- `max_num_faces`: максимальна кількість облич, які можна розпізнати.

- `min_detection_confidence`: мінімальна ймовірність для розпізнавання обличчя.

- `min_tracking_confidence`: мінімальна ймовірність для трекінгу ключових точок на обличчі.

3. `mp.solutions.pose` - цей модуль використовується для відстеження ключових точок тіла та пози людини.

- клас:

```
`mp.solutions.pose.Pose()`
```

- синтаксис:

```
`mp.solutions.pose.Pose(static_image_mode=False, model_complexity=1,
smooth_landmarks=True, enable_segmentation=False,
min_detection_confidence=0.5, min_tracking_confidence=0.5)`
```

- параметри:

- ``static_image_mode``: обробка кожного зображення незалежно.
- ``model_complexity``: складність моделі (0, 1 або 2), впливає на точність і швидкість.
- ``smooth_landmarks``: згладжування ключових точок для плавного трекінгу.
- ``enable_segmentation``: якщо ``True``, активує сегментацію фону.
- ``min_detection_confidence``: мінімальна ймовірність для розпізнавання пози.
- ``min_tracking_confidence``: мінімальна ймовірність для трекінгу пози.

4. **``mp.solutions.holistic``** - цей модуль поєднує відстеження рук, обличчя та тіла для повного розпізнавання рухів людини.

- клас:

```
`mp.solutions.holistic.Holistic()`
```

- синтаксис:

```
`mp.solutions.holistic.Holistic(static_image_mode=False,
model_complexity=1, smooth_landmarks=True, enable_segmentation=False,
min_detection_confidence=0.5, min_tracking_confidence=0.5)`
```

- параметри:

- ``static_image_mode``: режим обробки статичних зображень.
- ``model_complexity``: складність моделі для різних частин тіла.
- ``smooth_landmarks``: чи згладжувати ключові точки для плавного руху.
- ``enable_segmentation``: чи вмикати сегментацію фону.
- ``min_detection_confidence``: ймовірність для розпізнавання людини.
- ``min_tracking_confidence``: мінімальна ймовірність для трекінгу.

5. **``mp.solutions.drawing_utils``** - цей модуль містить інструменти для малювання ключових точок та сіток на зображеннях або відео.

- функція:

```
`mp.solutions.drawing_utils.draw_landmarks()`
```

- синтаксис:

```
`mp.solutions.drawing_utils.draw_landmarks(image, landmark_list,
connections=None, landmark_drawing_spec=None,
connection_drawing_spec=None)`
```

- параметри:
- ``image``: зображення або кадр відео, на якому ставляться точки.
- ``landmark_list``: список ключових точок (landmarks).
- ``connections`` (optional): зв'язки між ключовими точками для відображення сітки.
- ``landmark_drawing_spec`` (optional): специфікація для малювання точок (колір, товщина).
- ``connection_drawing_spec`` (optional): специфікація для малювання зв'язків між точками.

6. ``mp.solutions.objectron`` - цей модуль використовується для тривимірного розпізнавання об'єктів (3D Object Detection), таких як коробки або предмети.

- клас:
- ``mp.solutions.objectron.Objectron()``
- синтаксис:
- ``mp.solutions.objectron.Objectron(static_image_mode=False, max_num_objects=5, min_detection_confidence=0.5, min_tracking_confidence=0.5, model_name='Shoe')``
- параметри:
- ``static_image_mode``: режим обробки статичних зображень.
- ``max_num_objects``: максимальна кількість об'єктів, які можна розпізнати.
- ``min_detection_confidence``: мінімальна ймовірність для розпізнавання об'єктів.
- ``min_tracking_confidence``: мінімальна ймовірність для трекінгу об'єктів.
- ``model_name``: ім'я моделі (наприклад, ``Shoe``, ``Chair``, ``Cup``).

7. ``mp.solutions.face_detection`` - модуль для розпізнавання облич на зображеннях або відео.

- клас:
- ``mp.solutions.face_detection.FaceDetection()``
- синтаксис:
- ``mp.solutions.face_detection.FaceDetection(min_detection_confidence=0.5)``
- параметри:
- ``min_detection_confidence``: мінімальна ймовірність для успішного розпізнавання обличчя.

