

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютерних технологій
(повна назва)

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

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)
Розробка комп'ютерного зору для автоматизованої системи управління
промисловими роботами
(тема)

Виконав:
здобувач 3 року навчання,
(скорочений строк навчання)
групи АКТАКІТу -22-1
Дмитро ФІРСОВ
(власне ім'я, прізвище)

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

Тип програми освітньо-професійна
Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(повна назва освітньої програми)

Керівник Данило БЛИЗНЮК
(посада, власне ім'я, прізвище)

Допускається до захисту
Завідувач кафедри КІТАР

Ігор НЕВЛЮДОВ
(власне ім'я, прізвище)

2025 р.

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

A square image showing a handwritten signature in blue ink. The signature is stylized and appears to be 'ДФ' followed by a flourish.

«09» липня 2025 р

Дмитро ФІРСОВ

Харківський національний університет радіоелектроніки

Факультет _____ АКТ _____
Кафедра _____ КІТАР _____
Рівень вищої освіти _____ перший (бакалаврський) _____
Спеціальність _____ 151 Автоматизація та комп'ютерно-інтегровані технології _____
Тип програми _____ Освітньо-професійна _____
Освітня програма _____ Автоматизація та комп'ютерно-інтегровані технології _____
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«19» травня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачу _____ Фірсову Дмитру Віталійовичу _____
(прізвище, ім'я, по батькові)

1 Тема роботи _____ Розробка комп'ютерного зору для автоматизованої системи
управління промисловими роботами _____

Затверджена наказом по університету від _____ 21.05.2025 р. №405 Ст _____

2 Термін подання студентом роботи до екзаменаційної комісії _____ 09.07.2025 _____

3 Вихідні дані до роботи _____

3.1 Мова програмування Python _____

3.2 Модель комп'ютерного зору YOLOv8 _____

3.3 Бібліотека NumPy _____

3.4 Бібліотека Ultralytics YOLOv8 _____

4 Перелік питань, що потрібно опрацювати в роботі _____

4.1 Вступ _____

4.2 Аналіз технічного завдання _____

4.3 Проектування системи комп'ютерного зору _____

4.4 Розробка програмного забезпечення для розпізнавання та фіксації об'єктів _____

4.5 Висновки та перелік джерел посилань _____

5 Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

Демонстраційний матеріал, представлений у форматі презентації

PowerPoint (*.ppt) 12 с. формату А4

6 Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз технічного завдання	21.11.2024	Виконано
2	Проектування системи комп'ютерного зору	07.12.2024	Виконано
3	Визначення вимог до роботи автоматизованої системи	14.03.2025	Виконано
4	Розробка програмного забезпечення для розпізнавання та фіксації об'єктів	20.04.2025	Виконано
5	Висновки та перелік джерел посилань.	19.06.2025	Виконано

Дата видачі завдання 19.05.2025

Здобувач

(підпис)

Дмитро ФІРСОВ

(власне ім'я, прізвище)

Керівник роботи

(підпис)

Данило БЛИЗНЮК

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 64 с., 14 рис., 2 дод., 12 джерел.

ШТУЧНИЙ ІНТЕЛЕКТ, PYTHON, YOLO, КОМП'ЮТЕРНИЙ ЗІР, РОЗПІЗНАВАННЯ ОБ'ЄКТІВ, SSIM

Мета роботи – розроблення програмного забезпечення для розпізнавання об'єктів у робочій зоні промислового робота за допомогою алгоритмів комп'ютерного зору, зокрема виявлення еталонних об'єктів та сторонніх предметів.

Об'єкт розробки – процеси візуального моніторингу робочої зони промислових роботизованих комплексів.

Предмет розробки – автоматизована система розпізнавання об'єктів візуальних даних з використанням методів комп'ютерного зору.

У даній роботі було проаналізовано предметну область – автоматизацію контролю робочого середовища промислових роботів за допомогою систем технічного зору. Розглянуто сучасні методи розпізнавання об'єктів, проаналізовано актуальні підходи до впровадження систем розпізнавання у промислових умовах, визначено їх переваги та недоліки. У процесі реалізації було зібрано навчальний датасет із зображеннями еталонних об'єктів та сторонніх предметів. Для кращої точності додатково було використано метрику для визначення відхилень від еталонного зображення.

Було розроблено програмне забезпечення на мові Python, яке виконує завдання розпізнавання, аналізу та відображення результатів у зручному вигляді. Вся система була протестована на реальних прикладах, що дозволило підтвердити її ефективність у виявленні сторонніх об'єктів і контролі справності обладнання.

ABSTRACT

The explanatory note: 64 p., 14 fig., 2 app, 12 sources.

ARTIFICIAL INTELLIGENCE, PYTHON, YOLO, COMPUTER VISION, OBJECT RECOGNITION, SSIM

The purpose of the study object recognition in the workspace of an industrial robot using computer vision algorithms, specifically for detecting reference objects (e.g., a computer mouse) and foreign items.

Object of development visual monitoring processes within the workspace of industrial robotic systems.

The subject of development is an object recognition system for visual data using computer vision methods.

The bachelor's thesis work presents the automation of monitoring the industrial robot's working environment using technical vision systems. Modern object recognition methods were reviewed, along with current approaches to implementing recognition systems in industrial environments. Their advantages and disadvantages were identified.

To design the custom system, the object of development was described, and its main components were defined.

During implementation, a custom training dataset containing images of both reference and foreign objects was created using the Roboflow platform. The model was then trained in the Google Colab environment using the Ultralytics library. For improved accuracy, the SSIM metric was additionally used to detect deviations from the reference image.

The developed software, written in Python, performs recognition, analysis, and displays results in a user-friendly format. The entire system was tested on real-world examples, which confirmed its effectiveness in detecting foreign objects and monitoring equipment integrity.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ДСТУ – державний стандарт України;

ЦСР – цілі сталого розвитку;

ПК – персональний комп’ютер;

API – application programming interface;

CPU – central processing unit;

Flask – веб-фреймворк для Python;

NumPy – бібліотека для роботи з масивами і математичних операцій у Python;

OpenCV – open source computer vision library;

Python – мова програмування;

Roboflow – платформа для підготовки і управління датасетами комп’ютерного зору;

YOLO – you only look once.

ЗМІСТ

Перелік умовних скорочень	7
Вступ.....	9
1 Аналіз технічного завдання.....	12
1.1 Поняття та функції комп'ютерного зору	12
1.2 Аналіз задач комп'ютерного зору у виробничих процесах	14
1.3 Огляд використання комп'ютерного зору в роботизованих системах.....	15
1.4 Визначення проблем та викликів при застосуванні зору у виробництві ..	16
1.5 Аналіз існуючих рішень для розробки.....	18
2.Проектування системи комп'ютерного зору	20
2.1 Розробка загальної схема автоматизованої схеми	20
2.2 Визначення вимог до роботи автоматизованої системи	20
2.3 Розпізнавання об'єктів за допомогою YOLO.....	25
2.4 Обґрунтування вибору компонентів реалізації	27
3 Розробка програмного забезпечення для розпізнавання та фіксації об'єктів ..	30
3.1 Реалізація розпізнавання об'єктів з використанням YOLO.....	30
3.2 Програмна реалізація модуля розпізнавання	34
3.3 Охорона праці	46
3.4 Теорія автоматичного управління в автоматизованій системі комп'ютерного зору	46
Висновки	51
Перелік джерел посилання	53
Додаток А Листінг програми Python.....	56
Додаток Б Демонстраційний матеріал.....	63

ВСТУП

У сучасному промисловому виробництві автоматизація процесів є важливою частиною підвищення ефективності, зниження витрат та покращення якості продукції. Впровадження новітніх технологій дозволяє забезпечити високу продуктивність, стабільність та надійність виробничих систем.

Однією з перспективних та інноваційних технологій у цій сфері є комп'ютерний зір. Це технологія, що дозволяє автоматизованим пристроям отримувати, аналізувати та інтерпретувати візуальну інформацію з навколишнього середовища. Такі можливості відкривають широкі перспективи для модернізації автоматизованих систем управління, що особливо актуально для галузей із високими вимогами до точності, швидкості та якості виконання операцій.

Комп'ютерний зір вже активно застосовується в промисловості для вирішення таких завдань, як контроль якості продукції, орієнтація об'єктів на конвеєрі, автоматичне виявлення дефектів, позиціонування та розпізнавання форми й кольору. Завдяки поєднанню комп'ютерного зору з нейронними мережами стає можливим створення інтелектуальних рішень, що адаптуються до змін середовища та приймають оптимальні рішення у реальному часі. Це значно підвищує ефективність функціонування роботизованих систем, дозволяючи їм працювати з більшою автономністю та точністю.

У сучасних умовах розвитку промисловості автоматизація виробничих процесів є ключовим фактором підвищення ефективності, якості продукції та конкурентоспроможності підприємств. Впровадження комп'ютерного зору в системи управління промисловими роботами відкриває нові можливості для оптимізації операцій, зменшення людського фактора та підвищення точності виконання завдань. Комп'ютерний зір дозволяє автоматизувати контроль якості, своєчасно виявляти дефекти, адаптувати роботу системи до змін навколишнього середовища, що є особливо актуальним у складних виробничих умовах.

Інтеграція інтелектуальних алгоритмів, зокрема нейронних мереж, підвищує рівень автономності роботизованих комплексів, забезпечує швидку реакцію на зміни та зменшує час простою обладнання

Мета роботи – розроблення програмного забезпечення для розпізнавання об'єктів у робочій зоні промислового робота за допомогою алгоритмів комп'ютерного зору, зокрема виявлення еталонних об'єктів та сторонніх предметів.

