

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет робототехніки та приладобудування

Кафедра автоматизації та систем неруйнівного контролю

До захисту допущено:

Завідувач кафедри

_____ Юрій КИРИЧУК

«___» _____ 20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Комп'ютерно-інтегровані
системи та технології в приладобудуванні»
спеціальності 151 «Автоматизація та комп'ютерно-інтегровані
технології»**

**на тему: «Комп'ютерно-інтегрована система автоматизованого
контролю якості продукції на основі алгоритмів машинного
навчання»**

Виконав:

студент IV курсу, групи ПК-21

Дейна Дмитро Іванович _____

Керівник:

доц., к.т.н., доц. кафедри АСНК

Лисенко Юлія Юріївна _____

Рецензент:

доц., к.т.н., доц. кафедри КІТВП

Вислоух Сергій Петрович _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____

Київ – 2026 року

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проєкт	2	
2	A4	ДПБ ПК-21.07.1760.01 ПЗ	Пояснювальна записка	73	
3	A1	ДПБ ПК-21.07.1760.02СхС	Структурна схема	1	
4	A1	ДПБ ПК-21.07.1760.03СхС	Блочна електрична схема	1	
5	A1	ДПБ ПК-21.07.1760.02СхА	Блок-схема алгоритму	1	

				ДПБ ПК-21.07.1760.01 ПЗ		
	ПБ	Підп.	Дата	Відомість дипломного проєкту		
Розробн.	Дейна Д.І.					
Керівн.	Лисенко Ю.Ю.					
Консулт.						
Н/контр.						
Зав.каф.	Киричук Ю.В.			КПІ ім. Ігоря Сікорського Каф. АСНК Гр. ПК-21		
					Лист	Листів
					2	73

**Пояснювальна записка
до дипломного проєкту
на тему: «Комп'ютерно-інтегрована система
автоматизованого контролю якості продукції на основі
алгоритмів машинного навчання»**

Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет робототехніки та приладобудування
Кафедра автоматизації та систем неруйнівного контролю

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Комп'ютерно-інтегровані системи та технології в приладобудуванні»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Юрій КИРИЧУК

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Дейна Дмитро Іванович

1. Тема проєкту «Комп'ютерно-інтегрована система автоматизованого контролю якості продукції на основі алгоритмів машинного навчання», керівник проєкту Лисенко Юлія Юріївна, Доцент кафедри АСНК, затверджені наказом по університету від «26» 05 2026 р. № 1905-с

2. Термін подання студентом проєкту _____

3. Вихідні дані до проєкту : Розробити та описати автоматизовану систему візуального контролю якості продукції з використанням методів машинного навчання та комп'ютерного зору. Система повинна забезпечувати наступний функціонал: виявляти об'єкт дослідження в робочій зоні; виконувати візуальну обробку на виявлення дефектів поверхні; приймати рішення про відбракування; виконувати фізичне видалення об'єкта з виробничої лінії у разі виявлення браку.

4. Зміст пояснювальної записки: Вступ. Розділ 1 – Загальні питання автоматизації процесів. Використання алгоритмів машинного навчання та комп'ютерного зору в автоматизації. Розділ 2 – Автоматизована система

виявлення дефектів. Структурна схема системи. Алгоритми комп'ютерного зору. Розділ 3 – Апаратна реалізація автоматизованої системи виявлення дефектів. Розробка електричної схеми. Розділ 4 – Розпізнавання візуальних дефектів. Програмна реалізація алгоритму. Висновки

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): структурна схема автоматизованої системи виявлення візуальних дефектів методами комп'ютерного зору – 1 арк. ф.А1; блочна електрична схема автоматизованої системи виявлення візуальних дефектів методами комп'ютерного зору – 1 арк. ф.А1; блок-схема алгоритму роботи автоматизованої системи виявлення візуальних дефектів методами комп'ютерного зору – 1 арк. ф. А1