ДОДАТОК Е

«Код програми розпізнавання людини в робочій зоні колаборативного робота
маніпулятора на базі глибокої нейронної мережі MobileNetV2 в режимі
реального часу»

```
import cv2
import mediapipe as mp

# Ініціалізація Mediapipe для розпізнавання поз
mp_pose = mp.solutions.pose
mp_drawing = mp.solutions.drawing_utils

# Налаштування для розпізнавання поз
pose = mp_pose.Pose(
    static_image_mode=False,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5
)

# Відкриття відеопотоку з камери
cap = cv2.VideoCapture(1)

while True:
    # Зчитування кадру з відео
    ret, frame = cap.read()
    if not ret:
        break

    # Перетворення кадру в RGB
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = pose.process(rgb_frame)

    # Перевірка наявності людського тіла у кадрі
    if results.pose_landmarks:
        # Малювання точок і ліній на зображенні
        mp_drawing.draw_landmarks(
            image=frame,
            landmark_list=results.pose_landmarks,
            connections=mp_pose.POSE_CONNECTIONS,
            landmark_drawing_spec=mp_drawing.DrawingSpec(color=(0, 255, 0),
                thickness=2, circle_radius=2),
```

```
        connection_drawing_spec=mp_drawing.DrawingSpec(color=(255, 0, 0),  
thickness=2)  
    )
```

```
# Відображення відео  
cv2.imshow('Pose Detection', frame)
```

```
# Вихід з циклу при натисканні клавіші 'q'  
if cv2.waitKey(1) & 0xFF == ord('q'):  
    break
```

```
# Закриття відеопотоку та вікон  
cap.release()  
cv2.destroyAllWindows()
```

ДОДАТОК К

«Код дослідження впливу частоти обробки кадрів при ідентифікації об'єктів на базі моделі нейронної мережі MobileNetV2 для мобільного робота»

```

import cv2
import numpy as np
import tensorflow as tf
import time

# Завантаження моделі MobileNetV2 для розпізнавання об'єктів
model = tf.saved_model.load(
    r"C:\Users\Vladyslav\.cache\kagglehub\models\tensorflow\ssd-mobilenet-
v2\tensorFlow2\fpnlite-320x320\1")

# Список класів об'єктів
class_names = [
    'N/A', 'person', 'bicycle', 'car', 'motorcycle', 'airplane',
    'bus', 'train', 'truck', 'boat', 'traffic light', 'fire hydrant',
    'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog',
    'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe',
    'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee',
    'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat',
    'baseball glove', 'skateboard', 'surfboard', 'tennis racket',
    'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl',
    'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot',
    'hot dog', 'pizza', 'donut', 'cake', 'chair', 'couch', 'potted plant',
    'bed', 'dining table', 'toilet', 'TV', 'laptop', 'mouse',
    'remote', 'keyboard', 'cell phone', 'microwave', 'oven', 'toaster',
    'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors',
    'teddy bear', 'hair drier', 'toothbrush'
]

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

# Ініціалізація камери
cap = cv2.VideoCapture(0)

```

```

# Зменшення частоти обробки кадрів
frame_skip = 5 # Обробляти кожний третій кадр

while True:
    ret, frame = cap.read()
    if not ret:
        break

    # Зменшити розмір зображення для швидшої обробки
    input_frame = cv2.resize(frame, (320, 320))

    # Перетворити зображення в uint8
    input_frame = np.asarray(input_frame, dtype=np.uint8)

    # Виконати розпізнавання об'єктів кожен 'frame_skip' кадр
    if int(cap.get(cv2.CAP_PROP_POS_FRAMES)) % frame_skip == 0:
        start_time = time.time()
        detections = detect_objects(input_frame)
        end_time = time.time()
        print(f"Inference Time: {end_time - start_time:.3f}s")

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

    # Виведення результатів на кадр
    for i in range(num_detections):
        if detection_scores[i] > 0.5:
            box = detection_boxes[i]
            y1, x1, y2, x2 = box
            y1, x1, y2, x2 = int(y1 * frame.shape[0]), int(x1 * frame.shape[1]), int(y2 *
frame.shape[0]), int(
            x2 * frame.shape[1])

            # Отримати назву класу
            class_id = detection_classes[i]
            class_name = class_names[class_id] if class_id < len(class_names) else
'Unknown'

    # Виведення результатів на кадр
    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