Об'єкт розробки – процеси візуального моніторингу робочої зони промислових роботизованих комплексів.

Предмет розробки – автоматизована система розпізнавання об'єктів візуальних даних з використанням методів комп'ютерного зору.

Автоматизована система управління повинна забезпечити можливість виявлення та класифікації об'єктів, визначення їх просторових координат та передачу цієї інформації до виконавчих механізмів робота.

Для досягнення мети було визначено наступні завдання:

- провести Аналіз задач комп'ютерного зору у виробничих процесах;
- розробити та впровадити алгоритми розпізнавання об'єктів із застосуванням нейронних мереж;
- сформуванати та анотувати датасет зображень для навчання та тестування моделей комп'ютерного зору;
- провести тестування та валідацію розробленої системи на реальних виробничих даних;
- оформити кваліфікаційну роботу згідно ДСТУ 3008:2015 [1], а також з методичними вказівками з підготовки й оформлення кваліфікаційної роботи здобувачами першого (бакалаврського) рівня вищої освіти спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» [2] з дипломним проєктуванням для студентів усіх форм навчання [3].

Отримані результати роботи можна віднести до Цілі сталого розвитку 9 “Промисловість, інновації та інфраструктура”, а саме п. 9.4 “Сприяти

прискореному розвитку високо- та середньо-високотехнологічних секторів переробної промисловості, які формуються на основі використання ланцюгів «освіта – наука – виробництво» та кластерного підходу за напрямками розвиток інноваційної екосистеми”, індикатор 9.4.1.

1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Поняття та функції комп'ютерного зору

Комп'ютерний зір, як галузь на стику прикладної математики, електроніки та інформаційних технологій, сформувався у відповідь на зростаючу потребу технічних систем «бачити» та інтерпретувати інформацію з навколишнього середовища. Його сутність полягає у здатності автоматизованих пристроїв аналізувати візуальні дані для прийняття керуючих рішень, орієнтації в просторі або виконання певних дій у виробничому процесі. На відміну від традиційних систем керування, які покладаються винятково на числові сигнали та логіку, комп'ютерний зір забезпечує сприйняття просторових та візуальних властивостей об'єктів, що відкриває можливість для гнучкіших і контекстно-залежних рішень.

У сучасному промисловому виробництві одним із ключових напрямів розвитку є автоматизація та використання роботизованих систем, які здатні виконувати широкий спектр операцій без участі людини. Проте для ефективної роботи таких систем необхідно забезпечити їх здатністю аналізувати навколишнє середовище та взаємодіяти з ним у режимі реального часу.

Поняття комп'ютерного зору передбачає обробку зображень у цифровому вигляді, перетворення аналогових сигналів, отриманих із камер або сенсорів, у структуровану інформацію, придатну для подальшого аналізу або інтеграції в систему керування. Цей процес включає кілька етапів: від первинної фільтрації зображення до інтерпретації просторових зв'язків між елементами сцени. Відповідно, комп'ютерний зір виступає інструментом, здатним зменшити вплив людського чинника, забезпечити безперервність візуального контролю та підвищити якість виробничого процесу завдяки точному виявленню відхилень або помилок у режимі реального часу (рис. 1.1).

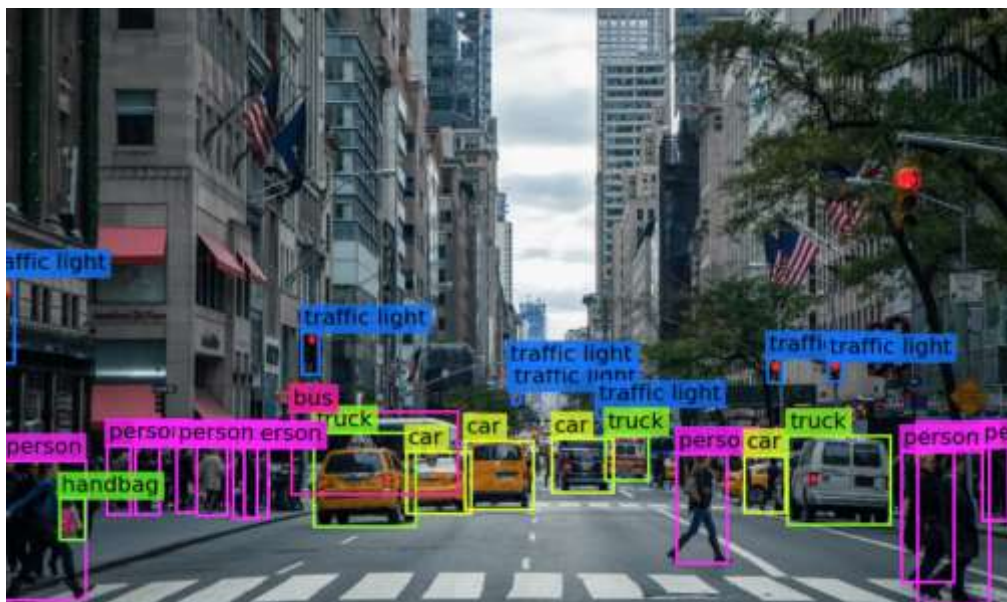


Рисунок 1.1 – Приклад комп'ютерного зору у повсякденності

У контексті промислових роботизованих систем комп'ютерний зір не лише доповнює сенсорні можливості технічного об'єкта, але й виконує функцію адаптації. Тобто він дозволяє машині реагувати на зміни у розташуванні деталей, виявляти дефекти або орієнтуватися у змінному середовищі без необхідності жорсткого перепрограмування. Така інтеграція забезпечує новий рівень гнучкості автоматизованого виробництва, де дії роботів залежать не тільки від заздалегідь визначених координат, а й від динамічного зчитування реальної ситуації.

З цього логічно випливає, що комп'ютерний зір у сучасному розумінні перетворився з експериментального елемента в інженерно повноцінний компонент системи керування. Його застосування охоплює як низькорівневі задачі. До цього можна віднести, наприклад, позиціонування або визначення контурів, так і складніші сценарії, включаючи прогнозування, класифікацію та візуальну навігацію. У промисловості ці можливості активно використовуються для підвищення ефективності, автоматичного реагування на нестандартні ситуації, що робить комп'ютерний зір важливою ланкою у побудові інтелектуальних виробничих систем.

Комп'ютерний зір є не просто технічним засобом, а концептуальною платформою, яка дозволяє інтегрувати сприйняття середовища у логіку керування. Його функціональність виходить за межі простої інтерпретації зображень. Він формує підґрунтя для прийняття рішень у динамічних і змінних умовах промислового виробництва.

1.2 Аналіз задач комп'ютерного зору у виробничих процесах

Впровадження комп'ютерного зору у виробничі процеси є стратегічним кроком до побудови більш автономного, адаптивного й технологічно гнучкого виробництва. Ця технологія виконує не лише допоміжну роль у контролі якості чи автоматизації рутинних операцій, а й стає критичним компонентом для забезпечення стійкості виробничих ланцюгів в умовах динамічних змін навколишнього середовища або специфікацій продукції.

На відміну від традиційних датчиків, що фіксують окремі параметри, комп'ютерний зір дозволяє виявляти просторові закономірності, зміни текстури, форми, кольору чи положення об'єкта. Це відкриває нові підходи до вирішення низки практичних задач, зокрема в умовах, де жорстко задані сценарії керування втрачають ефективність.

Комп'ютерний зір, інтегрований у систему управління, фактично виконує функції локального аналізу ситуації та передає високорівневу інформацію. Він не лише про наявність об'єкта, але й про його стан, орієнтацію чи відповідність заданим критеріям.

У виробничому середовищі це дозволяє роботизованим пристроям працювати заздалегідь запрограмовано, а також з урахуванням поточного розташування компонентів, їхньої взаємодії або змінних умов виконання операцій. Завдяки цьому можливим стає автоматичне коригування руху маніпулятора, виявлення відхилень від норми, а також забезпечення взаємодії з різними типами продукції без зміни апаратної конфігурації.

Крім того, комп'ютерний зір сприяє зменшенню кількості зупинок у роботі обладнання шляхом раннього виявлення аномалій, недопущення помилкових дій і зниження навантаження на оператора. Це, своєю чергою, забезпечує стабільність технологічного процесу та підвищує рентабельність виробництва за рахунок зменшення втрат і простоїв.

Тому задачі комп'ютерного зору у виробництві охоплюють не лише технічну підтримку певних операцій, а й побудову більш розумного рівня управління, що базується на аналізі візуальної інформації. Це дозволяє системі не просто виконувати послідовність дій, а взаємодіяти з навколишнім середовищем на основі актуальних даних, формуючи передумови для створення по-справжньому адаптивного та самокерованого виробництва.

1.3 Огляд використання комп'ютерного зору в роботизованих системах

Сучасна промисловість дедалі частіше звертається до рішень, які забезпечують гнучкість, точність і здатність до адаптації в режимі реального часу. Комп'ютерний зір, будучи ключовою технологією в складі роботизованих систем, демонструє високу ефективність у широкому спектрі завдань. А саме від базового позиціонування до складної контекстуальної інтерпретації ситуацій у виробничому середовищі.

У низці практичних кейсів комп'ютерний зір довів свою доцільність у задач точного захоплення деталей. У таких ситуаціях традиційні координатні системи є недостатніми через похибки в розташуванні об'єктів або їх змінну орієнтацію. Інтеграція візуального аналізу дозволяє роботу оперативно адаптуватися до реального положення деталі, що знижує потребу у жорсткому калібруванні та підвищує гнучкість у роботі з неоднорідною продукцією.