6. Дата видачі завдання 06.04.2026

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Отримання завдання на виконання	06.04.2026 – 12.04.2026	
2	Аналіз джерел. Розгляд питання задачі автоматизації	13.04.2026 – 19.04.2026	
3	Дослідження питання можливостей комп'ютерного зору в автоматизації	20.04.2026 – 26.04.2026	
4	Розробка концепції автоматизованої системи візуального контролю якості	27.04.2026 – 30.04.2026	
5	Розробка структурної схеми автоматизованої системи	01.05.2026 – 08.05.2026	
6	Вибір та обґрунтування вибору складових системи	09.05.2026 – 17.05.2026	
7	Розробка блочної електричної схеми системи	18.05.2026 – 21.05.2026	
8	Розробка та тестування алгоритму штучної нейронної мережі для виявлення візуальних дефектів	22.05.2026 – 29.05.2026	
9	Розробка блок-схеми алгоритму роботи системи	30.05.2026 – 06.06.2026	

Студент

Дмитро ДЕЙНА

Керівник

Юлія ЛИСЕНКО

АНОТАЦІЯ

Дипломний проєкт складається з вступу, чотирьох основних розділів, висновків та списку використаних джерел та літератури. Містить у складі 72 сторінки основного тексту, 32 ілюстрації, 1 таблицю та 16 посилань на використані літературні джерела.

Метою цієї роботи є опис та проєктування автоматизованої системи виявлення дефектів з використанням методів машинного навчання та алгоритмів комп'ютерного зору, яка буде здатна забезпечити оптимальний результат якості вихідної продукції у виробничих процесах.

Для виконання дипломної роботи було сформовано перелік завдань:

- виконання опису загальних принципів та концепції автоматизації та автоматизованого керування;
- розробити концепцію системи автоматизованого контролю якості з візуальним виявленням дефектів. Побудувати структурну схему автоматизованої системи виявлення візуальних дефектів на основі описаної концепції;
- виконати вибір складових елементів автоматизованої системи, зокрема центральної обчислювальної системи, камери, виконавчих механізмів та інших складових. Розробки блочну електричну схему системи з детальним описом зв'язків між блоками;
- розробити алгоритм навчання штучної нейронної мережі для розпізнавання візуальних дефектів. Виконати оцінку отриманих результатів з використанням нових вхідних даних. Побудувати блок-схема алгоритму автоматизованої системи візуального виявлення дефектів.

Ключові слова: автоматизована система контролю якості, автоматизація, система керування, комп'ютерний зір, машинне навчання

ABSTRACT

The diploma project consists of an introduction, four main chapters, conclusions, and a list of references. The project contains 72 pages of main text, 32 figures, 1 table, and 16 references.

The purpose of this work is to describe the design of an automated defect detection system using machine learning methods and computer vision algorithms capable of ensuring optimal output product quality in manufacturing processes.

To accomplish the objectives of the diploma project, the following tasks were defined:

- to describe the general principles and concepts of automation and automated control;
- to develop a concept for an automated quality control system with visual defect detection and to create a structural diagram of the automated visual defect detection system based on the proposed concept;
- to select the components of the automated system, including the central computing unit, camera, actuators, and other elements, and to develop a block electrical diagram of the system with a detailed description of the connections between the blocks;
- to develop a training algorithm for an artificial neural network for visual defect recognition, evaluate the obtained results using new input data, and create a flowchart of the automated visual defect detection system algorithm.

Keywords: automated quality control system, automation, control system, computer vision, machine learning.

ЗМІСТ

ВСТУП.....	10
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	11
1. ЗАГАЛЬНІ ПИТАННЯ АВТОМАТИЗАЦІЇ ПРОЦЕСІВ. ВИКОРИСТАННЯ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ ТА КОМП'ЮТЕРНОГО ЗОРУ В АВТОМАТИЗАЦІЇ.....	13
1.1. Автоматизація виробничих процесів.....	13
1.2. Концепція машинного навчання. Використання комп'ютерного зору в автоматизації.....	15
1.3. Огляд та аналіз наявних аналогів.....	18
1.4. Висновки до розділу 1.....	22
2. АВТОМАТИЗОВАНА СИСТЕМА ВИЯВЛЕННЯ ДЕФЕКТІВ. СТРУКТУРНА СХЕМА СИСТЕМИ. АЛГОРИТМИ КОМП'ЮТЕРНОГО ЗОРУ.....	23
2.1. Автоматизована система виявлення дефектів з використанням комп'ютерного зору.....	23
2.2. Розробка структурної схеми автоматизованої системи виявлення дефектів.....	24
2.3. Огляд алгоритмів аналізу зображень методами комп'ютерного зору. Бібліотека комп'ютерного зору OpenCV.....	27
2.4. Висновки до розділу 2.....	29
3. АПАРАТНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ВИЯВЛЕННЯ ДЕФЕКТІВ. РОЗРОБКА ЕЛЕКТРИЧНОЇ СХЕМИ.....	30
3.1. Обґрунтування вибору апаратних елементів системи.....	30
3.1.1. Блок обчислювальної системи.....	30
3.1.2. Вибір модуля камери.....	35
3.1.3. Вибір складових модуля виконавчих механізмів.....	37