```

```
    label = f"{class_name}: {detection_scores[i]*100:.2f}%"
    cv2.putText(frame, label, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.9, (0, 255, 0), 2)

cv2.imshow('Video', frame)

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

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

ДОДАТОК Л

«Код програми реалізації алгоритму SAMShift для трекінгу людини в
робочій зоні колаборативного робота»

```
import cv2
import numpy as np
import tensorflow as tf

# Завантаження моделі MobileNetV2 для розпізнавання об'єктів
model = tf.saved_model.load(
    r"C:\Users\Vladyslav\.cache\kagglehub\models\tensorflow\ssd-mobilenet-
v2\tensorFlow2\fpnlite-320x320\1")

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

# Ініціалізація камери
cap = cv2.VideoCapture(1)

# Змінна для зберігання трекінг-області
tracking_window = None

while True:
    ret, frame = cap.read()
    if not ret:
        break

    # Змінити розмір кадру до потрібного для моделі
    input_frame = cv2.resize(frame, (320, 320))

    # Виконати розпізнавання об'єктів
    detections = detect_objects(input_frame)

    # Обробка результатів розпізнавання
    num_detections = int(detections['num_detections'][0])
```

```

detection_classes = detections['detection_classes'][0].numpy().astype(np.int64)
detection_boxes = detections['detection_boxes'][0].numpy()
detection_scores = detections['detection_scores'][0].numpy()

# Якщо трекінг-область не ініціалізована, знайти перший об'єкт "Person" для
трекінгу
if tracking_window is None:
    for i in range(num_detections):
        if detection_scores[i] > 0.5 and detection_classes[i] == 1: # Клас 1 відповідає
людині
            box = detection_boxes[i]
            y1, x1, y2, x2 = box
            y1, x1, y2, x2 = int(y1 * frame.shape[0]), int(x1 * frame.shape[1]), int(y2 *
frame.shape[0]), int(
                x2 * frame.shape[1])
            tracking_window = (x1, y1, x2 - x1, y2 - y1)
            break

# Виконати трекінг з CAMShift, якщо трекінг-область ініціалізована
if tracking_window is not None:
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    roi = frame[tracking_window[1]:tracking_window[1] + tracking_window[3],
        tracking_window[0]:tracking_window[0] + tracking_window[2]]

    # Переконайтеся, що roi має розмірність 3 канали
    if roi.size > 0:
        roi_hsv = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)
        roi_hist = cv2.calcHist([roi_hsv], [0], None, [180], [0, 180])
        cv2.normalize(roi_hist, roi_hist, 0, 255, cv2.NORM_MINMAX)

    # Трекінг за допомогою CAMShift
    dst = cv2.calcBackProject([hsv], [0], roi_hist, [0, 180], 1)
    ret, tracking_window = cv2.CamShift(dst, tracking_window, (255, 0, 0))

    # Виведення результатів на кадр
    pts = cv2.boxPoints(ret)
    pts = np.intp(pts) # Зміна з np.int0 на np.intp
    cv2.polylines(frame, [pts], True, (255, 0, 0), 2)
    cv2.putText(frame, 'Tracking', (tracking_window[0], tracking_window[1] -
10), cv2.FONT_HERSHEY_SIMPLEX, 0.9,
        (255, 0, 0), 2)

# Відображення кадру
cv2.imshow('Video', frame)

```

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

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

ДОДАТОК М

«Код програми використання фільтра Калмана для представлення ймовірнісних моделей визначення місцезнаходження людини в робочій зоні колаборативного робота»