Ще одним поширеним прикладом є системи контролю якості, де комп'ютерний зір забезпечує виявлення дефектів поверхні, невідповідностей геометричних параметрів чи порушень в зібраній конструкції. На відміну від

ручного контролю, такий підхід забезпечує стабільність критеріїв оцінювання, масштабованість та можливість безперервного контролю у потоці виробництва.

Також значного поширення набули роботизовані системи пакування й сортування, де візуальні алгоритми використовуються для класифікації об'єктів за формою, розміром чи кольором. У подібних задачах комп'ютерний зір дозволяє досягати високої точності навіть при змінному темпі подачі або варіативності вхідного матеріалу, що є типовим для харчової, фармацевтичної та легкої промисловості.

Значний прогрес спостерігається також у галузі колаборативної робототехніки, де візуальне розпізнавання жестів, позицій руки людини або наявності перешкод забезпечує безпечну взаємодію між людиною й машиною. В таких умовах комп'ютерний зір не лише виконує технічну роль, а й слугує інтерфейсом, що формує природну і зрозумілу логіку співпраці.

Промислові приклади впровадження комп'ютерного зору свідчать про його здатність якісно змінювати структуру виробничих процесів. Він не лише автоматизує окремі операції, а й дозволяє реалізовувати моделі управління, що враховують контекст, змінність і потребу в самостійній оцінці ситуацій. Саме тому застосування комп'ютерного зору розглядається не як допоміжна опція, а як невід'ємна складова сучасних інтелектуальних роботизованих систем.

1.4 Визначення проблем та викликів при застосуванні зору у виробництві

Попри значні досягнення у сфері комп'ютерного зору, його повноцінне впровадження у виробничі процеси залишається складним і багатоаспектним завданням. Причиною цього є не лише технічна складність візуального аналізу, а й обмеження, які накладаються реальними умовами промислового середовища. Виробництво, як правило, характеризується високим рівнем динаміки, мінливістю об'єктів контролю, впливом зовнішніх чинників, а також жорсткими вимогами до швидкості та надійності.

Однією з основних проблем є нестабільність умов освітлення, що істотно впливає на якість зображень, які аналізує система. Навіть незначні зміни в яскравості або тіні можуть призводити до хибних розпізнавань або втрати точності при сегментації об'єктів. У традиційних промислових умовах не завжди можливо забезпечити ідеальне або контрольоване освітлення, що створює необхідність у застосуванні додаткових фільтрів, адаптивних алгоритмів чи складного апаратного забезпечення.

Іншим серйозним викликом є варіативність об'єктів, з якими працює система. Навіть у межах однієї серії продукції можуть спостерігатися зміни форми, розміру чи текстури, які потребують або глибокого навчання моделей, або постійного переналаштування параметрів розпізнавання. Це ускладнює побудову універсальних рішень, здатних працювати ефективно без втручання оператора або технічного персоналу.

Особливу складність становить інтеграція комп'ютерного зору з існуючими системами керування, зокрема в умовах, коли виробниче обладнання не підтримує відкриті протоколи або не дозволяє гнучкої взаємодії з новими модулями. У таких випадках виникає потреба в проміжних програмних компонентах або навіть у фізичній модернізації обладнання, що підвищує вартість та тривалість впровадження.

Не менш важливою є проблема обчислювальних ресурсів. Сучасні алгоритми комп'ютерного зору, особливо ті, що базуються на глибокому навчанні, вимагають значних обчислювальних потужностей і високої пропускної здатності систем зберігання та обміну даними. Це стає критичним фактором у випадках, коли обробка має здійснюватися в режимі реального часу або на рівні вбудованих пристроїв з обмеженими ресурсами.

Окремої уваги потребує й питання надійності та стійкості до помилок. Виробництво не терпить неточностей, і навіть поодинокі некоректні рішення системи комп'ютерного зору може призвести до виходу з ладу устаткування, зіпсованої продукції або зупинки лінії. Тому важливим аспектом стає не лише

точність алгоритмів, але й здатність системи до самодіагностики, виявлення невизначеностей і безпечного переходу в резервні режими роботи.

Використання комп'ютерного зору у виробництві супроводжується низкою технологічних, інженерних і організаційних бар'єрів. Подолання цих викликів вимагає комплексного підходу: удосконалення алгоритмів і технічного забезпечення до переосмислення архітектури систем керування, орієнтованих на гнучку взаємодію зі змінним середовищем та інтелектуальними модулями.

1.5 Аналіз існуючих рішень для розробки

Ефективне впровадження комп'ютерного зору в автоматизовані системи керування роботизованими установками неможливе без попереднього аналізу існуючих інструментів та технологій, що забезпечують розробку, тестування і впровадження відповідних рішень. На сучасному етапі ринок програмно-апаратних засобів пропонує широкий спектр платформ і бібліотек, кожна з яких має свої переваги, обмеження та сфери доцільного застосування.

У сегменті програмного забезпечення чільне місце посідають фреймворки з відкритим кодом, серед яких особливо виділяються OpenCV, TensorFlow, PyTorch, а також спеціалізовані рішення на базі бібліотек глибинного навчання. OpenCV забезпечує достатній рівень функціональності для базових та середньої складності задач обробки зображень і часто використовується як стартовий інструмент при створенні прототипів. Його перевагою є широка документація, міжплатформеність та активна спільнота розробників. Водночас, для задач, що потребують глибокого розуміння контексту сцени або високого рівня узагальнення, більш ефективними виявляються нейромережеві підходи, реалізовані через TensorFlow або PyTorch. Ці інструменти дозволяють створювати моделі, здатні навчатися на великих об'ємах даних і адаптуватися до змінних умов середовища.

З апаратної точки зору важливу роль відіграє вибір платформи для обчислень. Традиційні ПК-станції можуть забезпечувати високу продуктивність,

але часто є надлишковими для локалізованих систем. У таких випадках доцільним є використання вбудованих рішень, таких як NVIDIA Jetson, Raspberry Pi з модулями камер або спеціалізовані системи на базі FPGA. Вибір апаратного забезпечення визначається не лише технічними характеристиками, а й вимогами до енергоспоживання, тепловиділення, доступності периферійних інтерфейсів та здатності до інтеграції у виробниче середовище.

Важливим критерієм при виборі інструментів також виступає можливість інтеграції з системами керування, зокрема через промислові протоколи або середовища типу ROS (Robot Operating System), що забезпечують уніфіковану архітектуру взаємодії між сенсорами, виконавчими механізмами та модулями прийняття рішень.

Не менш суттєвим є й аспект доступності середовищ моделювання та тестування, які дають змогу проводити віртуальну валідацію рішень до їх фізичного впровадження. Інструменти на кшталт RoboDK, Gazebo або MATLAB Simulink сприяють розробці і перевірці логіки керування без ризику для обладнання або простою виробничої лінії.

Узагальнюючи, можна відзначити, що ефективне рішення для розробки систем комп'ютерного зору не обмежується вибором однієї технології. Успішна реалізація потребує поєднання кількох інструментів з урахуванням особливостей задачі, вимог до продуктивності, умов роботи та рівня інтеграції із суміжними компонентами. Такий підхід дозволяє створити систему, що є не лише технічно ефективною, а й адаптованою до специфіки конкретного виробничого середовища.

2 ПРОЄКТУВАННЯ СИСТЕМИ КОМП'ЮТЕРНОГО ЗОРУ

2.1 Розробка загальної схеми автоматизованої схеми

Загальна структура системи комп'ютерного зору, розробленої для автоматизованого управління промисловими роботами, побудована на принципах модульності, адаптивності та інтеграції з наявними елементами виробничої інфраструктури. Вона поєднує засоби технічного спостереження (відеокамери), алгоритми обробки зображень (модуль комп'ютерного зору), механізми аналізу (класифікація, порівняння з еталоном) і засоби передавання інформації до системи управління (рис. 2.1).



Рисунок 2.1 – Схема автоматизованої системи

2.2 Визначення вимог до роботи автоматизованої системи

Проектування системи комп'ютерного зору в контексті управління промисловими роботами вимагає чіткого формулювання як функціональних, так

і технічних вимог. Відповідність цим вимогам забезпечує працездатність модуля та його інтеграцію у виробничий цикл без шкоди для ефективності, безпеки чи стабільності технологічного процесу.

З функціональної точки зору, система повинна виконувати точне виявлення об'єктів у реальному часі та забезпечувати передачу координат або інших параметрів розпізнаних елементів до блоку керування роботом. Ключовим завданням виступає здатність ідентифікувати положення та орієнтацію об'єктів у межах зони дії маніпулятора. При цьому система повинна враховувати можливу змінність положення виробів, їхню часткову перекритість або варіації форми та розміру. Крім того, необхідною є наявність алгоритмів перевірки якості вхідних зображень, а також базової фільтрації шумів і фонового зашумлення, що часто присутні у виробничому середовищі.

Функціональність модуля повинна включати можливість самодіагностики та контролю помилок, а також реагування на ситуації невизначеності або втрати зображення. Важливо, щоб система могла працювати як у режимі потоку, так і з окремими кадрами, забезпечуючи гнучкість при зміні технологічної задачі або параметрів робота.

У технічному аспекті система комп'ютерного зору повинна відповідати низці критичних критеріїв. По-перше, її продуктивність має забезпечувати обробку візуальних даних із мінімальною затримкою, що не перевищує допустимих значень для даного класу виробництва. Це передбачає вибір ефективного алгоритмічного ядра та обчислювальної платформи, здатної підтримувати необхідну частоту кадрів і швидкість прийняття рішень.

По-друге, камера або інші сенсори мають мати достатню роздільну здатність та частоту оновлення, щоб гарантувати виявлення дрібних об'єктів або дефектів з урахуванням необхідної точності позиціонування. Важливо також забезпечити сумісність із системами керування промислового робота, що може потребувати реалізації підтримки промислових протоколів або спеціалізованих API для двостороннього обміну даними.