3.1.4. Вибір складових модуля введення\виведення інформації.....	41
3.2. Електрична схема автоматизованої системи виявлення дефектів. Розрахунок загальної потужності системи.....	41
3.3. Висновки до розділу 3.....	49
4. РОЗПІЗНАВАННЯ ВІЗУАЛЬНИХ ДЕФЕКТІВ. ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ.....	50
4.1. Опис використаних програмних засобів.....	50
4.2. Підготовка та обробка вхідних даних.....	51
4.3. Опис алгоритму та програмних блоків.....	56
4.4. Аналіз результатів роботи алгоритму.....	62
4.5. Блок-схема алгоритму роботи автоматизованої системи візуального контролю якості методами комп'ютерного зору.....	66
4.6. Висновки до розділу 4.....	68
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ ТА ДЖЕРЕЛ.....	72
ДОДАТКИ	

ВСТУП

В сучасних умовах розвитку промисловості та переходу до нових концепцій виробництва особливої актуальності набувають розробка та впровадження автоматизованих виробничих систем.

Забезпечення стабільно високого рівня якості вихідної продукції є критично важливим фактором конкурентоспроможності сучасного підприємства.

Традиційні методи контролю якості, які в більшості ґрунтуються на візуальному контролі та перевірці оператором характеризується суб'єктивністю, обмеженою швидкістю обробки, досить високим ризиком помилки. У зв'язку з цим зростає потреба у розробці та впровадженні автоматизованих систем візуального контролю, які здатні працювати в режимі реального часу та забезпечувати високу точністю виявлення дефектів та високу швидкість обробки даних.

Одним з найперспективніших напрямків для вирішення цієї задачі є поєднання роботизованих систем з алгоритмами машинного навчання та комп'ютерного зору.

Поточний рівень розвитку програмних алгоритмів обробки зображень дозволяє забезпечити ефективний та швидкий процес аналізу, зокрема в режимі реального часу, виконуючи процес виявлення дефектів та інших відхилень з класифікацією за різними ознаками.

Таким чином розробка системи автоматизованого виявлення дефектної продукції використовуючи методи машинного навчання та комп'ютерного зору є актуальним та перспективним рішенням у сфері автоматизації виробничих процесів.

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ПЗ - програмне забезпечення;

ЕОМ - електронна обчислювальна машина;

САР - система автоматичного регулювання;

СК - система керування;

ОК - об'єкт керування;

ЗЗ - зворотний зв'язок;

НЛ - нечітка логіка;

ГА - генетичні алгоритми;

НМ - нейронні мережі;

МН - (ML - machine learning) машинне навчання;

МК - мікроконтролер;

ШНМ - штучна нейронна мережа;

Вт - ват, одиниця вимірювання значення потужності;

А - ампер, одиниця вимірювання значення струму;

В - вольт, одиниця вимірювання значення напруги.

1. ЗАГАЛЬНІ ПИТАННЯ АВТОМАТИЗАЦІЇ ПРОЦЕСІВ. ВИКОРИСТАННЯ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ ТА КОМП'ЮТЕРНОГО ЗОРУ В АВТОМАТИЗАЦІЇ

1.1. Автоматизація виробничих процесів

Автоматизація є процесом, що передбачає поєднання технічних та програмних засобів, які у сукупності призначені для виконання різноманітних завдань без, або з мінімальною участю у цьому процесі людини. Як правило, людина у цій структурі займає роль спостерігача або контролера, що аналізує результат роботи та загальний стан системи.

До технічних засобів відносяться електронні обчислювальні пристрої та системи, сенсори, виконавчі механізми, засоби введення та виведення інформації тощо.

В свою чергу до програмних засобів можна віднести програмне забезпечення (ПЗ) апаратного рівня, яке призначене для забезпечення роботи та взаємодії електронних пристроїв, а також прикладне ПЗ, яке представляє собою інтерфейс для керування та відображення показників роботи системи або процесу.