```
import cv2
import numpy as np

def initialize_kalman():
    kalman = cv2.KalmanFilter(4, 2)

    kalman.statePre = np.array([0, 0, 0, 0], dtype=np.float32)
    kalman.statePost = np.array([0, 0, 0, 0], dtype=np.float32)

    kalman.transitionMatrix = np.array([[1, 0, 1, 0],
                                         [0, 1, 0, 1],
                                         [0, 0, 1, 0],
                                         [0, 0, 0, 1]], dtype=np.float32)

    kalman.measurementMatrix = np.array([[1, 0, 0, 0],
                                          [0, 1, 0, 0]], dtype=np.float32)

    kalman.processNoiseCov = np.array([[1e-2, 0, 0, 0],
                                        [0, 1e-2, 0, 0],
                                        [0, 0, 1e-2, 0],
                                        [0, 0, 0, 1e-2]], dtype=np.float32)

    kalman.measurementNoiseCov = np.array([[1e-1, 0],
                                           [0, 1e-1]], dtype=np.float32)

    kalman.errorCovPost = np.eye(4, dtype=np.float32)

    return kalman

def main(camera_index):
    # Ініціалізація фільтра Калмана
    kalman = initialize_kalman()

    # Відкриття відеопотоку з обраної камери
```

```

cap = cv2.VideoCapture(camera_index)

while True:
    ret, frame = cap.read()
    if not ret:
        print(f'Не вдалося зчитати кадри з камери з індексом {camera_index}.')
        break

    # Перетворення зображення в сірий формат
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Простий детектор обличчя для демонстрації
    face_cascade = cv2.CascadeClassifier(cv2.data.harcascades +
'haarcascade_frontalface_default.xml')
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
minSize=(30, 30))

    if len(faces) > 0:
        x, y, w, h = faces[0]
        measurement = np.array([[x + w / 2], [y + h / 2]], dtype=np.float32)

        # Оновлення фільтра Калмана
        kalman.correct(measurement)
        prediction = kalman.predict()

        # Візуалізація результатів
        pred_x, pred_y = int(prediction[0]), int(prediction[1])
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
        cv2.circle(frame, (pred_x, pred_y), 5, (0, 255, 0), -1)

    cv2.imshow('Kalman Filter Tracking', frame)

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

cap.release()
cv2.destroyAllWindows()

if __name__ == "__main__":
    # Введіть індекс камери, яку ви хочете використовувати
    camera_index = int(input("Введіть індекс камери (наприклад, 0 для першої
камери): "))
    main(camera_index)

```

ДОДАТОК Н

«Код програми використання Convolutional Neural Networks (CNNs) для аналізу та виявлення ключових точок об'єктів на зображеннях»

```
import cv2
import mediapipe as mp

# Ініціалізація Mediapipe для розпізнавання жестів
mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils

# Налаштування для розпізнавання жестів
hands = mp_hands.Hands(min_detection_confidence=0.7,
min_tracking_confidence=0.5)

# Відкриття відеопотоку з камери
cap = cv2.VideoCapture(1)

while True:
    # Зчитування кадру з відео
    ret, frame = cap.read()
    if not ret:
        break

    # Перетворення кадру в RGB
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = hands.process(rgb_frame)

    # Перевірка наявності рук у кадрі
    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            # Малювання точок на зображенні
            mp_drawing.draw_landmarks(frame, hand_landmarks,
mp_hands.HAND_CONNECTIONS)

            # Доступ до координат ключових точок
            for id, landmark in enumerate(hand_landmarks.landmark):
                h, w, c = frame.shape
                cx, cy = int(landmark.x * w), int(landmark.y * h)
                print(f'Landmark {id}: ({cx}, {cy})')
```

```
# Відображення відео
cv2.imshow('Hand Gesture Recognition', frame)

# Вихід з циклу при натисканні клавіші 'q'
if cv2.waitKey(1) & 0xFF == ord('q'):
    break

# Закриття відеопотоку та вікон
cap.release()
cv2.destroyAllWindows()
```

ДОДАТОК П

«Код програми використання рекурентних нейронних мереж LSTM для прогнозування траєкторії переміщення рук людини в робочій зоні колаборативного робота маніпулятора»

```
import cv2
import numpy as np
import mediapipe as mp
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from collections import deque

# Налаштування MediaPipe для виявлення руки
mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils

# Параметри
sequence_length = 10
prediction_length = 30

# Ініціалізація моделі LSTM
model = Sequential([
    LSTM(50, activation='relu', input_shape=(sequence_length, 2),
        return_sequences=True),
    LSTM(50, activation='relu'),
    Dense(2)
])
model.compile(optimizer='adam', loss='mse')