Крім того, система повинна відповідати вимогам до експлуатаційної надійності: стабільність роботи у складних умовах (пил, вібрації, змінне освітлення), низький рівень відмов, здатність до автономного відновлення після збоїв. Пристрої, що входять до її складу, повинні мати достатній ресурс роботи та бути адаптованими до промислових норм (захист за IP-рейтингом, електромагнітна сумісність тощо). Таким чином, сукупність функціональних і технічних вимог формує основу для проєктування системи комп'ютерного зору, яка повинна не лише розв'язувати поставлену задачу, а й гармонійно взаємодіяти з іншими елементами виробничого комплексу, забезпечуючи надійність, точність і динамічну адаптивність під час роботи.

Призначення розробки системи комп'ютерного зору для автоматизованої системи управління промисловими роботами полягає у створенні інтелектуальної підсистеми, яка забезпечує розпізнавання об'єктів, їх аналіз, виявлення дефектів або сторонніх елементів, а також передавання відповідної інформації до керувального модуля в режимі реального часу. Головною метою є підвищення продуктивності, точності, безпеки та автономності роботи роботизованих систем у промисловому середовищі (рис. 2.2).

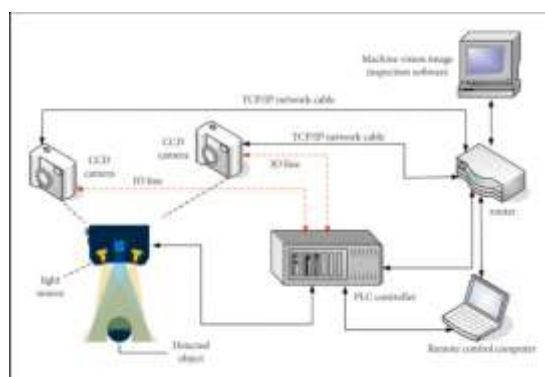


Рисунок 2.2 – Схематичне зображення системи комп'ютерного зору в промисловому процесі

Для забезпечення ефективного функціонування автоматизованої системи керування промисловими роботами з вбудованим комп'ютерним зором необхідно визначити низку вимог, які стосуються як функціональності

програмного забезпечення, так і технічних характеристик апаратної складової. Ці вимоги формуються з урахуванням особливостей виробничого середовища, необхідного рівня точності, швидкодії, масштабованості та інтеграційних можливостей.

Функціональна модель системи повинна охоплювати декілька ключових аспектів. Насамперед, це здатність до точного та надійного розпізнавання об'єктів. У рамках виробничого процесу це означає ідентифікацію деталей, вузлів, механізмів або готових елементів продукції на основі аналізу відеопотоку або окремих зображень. При цьому система повинна функціонувати в умовах змінного положення об'єктів, різного фону або часткової видимості, що є типовим у реальному виробничому середовищі.

Наступною важливою функцією є виявлення дефектів та сторонніх предметів. Завдяки використанню методів аналізу геометричних і текстурних характеристик об'єкта, система має виявляти відхилення від встановлених нормативних параметрів. Це дозволяє оперативно і точно ідентифікувати браковану продукцію та запобігти її подальшому потраплянню на наступні етапи виробництва.

Важливим компонентом архітектури є механізм взаємодії з системою управління роботизованим маніпулятором. Результати аналізу візуальної інформації мають передаватися до контролера або керувального програмного середовища з метою адаптації дій робота відповідно до поточної ситуації – наприклад, змін у положенні деталі або виявлення дефекту. Такий тип інтеграції передбачає підтримку протоколів обміну даними, сумісних з існуючими ПЛК, HMI або SCADA-рішеннями.

Додатково, у системі передбачено механізми віддаленого моніторингу та контролю. Це включає можливість доступу до результатів обробки в реальному часі через внутрішню локальну мережу або інтернет, що значно спрощує нагляд за технічним станом системи та зменшує потребу в постійній фізичній присутності оператора.

З технічної точки зору, система повинна забезпечувати високу швидкодію, що означає здатність до обробки відеопотоку або зображень у реальному часі з мінімальними затримками. Це критично важливо для процесів, де робот виконує дії у динаміці або з високою частотою повторення операцій. Усі обчислення мають відбуватися з урахуванням апаратних обмежень і допустимого часу реакції системи управління.

Надійність системи розглядається як здатність до стабільної роботи в умовах впливу вібрацій, пилу, вологості та шуму. Тобто факторів, характерних для більшості промислових підприємств. Усі компоненти, включаючи камери, освітлення, обчислювальні модулі та комунікаційні пристрої, повинні відповідати індустріальним стандартам експлуатації.

Масштабованість є ще одним важливим критерієм: архітектура системи повинна дозволяти додавання нових камер, сенсорів або обчислювальних вузлів без потреби у повній реконфігурації всієї системи. Це забезпечує гнучкість під час модернізації виробництва або зміни технологічного процесу.

Окремо варто згадати вимоги до безпеки: система повинна включати механізми захисту даних, обмеження доступу до конфіденційної інформації та контроль за збереженням результатів обробки. Програмне середовище повинно бути інтуїтивно зрозумілим для операторів та інженерного персоналу, а процедури оновлення моделей і алгоритмів максимально спрощеними.

У розробці системи передбачається використання сучасних інструментів, серед яких ключову роль відіграє модель YOLO (You Only Look Once), що забезпечує ефективне розпізнавання об'єктів у реальному часі. Для реалізації алгоритмів обробки та роботи з відео застосовується бібліотека OpenCV, у поєднанні з платформою Ultralytics, яка спрощує процес тренування та впровадження YOLO-моделей. Побудова та навчання нейронних мереж відбувається у середовищі PyTorch, тоді як для супутніх обчислень і візуалізації використовуються NumPy та Matplotlib.

Ключова задача системи – автоматичне виявлення об'єктів у полі зору камери, що охоплює визначення їхнього місцеположення, класифікацію за

типом, а також оцінку відповідності заданим критеріям якості. У контексті даної роботи об'єктом розпізнавання є комп'ютерна миша, яка має бути точно локалізована на зображенні та класифікована як справна або така, що містить візуальні дефекти. Крім того, система повинна виявляти сторонні об'єкти, присутність яких у зоні огляду є небажаною і може свідчити про порушення технологічного процесу.

Важливою складовою є здатність системи не лише фіксувати факт наявності об'єкта, а й виконувати його аналіз на основі візуальних ознак – форми, кольору, структури поверхні. Для цього необхідна інтеграція методів глибинного навчання та алгоритмів порівняння для оцінки схожості із заздалегідь визначеним еталоном. Такий підхід дозволяє не лише виконати класифікацію, але й забезпечити гнучкість до варіацій об'єктів, що природно виникають у реальних виробничих умовах.

Серед другорядних задач, які також входять до функціоналу модуля, слід виділити: автоматичне формування висновку щодо поточного стану виявлених об'єктів; генерацію повідомлень у зручному для оператора вигляді; збереження проаналізованих кадрів із результатами; а також можливість віддаленого моніторингу результатів через вбудовані інтерфейси.

Вимоги до розроблюваної системи формуються не лише на основі технічних параметрів, а й з урахуванням умов експлуатації, вимог до продуктивності, безпеки та зручності подальшого масштабування. Виконання цих умов є запорукою ефективного впровадження комп'ютерного зору в промислове середовище з високими вимогами до надійності та точності.

2.3 Розпізнавання об'єктів за допомогою YOLO

У межах розробки системи комп'ютерного зору для автоматизованого управління промисловими роботами було обрано сучасну архітектуру нейронних мереж YOLO (You Only Look Once). Вона є одною з найбільш продуктивних і перевірених моделей для виконання задач детекції в реальному

часі. В основу розробки покладено останню версію YOLOv8, запропоновану компанією Ultralytics, що поєднує високу швидкість, точність та гнучкість налаштування.

Принципова особливість алгоритму YOLO полягає в тому, що виявлення всіх об'єктів на зображенні виконується за один прохід мережі, без необхідності багатокрокового сканування або поетапного аналізу фрагментів. Такий підхід дозволяє значно зменшити затримки при обробці відеопотоку, що критично важливо для застосування в реальному часі, зокрема, в умовах динамічного виробництва, де робот повинен оперативно реагувати на зміну розташування об'єктів.

Модель YOLOv8 забезпечує можливість одночасного виявлення кількох об'єктів на одному кадрі, що дозволяє ефективно працювати у складних сценаріях, де присутні різноманітні елементи продукції або сторонні предмети. Завдяки підтримці навчання на кастомних датасетах, модель була адаптована до специфіки виробничого середовища: зображення об'єктів, деталей і типових дефектів пройшли процедуру анотування та попередньої обробки, що дозволило навчити мережу розпізнавати як штатні компоненти, так і потенційно небезпечні або неприпустимі елементи.

Для реалізації навчання та подальшої інтеграції було використано фреймворк Ultralytics YOLOv8, який забезпечує інтерфейс командного рядка та API для швидкого запуску процесів тренування, валідації, оцінювання результатів та експорту моделей у придатному для вбудовування форматі. На етапі підготовки було сформовано спеціалізований датасет, який включав репрезентативні зображення об'єктів з різними варіантами розміщення, масштабування та освітлення. Навчання здійснювалося з урахуванням потреб у точному позиціонуванні та класифікації.

Інтеграція навченої моделі у програмний модуль комп'ютерного зору забезпечила обробку відеопотоку в режимі реального часу, аналіз зображень кадр за кадром, передавання координат виявлених об'єктів до керувального модуля системи. За наявності графічного прискорювача (GPU) обчислення

виконувалися в прискореному режимі, що дозволяло досягати високої частоти оновлення без втрати якості розпізнавання.