Основною метою автоматизації є - підвищення ефективності виконання процесів, мінімізація ймовірності виникнення помилок, забезпечення стабільності та повторюваності результатів тощо.

Також варто наголосити [1], що одна з ключових задач автоматизації полягає в отриманні та обробці інформації про стан всіх складових елементів виробничої системи та виробничого процесу.

В автоматизованих системах важливого значення набувають системи автоматичного керування та регулювання, структурна схема якої зображена нижче (див. рис. 1.1).

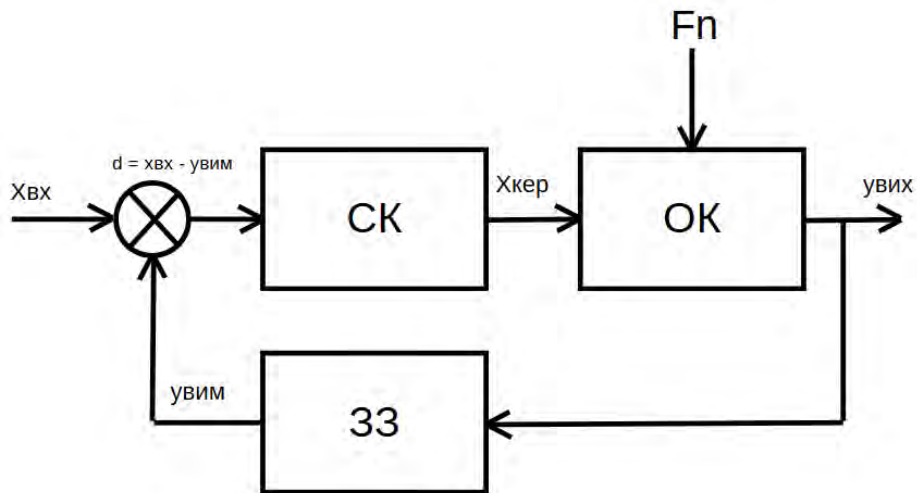


Рисунок 1.1. Структурна схема системи автоматичного керування

Структурна схема системи складається з трьох ключових блоків, а саме системи керування (СК), об'єкту керування (ОК) та блоку зворотного зв'язку (ЗЗ).

Такого роду автоматичні системи відносяться до систем керування з замкненим ланцюгом впливу. У даній системі керування відбувається на основі вхідного значення $x_{вх}$ що порівнюється із вихідним значенням блоку зворотного зв'язку $u_{вим}$. У результаті такого порівняння формується вихідний результат, який задає умову для функціонування блоку керування. Варто зазначити, що на об'єкт керування здійснюється вплив не тільки керуючого блоку, а і зовнішніх збурень, які у свою чергу спотворюють вихідний результат.

Автоматичне регулювання є зовнішнім впливом на регульований об'єкт для підтримання в процесі роботи певного заданого режиму, який характеризується одним або кількома параметрами. Такими параметрами є визначаючі командні параметри процесу, що проходять в об'єкті,

підтримання яких на заданому рівні забезпечує якісне виконання процесу відповідно до вимог виробництва [1].

У різних сферах практичної діяльності, що пов'язані із розв'язанням задач управління та контролю, оптимізації, моделювання, пошуку, вибору, розпізнавання та класифікації, виникає необхідність в інтелектуальній підтримці для подолання складних ситуацій, особливо з урахуванням обмежених ресурсів. Алгоритми та засоби штучного інтелекту знаходять практичне застосування у вигляді інтелектуальних технологій, які є універсальними та можуть бути використані незалежно від конкретної області застосування [2].

До переліку основних інтелектуальних технологій, які поширено використовуються при розробці складних, багатофункціональних систем керування зазвичай відносять [2]:

- нечітку логіку (НЛ);
- генетичні алгоритми (ГА);
- нейронні мережі (НА).

Варто зазначити, що до систем керування висуваються високі вимоги, зокрема забезпечення надійного функціонування об'єкта в різних режимах роботи, стійкість до різких змін і поступової деградації параметрів, а також здатність враховувати вплив зовнішніх факторів, таких як шум чи завади [2].