# Ініціалізація відео
cap = cv2.VideoCapture(1)
hands = mp_hands.Hands(min_detection_confidence=0.2,
    min_tracking_confidence=0.2)

# Ініціалізація історії координат руки
hand_landmarks_history = deque(maxlen=sequence_length + prediction_length)
trajectory = []

while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
```

```

break

# Перетворення зображення для MediaPipe
rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
results = hands.process(rgb_frame)

# Обробка результатів
if results.multi_hand_landmarks:
    for hand_landmarks in results.multi_hand_landmarks:
        # Вибір точки на руці для відстеження (наприклад, кінчик індексного
        пальця)
        ih, iw, _ = frame.shape
        x, y = hand_landmarks.landmark[8].x, hand_landmarks.landmark[8].y # Точка
        8 - кінчик індексного пальця
        cv2.circle(frame, (int(x * iw), int(y * ih)), 5, (0, 255, 0), -1)

        # Оновлення історії координат руки
        hand_landmarks_history.append((x, y))

        if len(hand_landmarks_history) >= sequence_length:
            # Підготовка даних для прогнозування
            input_sequence = np.array([hand_landmarks_history[i] for i in range(-
            sequence_length, 0)])
            input_sequence = np.expand_dims(input_sequence, axis=0)

            # Прогнозування
            predicted = model.predict(input_sequence)[0]
            trajectory = np.array([hand_landmarks_history[i] for i in range(-
            sequence_length, 0)] + [predicted])

            # Візуалізація прогнозованої траєкторії
            for j in range(len(trajectory) - 1):
                start_point = (int(trajectory[j][0] * iw), int(trajectory[j][1] * ih))
                end_point = (int(trajectory[j + 1][0] * iw), int(trajectory[j + 1][1] * ih))
                cv2.line(frame, start_point, end_point, (0, 0, 255), 2)

            # Малювання скелету руки на кадрі
            mp_drawing.draw_landmarks(frame, hand_landmarks,
            mp_hands.HAND_CONNECTIONS)

            # Відображення відео
            cv2.imshow('Hand Trajectory Prediction', frame)

            # Вихід за натисканням клавіші 'q'

```

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

```
# Завершення роботи  
cap.release()  
cv2.destroyAllWindows()
```

ДОДАТОК Р

«Код програми ідентифікації людини оператора в робочій зоні
коллаборативного робота для отримання команд в рамках концепції
Industry 5.0»

Код програми створення файлу для збереження контрольних точок
обличчя (face_landmarks_sample.py)

```
import cv2
import mediapipe as mp
import numpy as np
import tkinter as tk
from tkinter import messagebox

# Ініціалізація Mediapipe Face Mesh
mp_face_mesh = mp.solutions.face_mesh
face_mesh = mp_face_mesh.FaceMesh(static_image_mode=False,
max_num_faces=1, min_detection_confidence=0.5)

# Ініціалізація вікна Tkinter
window = tk.Tk()
window.title("Збереження контрольних точок обличчя")

# Функція для збереження контрольних точок обличчя у файл
def save_landmarks():
    cap = cv2.VideoCapture(1)
    saved = False

    while not saved:
        ret, frame = cap.read()
        if not ret:
            break

        # Перетворюємо кадр у RGB
        rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        # Виконуємо обробку кадру для виявлення контрольних точок
        results = face_mesh.process(rgb_frame)

        if results.multi_face_landmarks:
            for face_landmarks in results.multi_face_landmarks:
```

```

landmarks = []
for lm in face_landmarks.landmark:
    x = lm.x
    y = lm.y
    z = lm.z
    landmarks.append((x, y, z))

# Збереження контрольних точок у файл
np.save('face_landmarks_sample.npy', landmarks)
saved = True
messagebox.showinfo("Успіх", "Контрольні точки обличчя збережено!")
break

cv2.imshow("Захоплення обличчя", frame)

if cv2.waitKey(5) & 0xFF == 27: # Натисніть ESC для виходу
    break

cap.release()
cv2.destroyAllWindows()

# Створення кнопки для збереження контрольних точок
save_button = tk.Button(window, text="Зберегти", command=save_landmarks)
save_button.pack(pady=20)

# Запуск основного циклу Tkinter
window.mainloop()

```

Код програми розпізнавання людини в робочій зоні колаборативного робота маніпулятора на базі глибокі нейронні мережі MobileNetV2.