Застосування YOLOv8 у цьому проєкті має низку практичних переваг. По-перше, це швидка реакція системи, яка дає змогу миттєво виявляти об'єкти та передавати відповідну інформацію до модуля керування. По-друге, висока точність та низький рівень хибнопозитивних спрацьовувань створюють умови для стабільної роботи навіть за складного фону або варіативних умов освітлення. По-третє, гнучкість моделі дозволяє оперативно її перенавчати в разі зміни технологічного процесу або впровадження нових виробничих ліній. Використання YOLO як базової технології розпізнавання об'єктів дозволяє забезпечити точність, швидкість і надійність системи комп'ютерного зору, що є фундаментальними вимогами для її ефективного застосування у виробничих умовах та надійної взаємодії з роботизованими системами управління.

2.4 Обґрунтування вибору компонентів реалізації

Для виявлення дефектів об'єктів у виробничому середовищі було реалізовано підхід, заснований на метриці SSIM (Structural Similarity Index Measure). Цей метод дозволяє обчислювати ступінь структурної схожості між двома зображеннями: еталонним та контрольованим, що надходить із камери у реальному часі. На відміну від простих піксельних операцій, SSIM враховує характеристики, що наближають аналіз до візуального сприйняття людиною. Тобто яскравість, контрастність і локальну структуру зображення.

Алгоритм SSIM оцінює подібність між відповідними ділянками двох зображень, і на основі результату формується коефіцієнт, значення якого коливається в діапазоні від 0 до 1. Значення, близькі до одиниці, свідчать про майже повну відповідність зразку. В межах автоматизованої системи визначено експериментальний поріг схожості 0.6, нижче якого система інтерпретує зображення як потенційно браковане або таке, що містить сторонні або неприпустимі візуальні елементи.

Для реалізації цього методу використано алгоритмічну реалізацію SSIM із бібліотеки `scikit-image` у зв'язці з модулем обробки зображень `OpenCV`. Отримані зображення приводяться до єдиного масштабу та формату, після чого обчислюється коефіцієнт подібності між контрольним та еталонним фреймами. У разі перевищення встановленого порогу зображення вважається коректним, і процес проходить без втручання. Інакше система генерує повідомлення про відхилення, яке передається до модуля контролю та відбракування.

Застосування SSIM забезпечує більш гнучкий та точний аналіз у порівнянні з пороговими фільтрами чи абсолютною піксельною різницею, особливо в умовах змінного освітлення або незначних варіацій об'єктів, які не впливають на якість продукції.

Розробка та реалізація модуля комп'ютерного зору в межах роботи здійснювалась із використанням сучасного інструментарію на базі мови програмування Python. Було застосовано набір бібліотек, що забезпечують повний цикл обробки даних. А саме від зчитування відеопотоку до візуалізації та збереження результатів аналізу.

Велику роль відіграє `OpenCV`, яка відповідає за обробку зображень, захоплення кадрів з камери, базову фільтрацію, зміну розмірів, накладання контурів та побудову масок. Саме ця бібліотека використовується для попередньої обробки перед подачею даних до моделей розпізнавання.

Для реалізації алгоритму детекції об'єктів інтегровано платформу `Ultralytics`, яка надає високорівневий інтерфейс до моделі `YOLOv8`. Вона дозволяє виконувати навчання, валідацію та експлуатацію нейронної мережі без потреби у глибокій модифікації архітектури.

`NumPy` застосовується для роботи з масивами зображень та реалізації математичних операцій, необхідних для нормалізації, обчислення SSIM та внутрішніх процедур підготовки даних. Для виведення зображень, результатів виявлення та налагодження візуальної частини використовується `Matplotlib`.

У межах файлового контролю бібліотеки `os` та `shutil` забезпечують роботу з директоріями, створення логів, збереження кадрів та структуровану

організацію виводу. Це дозволяє не лише протоколювати результати, а й формувати базу даних для подальшого донавчання моделей.

Допоміжні бібліотеки, такі як Pillow (PIL), використовуються для покращеної роботи з форматами зображень високої якості, а roboflow для підготовки, структурування та анотації датасету. Часова мітка результатів фіксується за допомогою вбудованих модулів time і datetime.

Сукупне використання цього інструментарію забезпечує високу продуктивність, масштабованість та простоту підтримки системи, а також дозволяє оперативно адаптувати рішення до зміни умов або виробничих вимог.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗПІЗНАВАННЯ ТА ФІКСАЦІЇ ОБ'ЄКТІВ

3.1 Реалізація розпізнавання об'єктів з використанням YOLO

У межах реалізації системи автоматизованого візуального контролю та управління промисловим роботом було обрано архітектуру нейронної мережі YOLO, яка демонструє високу ефективність у задачах об'єктної детекції в реальному часі. Для розв'язання поставленої задачі застосовано останню версію моделі YOLOv8, що поєднує точність класифікації, високу швидкодію та гнучкість у використанні кастомізованих датасетів.

Принцип роботи YOLO базується на одноетапному аналізі зображення, під час якого виявлення, локалізація та класифікація об'єктів виконуються в межах одного проходу нейронної мережі. Така архітектура забезпечує стабільну роботу системи в умовах змінного середовища виробництва, а також дозволяє обробляти відеопотік з камер у режимі реального часу без критичних затримок.

YOLOv8 використовується для виявлення комп'ютерної миші як цільового об'єкта, а також сторонніх елементів, які не повинні бути присутні на виробничій лінії. Детекція охоплює ідентифікацію класу об'єкта, його локалізацію на зображенні та передачу результатів до наступного блоку системи аналізу. Для задачі автоматичного виявлення комп'ютерної миші та сторонніх об'єктів, що можуть з'являтися на виробничій лінії, було обрано алгоритм YOLOv8. Даний алгоритм виконує одноетапне виявлення, локалізацію та класифікацію об'єктів за один прохід нейронної мережі, що забезпечує низькі затримки при обробці відеопотоку. Це особливо важливо для управління роботизованими системами, де швидка реакція на зміну ситуації є критичною.

Підготовка навчального датасету включала збір зображень комп'ютерної миші у різноманітних ракурсах, варіаціях освітлення, а також зразків із наявними дефектами. Окрім того, до датасету були включені зображення сторонніх

предметів: інструментів, упаковок та інших об'єктів, які необхідно виявляти для запобігання помилок на виробництві. Для анотації зображень застосовувався інструмент LabelImg, що забезпечив формування координат обмежувальних рамок та їх класифікацію у формат, сумісний з процесом тренування YOLO.

Навчання моделі відбувалося у два етапи: спершу на платформі Factory Vision для швидкого прототипування та отримання початкових ваг, а потім глибинне донавчання на ресурсах Google Colab із використанням GPU. Такий підхід дозволив покращити якість моделі, підвищити точність детекції та знизити кількість помилкових спрацьовувань (рис 3.1, рис 3.2).



Рисунок 3.1 – Приклад зображення з мишею

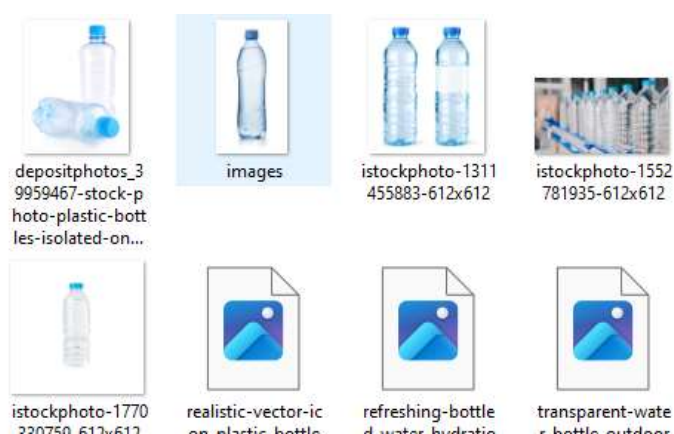


Рисунок 3.2 – Приклад зображення зі стороннім об'єктом

Для перевірки якості виявлених об'єктів реалізовано додатковий етап аналізу за допомогою метрики Structural Similarity Index Measure (SSIM). Вона оцінює структурну схожість між регіоном інтересу, виділеним модулем детекції,

та еталонним зображенням справної миші. SSIM враховує локальні характеристики яскравості, контрастності та структури, що дає змогу ефективно виявляти дефекти або пошкодження, які не завжди можна зафіксувати через прості піксельні відмінності. При значенні SSIM нижче 0.6 об'єкт позначається як дефектний.

Реалізований алгоритм у складі програмного модуля працює за наступною схемою: отримання зображення з камери або файла, детекція об'єктів за допомогою натренованої YOLOv8-моделі, класифікація знайдених об'єктів на «mouse» або «foreign_object», подальший аналіз виявленої миші за SSIM, прийняття рішення про стан об'єкта та вивід результатів із відповідною візуалізацією (рамки, підписи) українською мовою.

Для реалізації та інтеграції модуля використано бібліотеки Python: OpenCV для обробки зображень, Ultralytics для роботи з моделлю YOLOv8, NumPy для математичних операцій, Matplotlib для візуалізації, а також os і shutil для файлової роботи. Додатково застосовувалися Pillow для якісної обробки зображень, roboflow для управління датасетами та стандартні модулі time і datetime для відміток часу (рис. 3.3 – рис. 3.6).



Рисунок 3.3 – Еталонне зображення комп'ютерної миші для навчання та порівняння.



Рисунок 3.4 – Зображення дефектної миші.



Рисунок 3.5 – Приклад стороннього об'єкта, виявленого системою.



Рисунок 3.6 – Вхідне тестове зображення із результатами детекції.

Об'єднання високопродуктивної моделі розпізнавання та методу якісної перевірки об'єктів дозволяє досягти балансу між швидкістю реакції системи та точністю оцінки якості продукції, що робить розроблений модуль придатним для впровадження у промислові робототехнічні комплекси з високими вимогами до надійності та автоматизації.

3.2 Програмна реалізація модуля розпізнавання

У процесі розробки програмного забезпечення для розпізнавання комп'ютерної миші та виявлення сторонніх об'єктів було створено модуль, що базується на інтеграції нейронної мережі YOLOv8 та алгоритмів аналізу зображень. Програмна реалізація є важливим етапом впровадження системи комп'ютерного зору для автоматизованого управління промисловими роботами. Вона передбачає інтеграцію нейронної мережі YOLOv8 для детекції об'єктів, алгоритму SSIM для оцінки якості виявлених елементів, а також організацію обробки, візуалізації та передачі отриманих результатів у систему керування.

Програмна реалізація модуля розпізнавання була виконана мовою програмування Python з використанням бібліотек глибокого навчання та комп'ютерного зору. Основною задачею модуля є виявлення комп'ютерної миші на зображенні, ідентифікація сторонніх об'єктів, а також визначення наявності візуальних дефектів у порівнянні з еталонним зразком. Рішення реалізовано як автономний скрипт, що приймає на вхід зображення, обробляє його та формує проаналізоване зображення з текстовим висновком українською мовою.

Для завантаження і використання моделі застосовано бібліотеку Ultralytics YOLO, що надає зручний інтерфейс для обробки зображень та отримання детекцій об'єктів. Під час розробки було підготовлено набір еталонних зображень комп'ютерної миші, які використовувалися для порівняння та оцінки якості розпізнавання.

Для підвищення стійкості алгоритму до змін освітлення застосовано препроцесінг зображень із використанням CLAHE (Contrast Limited Adaptive Histogram Equalization). Цей підхід дозволяє покращити контрастність візуальної інформації перед детекцією.

Після детекції об'єктів нейронною мережею виконується аналіз регіону інтересу (ROI) миші, який порівнюється з еталонним зразком за допомогою метрики структурної подібності (SSIM). Відхилення у значенні SSIM нижче заданого порогу свідчать про можливу наявність дефекту.

Для візуалізації результатів на зображенні накладаються обмежувальні прямокутники з кольоровою індикацією стану. У модулі реалізована спеціальна функція `compare_mouse_roi`, яка виконує обробку області миші, її попередню фільтрацію за допомогою CLAHE (локального вирівнювання контрастності) та перетворення у відтінки сірого. Після цього порівнюється ROI з еталонним зображенням. Якщо значення SSIM нижче порогового значення (`threshold_ssim = 0.56`), система визначає наявність дефекту.

Результати класифікації та оцінки якості подаються у формі українських повідомлень, які накладаються на зображення із використанням шрифтів Windows (arial.ttf). Вивід статусів оформлено кольоровим кодуванням: (зелений

– справна миша, червоний – дефект, помаранчевий – сторонній об'єкт), а також текстові повідомлення українською мовою, що інформують користувача про результати розпізнавання.

Модуль може обробити окреме зображення або послідовність зображень. Наведені нижче приклади демонструють різні ситуації:

- справна миша без дефектів – текст: "Миша в порядку";
- дефектна миша – "Дефект миші!";
- сторонній об'єкт без миші – "Сторонній об'єкт без миші!";
- миша і сторонній об'єкт – "Миша і сторонній об'єкт виявлені!"

Усі результати зберігаються у вказаній директорії output/ зі збереженням відповідного зображення з позначками.

Код модуля розпізнавання написано у вигляді легко модифікованого скрипту. Потенційні напрями адаптації: підтримка відеопотоку з камери в реальному часі; логування в базу даних (наприклад, SQLite або PostgreSQL) збережених результатів; розгортання у вигляді API за допомогою Flask або FastAPI; підключення до SCADA-систем через OPC UA або Modbus TCP; додавання Telegram-бота для сповіщень про виявлені дефекти у вигляді зображень з підписами.

Для підвищення зручності та ефективності використання розробленого програмного модуля було прийнято рішення створити графічний інтерфейс користувача (GUI). Реалізація GUI дозволила не лише значно спростити взаємодію оператора із системою, а й підвищити загальну інтуїтивність і комфорт роботи, що є особливо важливим у виробничих умовах (рис. 3.7).



Рисунок 3.7 – Головне вікно програми

Графічний інтерфейс створено із застосуванням стандартної бібліотеки Tkinter, яка є вбудованою в мову Python та забезпечує необхідний функціонал для створення віконних додатків із можливістю побудови складних елементів управління. Такий вибір обумовлений його простотою, кросплатформенністю та широкою підтримкою спільнотою розробників.

У центрі інтерфейсу розміщена кнопка, що дозволяє користувачу вибрати зображення для аналізу за допомогою діалогового вікна вибору файлу. Після завантаження обробленого зображення воно виводиться в головному вікні з позначками та текстовим описом результатів розпізнавання. Це дає змогу оператору швидко оцінити стан об'єкта на зображенні, не вдаючись до командного рядка або додаткових програмних засобів (рис. 3.8).



Рисунок 3.8 – Оброблене зображення з виявленими об'єктами

Особливу увагу було приділено розробці верхнього меню програми, яке суттєво покращує організацію роботи користувача. Меню складається з двох основних вкладок – «Файл» та «Допомога». У вкладці «Файл» реалізовано функціонал збереження результату аналізу у вигляді зображення у вибрану користувачем директорію, що є важливим для подальшого зберігання архівів, ведення звітності або передачі результатів іншим системам. Також у цій вкладці розміщена команда виходу з програми, що забезпечує швидке та коректне завершення роботи інтерфейсу (рис. 3.9).

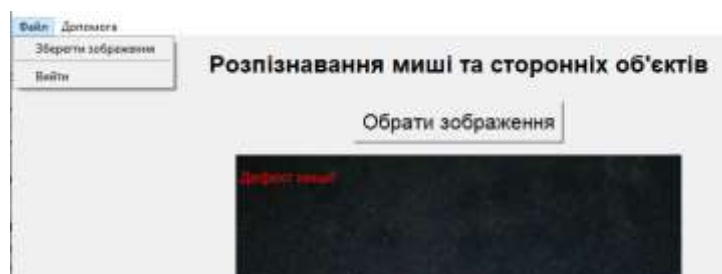


Рисунок 3.9 – Вкладка меню "Файл" із можливістю збереження зображення

Вкладка «Допомога» містить інформацію про призначення програми, її можливості та контактні дані або дані про автора, що сприяє більш швидкому

розумінню суті застосунку користувачем та надає можливість отримати підтримку у разі необхідності (рис. 3.10).

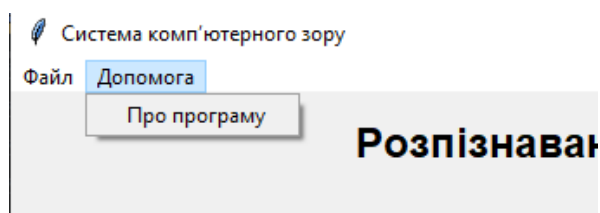


Рисунок 3.10 – Вкладка меню "Допомога" з інформацією про програму

Також у інтерфейс інтегровано можливість виводу текстових повідомлень про поточний статус розпізнавання безпосередньо в головному вікні, що полегшує аналіз та оперативне реагування на результати. Використання кольорового кодування статусів (зелений, червоний, помаранчевий) дозволяє швидко візуально оцінити якість продукції та наявність дефектів або сторонніх об'єктів (рис. 3.11).



Рисунок 3.11 – Виведення статусу розпізнавання у вікні програми

Загальна архітектура GUI розроблена з урахуванням модульності, що дає змогу у майбутньому легко розширювати функціонал – наприклад, додавати обробку відеопотоків у реальному часі, інтегрувати систему повідомлень або впроваджувати інші функції для підвищення автоматизації.

Реалізація меню у поєднанні з кнопками керування та зоною відображення результатів створює зручний інструмент, який забезпечує швидкий доступ до основних функцій системи без необхідності глибоких технічних знань з боку користувача. Це особливо важливо для виробничого середовища, де час на навчання персоналу обмежений, а зручність експлуатації – ключовий фактор.

У процесі розробки GUI було проведено тестування основних сценаріїв роботи, включно з вибором зображення, обробкою, відображенням результатів, збереженням обробленого зображення у файл та викликом допоміжних вікон меню. Завдяки використанню вбудованих діалогів системи, інтерфейс адаптується під операційну систему користувача, що забезпечує стабільність і передбачуваність поведінки додатка.