```

import cv2
import mediapipe as mp
import numpy as np

# Ініціалізація Mediapipe Face Mesh
mp_face_mesh = mp.solutions.face_mesh
face_mesh = mp_face_mesh.FaceMesh(static_image_mode=False,
max_num_faces=1, min_detection_confidence=0.5)

# Ініціалізація модуля малювання для візуалізації контрольних точок
mp_drawing = mp.solutions.drawing_utils

```

```

# Функція для обчислення евклідової відстані між двома точками (x, y, z)
def euclidean_distance(point1, point2):
    return np.linalg.norm(np.array(point1) - np.array(point2))

# Завантаження зразка контрольних точок обличчя для порівняння
sample_landmarks = np.load('venv/face_landmarks_sample.npy')

# Встановлюємо поріг для розпізнавання
RECOGNITION_THRESHOLD = 0.1

# Функція для перевірки схожості з зразком обличчя
def is_face_matching(landmarks, sample_landmarks, threshold):
    distances = []
    for i in range(len(landmarks)):
        # Переконаємося, що обидва масиви мають однаковий розмір (наприклад,
        (x, y, z))
        if len(landmarks[i]) == len(sample_landmarks[i]):
            distances.append(euclidean_distance(landmarks[i], sample_landmarks[i]))
    mean_distance = np.mean(distances)
    return mean_distance < threshold

# Відеопотік з камери
cap = cv2.VideoCapture(1)

while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    # Перетворюємо кадр у RGB
    rgb_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    # Виконуємо обробку кадру для виявлення контрольних точок
    results = face_mesh.process(rgb_frame)

    if results.multi_face_landmarks:
        for face_landmarks in results.multi_face_landmarks:
            landmarks = []
            for lm in face_landmarks.landmark:
                x = lm.x

```

```
y = lm.y
z = lm.z # Додаємо координату Z для уникнення помилок
landmarks.append((x, y, z))

# Перевірка чи обличчя відповідає зразку
if is_face_matching(landmarks, sample_landmarks,
RECOGNITION_THRESHOLD):
    cv2.putText(frame, "Access Granted", (50, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2, cv2.LINE_AA)
else:
    cv2.putText(frame, "Access Denied", (50, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2, cv2.LINE_AA)

# Візуалізація контрольних точок
mp_drawing.draw_landmarks(frame, face_landmarks,
mp_face_mesh.FACEMESH_TESSELATION,
    landmark_drawing_spec=None,
    connection_drawing_spec=mp_drawing.DrawingSpec(color=(0, 255,
0),
                thickness=1, circle_radius=1))

cv2.imshow('Face Recognition', frame)

if cv2.waitKey(5) & 0xFF == 27:
    break

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

НАУКОВЕ ВИДАННЯ

І.Ш. Невлюдов, В.В. Євсєєв, С.С.Максимова

**INDUSTRY 5.0 ТА КОЛАБОРАТИВНА РОБОТОТЕХНІКА:
ДИНАМІЧНИЙ ОПИС НАВКОЛИШНЬОГО СЕРЕДОВИЩА
РОБОТІВ-МАНІПУЛЯТОРІВ З ВИКОРИСТАННЯМ МОВИ PYTHON**

[Монографія]

*Рекомендовано Науково-технічною радою
Харківського національного університету радіоелектроніки
(протокол № 1 від 30.01.2026 року)*

Комп'ютерна верстка Н.К. Ляшова

[електронне видання]

Підп. до друку 20.02.2026 р Формат 60x84 1/16.
Папір офсетний. Друк ротаційний та цифровий лазерний.

Тираж 300 прим. Зам № 598

Видано на замовлення ХНУРЕ

Ігор Шакирович Невлюдов



Завідувач кафедри комп'ютерно-інтегрованих технологій, автоматизації, робототехніки та безпекової інженерії, доктор технічних наук, професор, Заслужений діяч науки і техніки України, Лауреат Державної премії в галузі науки і техніки України; Лауреат Державної премії України в галузі освіти; член вченої ради, член Президії НМР, голова секції 4 НМР, член Президії НТР, заступник голови секції 3 НТР, член редакційної колегії журналу «Сучасний стан наукових досліджень і технологій в промисловості», член Асоціації випускників ХНУРЕ.

e-mail: igor.nevliudov@nure.ua

ORCID: 0000-0002-9837-2309

Scopus ID: 57216434058

Google Академія:

<https://scholar.google.com.ua/citations?user=FEgWNZcAAAAJ&hl>

Владислав В'ячеславович Євсєєв



Професор кафедри комп'ютерно-інтегрованих технологій, автоматизації, робототехніки та безпекової інженерії, доктор технічних наук, професор, учений секретар спеціалізованої вченої ради Д 64.052.04, Ветеран праці України, член Асоціації випускників ХНУРЕ.

e-mail: vladyslav.yevsieiev@nure.ua

ORCID: 0000-0002-2590-7085

Scopus ID: 57190568855

Google Академія: <https://scholar.google.com.ua/citations?user=RumUrnAAAAAJ>

Світлана Святославівна Максимова



Доцент кафедри комп'ютерно-інтегрованих технологій, автоматизації, робототехніки та безпекової інженерії, кандидат технічних наук, доцент.

e-mail: svitlana.milyutina@nure.ua

ORCID: 0000-0002-1375-9337

Scopus ID: 57199329065

Google Академія: <https://scholar.google.com.ua/citations?user=riTSK-UAAAAJ&hl=uk&authuser=1>

Монографія "Industry 5.0 та колаборативна робототехніка: динамічний опис навколишнього середовища роботів-маніпуляторів з використанням мови Python" розглядає сучасні тенденції впровадження колаборативних роботів у виробничі процеси Industry 5.0, фокусуючись на їхній взаємодії з людиною, адаптації до динамічних умов середовища та використанні передових технологій програмування для їх контролю.

У вступі окреслюються основні проблеми й завдання, пов'язані з переходом до Industry 5.0, а також роль колаборативних роботів як ключового елементу нової виробничої парадигми. Перший розділ досліджує концепцію колаборативних роботів-маніпуляторів в Industry 5.0, висвітлюючи відмінності від Industry 4.0, нову екосистемну модель та їхні завдання у промислових умовах.

Другий розділ фокусується на динамічному описі навколишнього середовища колаборативного робота, зокрема на методах і алгоритмах, які дозволяють роботу адаптуватися до змін у середовищі в реальному часі. Важливу роль відіграє комп'ютерний зір і розпізнавання об'єктів за допомогою адаптивних алгоритмів.

Третій розділ присвячений використанню мови програмування Python для розробки робототехнічних систем. У ньому розглядаються основні бібліотеки для машинного навчання та комп'ютерного зору, такі як OpenCV, TensorFlow і MediaPipe, а також середовище розробки PyCharm.

Четвертий розділ демонструє практичні приклади реалізації програм для колаборативних роботів, зокрема використання нейронних мереж для ідентифікації об'єктів у робочій зоні робота, реалізація алгоритмів трекінгу, а також методів прогнозування траєкторії руху людини за допомогою LSTM і фільтра Калмана.

П'ятий розділ аналізує перспективи розвитку колаборативної робототехніки в умовах Industry 5.0, описуючи ключові виклики та тенденції, а також потенціал розширення використання таких роботів у різних галузях промисловості.

Монографія може бути корисна науковцям, аспірантам, здобувачам, які займаються питаннями розробки інтелектуальних робототехнічних систем та фахівцям в галузі знань 17 - Електроніка, автоматизація та електронні комунікації за спеціальністю 174 - Автоматизація, комп'ютерно-інтегровані технології та робототехніка та освітньо-професійних програм: «Робототехніка та кіберфізичні системи», «Автоматизація та комп'ютерно-інтегровані технології» та G7 Автоматизація, комп'ютерно-інтегровані технології та робототехніка.