Код основного файлу наведений нижче:

```
import cv2
import numpy as np
import tkinter as tk
from tkinter import filedialog
from PIL import Image, ImageTk, ImageDraw, ImageFont
from skimage.metrics import structural_similarity as ssim
from ultralytics import YOLO
model = YOLO("yolov8n.pt")
ref_mouse = cv2.imread("images/mouse_etalon.jpg")
threshold_ssim = 0.56
last_image_pil = None
def put_text_ua_cv2(img, text, position, font_path='C:/Windows/Fonts/arial.ttf',
font_size=32, color=(255,255,255)):
```

```
img_pil = Image.fromarray(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
draw = ImageDraw.Draw(img_pil)
font = ImageFont.truetype(font_path, font_size)
draw.text(position, text, font=font, fill=color[::-1])
return cv2.cvtColor(np.array(img_pil), cv2.COLOR_RGB2BGR)

def preprocess(img):
    lab = cv2.cvtColor(img, cv2.COLOR_BGR2LAB)
    l, a, b = cv2.split(lab)
    clahe = cv2.createCLAHE(clipLimit=3.0, tileGridSize=(8, 8))
    cl = clahe.apply(l)
    limg = cv2.merge((cl, a, b))
    return cv2.cvtColor(limg, cv2.COLOR_LAB2BGR)

def compare_mouse_roi(roi_new, roi_ref, threshold=threshold_ssim):
    roi_new = preprocess(roi_new)
    roi_ref = preprocess(roi_ref)
    gray_new = cv2.cvtColor(roi_new, cv2.COLOR_BGR2GRAY)
    gray_ref = cv2.cvtColor(roi_ref, cv2.COLOR_BGR2GRAY)
    gray_ref = cv2.resize(gray_ref, (gray_new.shape[1], gray_new.shape[0]))
    score, _ = ssim(gray_ref, gray_new, full=True)
    print(f"SSIM: {score:.3f}")
    return score < threshold

def analyze_image(image_path):
    img = cv2.imread(image_path)
    results = model(image_path)[0]
    class_names = model.names
    mouse_present = False
    foreign_present = False
    defect = False
    mouse_box = None
    for box in results.bboxes:
```

```
cls_id = int(box.cls)
cls_name = class_names[cls_id]
x1, y1, x2, y2 = box.xyxy[0].cpu().numpy().astype(int)
if cls_name == "mouse":
    mouse_present = True
    mouse_box = (x1, y1, x2, y2)
else:
    foreign_present = True
if mouse_present and mouse_box:
    x1, y1, x2, y2 = mouse_box
    roi_mouse = img[y1:y2, x1:x2]
    defect = compare_mouse_roi(roi_mouse, ref_mouse)
if mouse_present and mouse_box:
    if defect and foreign_present:
        status_text = "Дефект миші! І сторонній об'єкт виявлено!"
        color_status = (0, 0, 255)
    elif defect:
        status_text = "Дефект миші!"
        color_status = (0, 0, 255)
    elif foreign_present:
        status_text = "Миша і сторонній об'єкт виявлені!"
        color_status = (0, 165, 255)
    else:
        status_text = "Миша в порядку"
        color_status = (0, 255, 0)
else:
    if foreign_present:
        status_text = "Сторонній об'єкт без миші!"
        color_status = (0, 0, 255)
    else:
```

```

        status_text = "Мишу не знайдено"
        color_status = (0, 0, 255)
    for box in results.bboxes:
        cls_id = int(box.cls)
        cls_name = class_names[cls_id]
        x1, y1, x2, y2 = box.xyxy[0].cpu().numpy().astype(int)
        if cls_name == "mouse":
            color = (0, 0, 255) if defect else (0, 255, 0)
            label = "mouse (defect)" if defect else "mouse"
        else:
            color = (0, 0, 255)
            label = cls_name
        cv2.rectangle(img, (x1, y1), (x2, y2), color, 2)
        cv2.putText(img, label, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.8, color, 2)
    img = put_text_ua_cv2(img, status_text, (10, 30), font_size=32,
color=color_status)
    return img, status_text
def open_and_process():
    global last_image_pil
    path = filedialog.askopenfilename(filetypes=[("Image Files", "*.jpg;*.png")])
    if not path:
        return
    img, status = analyze_image(path)
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img_pil = Image.fromarray(img_rgb)
    img_pil.thumbnail((800, 600))
    last_image_pil = img_pil.copy()

    imgtk = ImageTk.PhotoImage(img_pil)

```

```

image_label.configure(image=imgtk)
image_label.image = imgtk
result_text.set(status)
def save_result():
    if last_image_pil is None:
        return
    path = filedialog.asksaveasfilename(defaultextension=".jpg",
filetypes=[("JPEG files", "*.jpg")])
    if path:
        last_image_pil.save(path)
        print(f"Зображення збережено у {path}")
root = tk.Tk()
root.title("Система комп'ютерного зору")
root.geometry("900x720")
tk.Label(root, text="Розпізнавання миші та сторонніх об'єктів",
font=("Arial", 18, "bold")).pack(pady=10)
tk.Button(root, text="Обрати зображення", font=("Arial", 16),
command=open_and_process).pack(pady=10)
image_label = tk.Label(root)
image_label.pack()
result_text = tk.StringVar()
tk.Label(root, textvariable=result_text, font=("Arial", 16),
fg="blue").pack(pady=10)
tk.Button(root, text="Зберегти зображення", font=("Arial", 12),
command=save_result).pack(pady=5)
root.mainloop()

```

Програмний модуль забезпечує інтегроване рішення для задачі контролю якості продукції з використанням комп'ютерного зору. Поєднання детекції об'єктів за допомогою нейронної мережі та аналізу якості з використанням

метрики SSIM дозволяє досягти високої точності розпізнавання дефектних елементів та сторонніх об'єктів.

Система виконує низку взаємопов'язаних функцій у такій послідовності. Захоплення зображення за допомогою промислової або веб-камери здійснюється отримання вхідного зображення або відеопотоку з контрольованої зони. Камера встановлюється таким чином, щоб охоплювати область, де відбувається розміщення або переміщення об'єктів (наприклад, транспортна стрічка, позиція захвату тощо). Попередня обробка зображення нормалізується, підвищується контрастність (зокрема за допомогою CLAHE), зменшується вплив шуму. Цей етап готує зображення до ефективного аналізу та мінімізує похибки при розпізнаванні.

Детекція об'єктів нейронна мережа YOLOv8 виконує розпізнавання об'єктів на зображенні, визначає їх координати (обмежувальні рамки) та класифікує за типом. У проєкті передбачено виявлення класу "mouse" та "foreign_object". Аналіз якості об'єкта для об'єктів класу mouse проводиться перевірка на наявність візуальних дефектів за допомогою метрики SSIM. Порівняння здійснюється з еталонним зображенням, яке зберігається в базі шаблонів. Якщо коефіцієнт структурної подібності нижче за встановлений поріг, об'єкт вважається дефектним.

Генерація висновку на основі результатів детекції та порівняння формується текстове повідомлення про поточний стан: «Миша в порядку», «Дефект миші», «Сторонній об'єкт виявлено» тощо. Повідомлення відображається на зображенні та може бути передане у зовнішню систему.

Результати обробки візуалізації та логування зберігаються у вигляді зображення з підписами та рамками, що дає змогу здійснювати подальший аналіз, налагодження або аудит процесу. Можливе збереження в базу даних або файлової систему. Передавання результатів модуль підтримує передачу інформації в систему керування роботом для подальших дій: вилучення дефектного виробу, зупинка лінії, маркування або повідомлення оператора.

3.3 Охорона праці

У процесі розробки системи комп'ютерного зору для автоматизованої системи керування промисловими роботами основним засобом праці є персональний комп'ютер. Характер діяльності передбачає тривале перебування в сидячому положенні, інтенсивну розумову активність, підвищене навантаження на зір, а також відповідальність за точність роботи програмного забезпечення. Усі ці чинники можуть призводити до зорової втоми, болю в спині та шії, психоемоційного виснаження і зниження продуктивності.

Для забезпечення безпечних умов праці необхідно врахувати основні гігієнічні вимоги до організації робочого місця. Зокрема, рівень освітленості повинен становити не менше 300 лк. Якщо площа приміщення становить 12 м² (4 м³), загальний світловий потік можна обчислити за формулою:

$$\Phi = E * S * k \quad (3.1)$$

де E – нормована освітленість (300 лк);

S – площа приміщення (12 м²);

k – коефіцієнт запасу (1,5).

$$\Phi = 300 * 12 * 1,5 = 5400 \text{ лм.} \quad (3.2)$$

Світловий потік у 5400 лм забезпечується п'ятьма світлодіодними лампами по 12 Вт (приблизно 1000 лм – 1100 лм кожна). Таке освітлення дозволяє уникнути перенапруження очей і забезпечити відповідність санітарним нормам.

Окрім освітлення, важливо забезпечити якісний повітрообмін. Відповідно до нормативів, мінімальна кратність повітрообміну для офісного приміщення – 2 рази на годину. Для кімнати площею 12 м² та висотою 2,7 м її об'єм обчислюється за формулою:

$$V = L \times W \times H \quad (3.3)$$

де V – об'єм приміщення;

L, W, H – довжина, ширина і висота відповідно.

$$V = 4 \times 3 \times 2,7 = 32,4 \text{ м}^3 \quad (3.4)$$

Необхідний повітрообмін:

$$Q = n \times V \quad (3.5)$$

де Q – об'єм повітря, що подається на годину;

n – кратність обміну.

$$Q = 2 \times 32,4 = 64,8 \text{ м}^3/\text{год} \quad (3.6)$$

Цього значення достатньо для підтримання належного повітрообміну. Його можна досягти шляхом природного провітрювання або встановленням припливно-витяжної вентиляції з продуктивністю щонайменше $70 \text{ м}^3/\text{год}$.

Також слід враховувати тепловиділення від обладнання та людини. Сумарне теплове навантаження можна обчислити за формулою:

$$Q = Q_{\text{пк}} + Q_{\text{м}} + Q_{\text{осв}} + Q_{\text{л}} \quad (3.7)$$

де $Q_{\text{пк}}$ – тепла потужність системного блоку (100 Вт);

$Q_{\text{м}}$ – монітор (30 Вт);

$Q_{\text{осв}}$ – освітлення (60 Вт);

$Q_{\text{л}}$ – тепло, що виділяється людиною (~ 100 Вт).

$$Q = 100 + 30 + 60 + 100 = 290 \text{ Вт}. \quad (3.8)$$

Цей рівень тепловиділень не потребує додаткового кондиціонування за умови якісної вентиляції.

Щодо ергономіки, дотримання вимог ДСТУ EN 527-1:2019 гарантує безпечне та комфортне робоче місце: стільниця розміром не менше 1200×600 мм, екран розташований на відстані 50–70 см від очей, верхній край – на рівні очей. Крім того, стілець має бути регульованим, із підтримкою спини й підлокітниками.

Для зниження ризиків перевтоми рекомендується щогодини робити перерви на 5 – 10 хвилин, виконувати вправи для очей і коротку розминку. Це зменшує м'язову напругу, втому та підвищує ефективність праці.

Таким чином, умови праці під час розробки та експлуатації системи комп'ютерного зору для автоматизованого керування промисловими роботами відповідають чинним вимогам з охорони праці. Робоче місце розробника обладнане відповідно до санітарно-гігієнічних та ергономічних нормативів, що мінімізує ризики для здоров'я працівника. Завдяки грамотній організації освітлення, мікроклімату, вентиляції та ергономіки, а також за рахунок використання програмного забезпечення, що не створює надмірного навантаження, створюються безпечні, комфортні та продуктивні умови праці.

3.4 Теорія автоматичного управління в автоматизованій системі комп'ютерного зору.

У межах теорії автоматичного управління розглядається контур автоматичного керування, в якому вихідні сигнали комп'ютерного зору виступають у ролі вхідних дій до системи управління роботом, а останній виконує конкретні операції у відповідь.

У спрощеній моделі об'єкт керування – промисловий робот.

Джерело збурення – вхідна інформація з камери (зміна розташування/відсутність/дефект виробу).

Керуюча дія – сигнал (цифровий або аналоговий) на виконавчі механізми робота.

Передатна функція об'єкта (робота) може бути представлена як:

$$W_S = \frac{K}{(T_1s+1)(T_2s+1)} \quad (3.9)$$

де K – коефіцієнт підсилення виконавчого актуатора;

T_1, T_2 – сталі часу обробки зображення і механічного виконання дії відповідно.

Щодо дослідження системи автоматичного управління – аналіз стійкості системи.

Алгебраїчні критерії – оцінка характеристичного рівняння системи за критерієм Гурвіца.

Частотні методи – аналіз амплітудно-фазових характеристик (Bode, Nyquist).

Аналіз якості перехідного процесу проводиться в Simulink/Matlab: час реакції системи на новий об'єкт, затримки в розпізнаванні і виконанні дії, точність формування сигналу керування при виявленні браку.

Для зменшення похибок та затримок – застосовується ПД-регулятор, що дозволяє швидко реагувати на зміни у стані об'єкта:

$$G(s) = Kp + \frac{Ki}{s} + Kd^s \quad (3.10)$$

Параметри регулятора оптимізуються методом мінімізації інтегральної квадратичної похибки (ISE) або ІТАЕ.

Щодо побудови області стійкості: досліджується вплив зміни параметрів модуля зору (наприклад, частоти кадрів або порогу виявлення дефекту) на стабільність і коректність дій робота. Визначаються допустимі межі затримки (τ), за яких контур залишається стійкими.

Система комп'ютерного зору успішно інтегрована в контур автоматичного керування, формуючи достовірні сигнали для дій промислового робота. Математичне моделювання підтвердило, що система є стійкою та забезпечує якісне керування при динамічній зміні ситуації на виробничій лінії. Оптимізований ПД-регулятор дозволяє компенсувати затримки, забезпечуючи швидку та точну реакцію на виявлені дефекти.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було проаналізовано предметну область, пов'язану з комп'ютерним зором у сфері автоматизованого управління промисловими роботами. Було розглянуто сучасні підходи до побудови систем технічного зору, а також обґрунтовано доцільність використання методів глибинного навчання, зокрема нейронної мережі YOLO (You Only Look Once) для задач виявлення об'єктів у реальному часі.

На основі вивченого матеріалу було сформовано технічне завдання, визначено мету, задачі, функціональні вимоги до системи, а також вимоги до точності розпізнавання та обробки зображень. Було розглянуто переваги різних підходів до обробки візуальної інформації та обґрунтовано вибір YOLOv8 як основи для розробки системи.

У рамках реалізації було зібрано та анотовано датасет з зображеннями робочої зони, який включає зображення еталонного об'єкта – комп'ютерної миші – та сторонніх об'єктів, що можуть свідчити про порушення технологічного процесу. Для зручності збору та підготовки даних використовувалася платформа Roboflow. Навчання моделі було виконано у середовищі Google Colab з використанням бібліотеки Ultralytics YOLOv8.

У розробленій системі реалізовано виявлення об'єктів, класифікацію миші як справної або дефектної за допомогою метрики SSIM, а також виявлення сторонніх предметів. Було розроблено програмний модуль, який виконує обробку зображень, виводить результати з візуалізацією об'єктів та їх статусом, що дозволяє оперативно реагувати на порушення.

Для реалізації програмної частини використовувалися Python, бібліотеки OpenCV, Ultralytics YOLO, а також додаткові інструменти для обробки зображень і збереження результатів.

Таким чином, було створено функціональну модель системи комп'ютерного зору, яка може бути інтегрована у промислове середовище для автоматизованого контролю виробів і виявлення відхилень у робочій зоні роботизованих комплексів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008:2015. Документація. Звіти у сфері науки та техніки структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, – 2017. – 29 с.
2. Методичні вказівки з підготовки кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти денної і заочної форми навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І.Ш. Невлюдов, О.І. Филипенко, О.В. Токарева, С.П. Новоселов, О.В Сичова. – Харків: ХНУРЕ, 2023. – 64 с
3. Гладков С.С. Особливості комп'ютерного зору/С.С. Гладков; наук. кер. СВ Шестопапов//Стан, досягнення і перспективи інформаційних систем і технологій: матеріали XVII Всеукр. наук.-техн. конф. молодих вчен., асп. та студ., Одеса, 19 квіт.2017 р./Одес. нац. акад. харч. технологій, Ф-т інформ. технологій та кібербезпеки. – Одеса: ОНАХТ.– 2017. – 2 с.
4. Євсєєв, В., and К. Стеценко. Розробка структурної схеми комп'ютерного зору для мобільного роботу типу SPOT. Diss. Черкаського національного університету імені Богдана Хмельницького, – 2023. – 202 с.
5. Грицик, В. В. "Обробка складних зображень та їх розпізнавання в інформаційно-аналітичних системах комп'ютерного зору." Доп. НАН України. – 2009. – 41 с.
6. Алексюк, Ю. А. "Детектування і розпізнавання об'єктів за допомогою комп'ютерного зору." Сучасні інформаційні технології в автоматизованому виробництві та управлінні : зб. наук. праць. – Львів : Видавництво Львівської політехніки, – 2018: – 102 с.
7. Грицик, В., and О. Пастух. "Порівняльне оцінювання розроблених інформаційних технологій для сучасних систем комп'ютерного зору." бірник

наукових праць професорсько-викладацького складу ДонНУ імені Василя Стуса – 2014. – 14 с.

8. Черненко, К. С., М. В. Макаров, and Ю. С. Антонов. "Бібліотеки комп'ютерного зору та проблеми керування транспортними засобами." Збірники наукових праць професорсько-викладацького складу ДонНУ імені Василя Стуса. – 2019: – 115 – 117 с.

9. Бунін, Ю. В., et al. "Оцінка розмірів стороннього тіла легені з застосуванням алгоритмів комп'ютерного зору." Експериментальна і клінічна медицина 89.4 – 2020: – 94 с.

10. Черногор, Максим Станіславович. "Моделювання систем комп'ютерного відтворення та розпізнавання цифрових зображень." робота на здобуття кваліфікаційного ступеня магістра; спец: 6.04.0301 – прикладна математика / М.С. Черногор; наук. кер. Т.С. Сушко. - Суми: СумДУ, – 2018. – 43 с.

11. Калашников М. Ю. Розробка методу ідентифікації деталей для процесу сортування на базі комп'ютерного зору : пояснювальна записка до атестаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології / М. Ю. Калашников ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків, 2020. – 80 с.

12. Велет А. В. Розробка автоматизованої системи термінального доступу до виробничого обладнання з використанням комп'ютерного зору : пояснювальна записка до кваліфікаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології / А. В. Велет ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки – Харків, 2022. – 78 с. Семеріков, С. О., et al. "Розробка системи комп'ютерного зору на основі машинного навчання для освітніх цілей." Educational Dimension 5.57 – 2021: – 60 с.

13. Сорока Д. О. Використання комп'ютерного зору для ідентифікації змісту етикеток харчових продуктів / Д. О. Сорока, наук. кер. Н. М. Сердюк //

Результати розвитку наукової думки: 2021: матеріали I Міжнародної студентської наукової конференції (Т. 2), м. Вінниця, 24 грудня, 2021 рік / ГО «Молодіжна наукова ліга». – Вінниця : ГО «Європейська наукова платформа», 2021. – С. 15 – 16.

14. Кондратюк М. В. Автоматизована система позиціонування маніпулятора із застосуванням оптичної системи робота : пояснювальна записка до атестаційної роботи здобувача вищої освіти на другому (магістерському) рівні, спеціальність 172 – Телекомунікації та радіотехніка / М. В. Кондратюк ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки – Харків, 2021 – 79 с.

15. Комплекс навчально-методичного забезпечення навчальної дисципліни «Безпека праці в індустрії ІТ-технологій» підготовки освітнього рівня бакалавр усіх спеціальностей та усіх напрямів університету[<http://catalogue.nure.ua/knmz>] / ХНУРЕ; розроб.: Т. Є. Стиценко, Г. В. Пронюк, Н. М. Сердюк. – Харків, 2017. – 122 с.