1.2. Концепція машинного навчання. Використання комп'ютерного зору в автоматизації

В умовах сучасного рівня розвитку інформаційних технологій машинне навчання є одним із ключових напрямів штучного інтелекту, який знаходить застосування для розв'язку широкого спектру практичних технічних задач. Ключовою особливістю таких систем є здатність навчатись

на основі нових вхідних даних, виявляти закономірності та приймати рішення без явно заданих умов на кожному кроці.

Методи машинного навчання знаходять широке застосування у різних галузях, зокрема і в автоматизації процесів, де їх впровадження дозволяє підвищити ефективність обробки інформації та точність прийняття рішень. Одним із найбільш перспективних напрямів є впровадження систем комп'ютерного розпізнавання образів та об'єктів з подальшою інтерпретацією та класифікацією.

Машинне навчання є розділом штучного інтелекту, що займається розробленням методів та алгоритмів, які надають можливість комп'ютерам та іншим обчислювальним системам навчатись на основі даних. Основна сутність такого навчання полягає у використанні попередньо зібраного набору даних, яка називається навчальна вибірка, для побудови моделі, яка буде здатна покращувати точність виконання завдань без явного програмування на кожному кроці чи виборі [3].

Алгоритми машинного навчання формують математичні моделі, які знаходять своє застосування у вирішенні різноманітних задач, а саме [3]: прогнозування, класифікації або прийняття рішень в різних прикладних задачах. Ці алгоритми знаходять своє застосування у різноманітних галузях, зокрема в медичній, робототехніці, системах виявлення шахрайства та кібербезпеки, фільтрації даних, розпізнаванні мовлення, автоматизації бізнес-процесів, комп'ютерному зорі та інших галузях, де застосування класичних алгоритмів є складним або зовсім неможливим.

Варто наголосити, що у контексті розв'язання прикладних задач машинне навчання часто розглядається як інструмент прогнозованої аналітики, що дозволяє виявляти приховані закономірності в даних та приймати обґрунтовані і зважені рішення [3].

В свою чергу комп'ютерний зір - це напрям штучного інтелекту, що займається задачами розробки методів та алгоритмів для отримання,

обробки, аналізу та інтерпретації візуальної інформації, що отримана із зображень або відеопотоків.

Основною метою галузі комп'ютерного зору є моделювання здатності людини сприймати навколишній світ та оточення за допомогою зору та перетворення візуальних даних у форму, яка є придатною для автоматизованого збору, аналізу для подальшого прийняття рішень.

Процес обробки зображень у системах де реалізовано алгоритми комп'ютерного зору зазвичай включає декілька етапів, а саме: отримання зображення, або з набору даних, або за допомогою камери якщо робота виконується в реальному часі; попередню обробку, яка включає фільтрацію шумів, нормалізацію тощо; виділення ознак, до яких відносяться контури, текстури, ключові точки; а також подальший аналіз та інтерпретацію отриманих результатів.

Відповідно, ключовими задачами комп'ютерного зору є виявлення та класифікація об'єктів на зображенні, сегментація, розпізнавання образів, відстеження визначених об'єктів у відеопотоці. Для вирішення цього спектра задач широко використовуються методи машинного навчання, зокрема глибокі нейронні мережі, які дозволяють досягати високої точності обробки візуальних даних.

Алгоритми та рішення галузі комп'ютерного зору знайшли широке застосування у різних галузях, зокрема в медицині, системах безпеки, транспорту, різного типу промисловості, робототехніці тощо. Особливого значення ці алгоритми набули у складі автоматизованих систем контролю якості продукції, зокрема для вирішення задач виявлення дефектів, аналізу поверхонь, класифікації виробів. Це в свою чергу дозволяє значною мірою підвищити ефективність виробництва, мінімізувати кількість браку та вплив людського фактору.

1.3. Огляд та аналіз наявних аналогів

У сфері автоматизації виявлення дефектів з використанням комп'ютерного зору існує ряд рішень різного рівня та цільового призначення. Для детального розгляду питання необхідно розглянути та проаналізувати декілька прикладів таких систем.

У розглянутій роботі [4] запропоновано підхід до побудови системи технічного зору, яка призначена для експериментальних досліджень дефектів матеріалів і виробів, з можливістю застосування сучасних методів обробки та ідентифікації об'єктів за їх зображенням. Структура системи зображена на рисунку нижче (див. рис. 1.2).

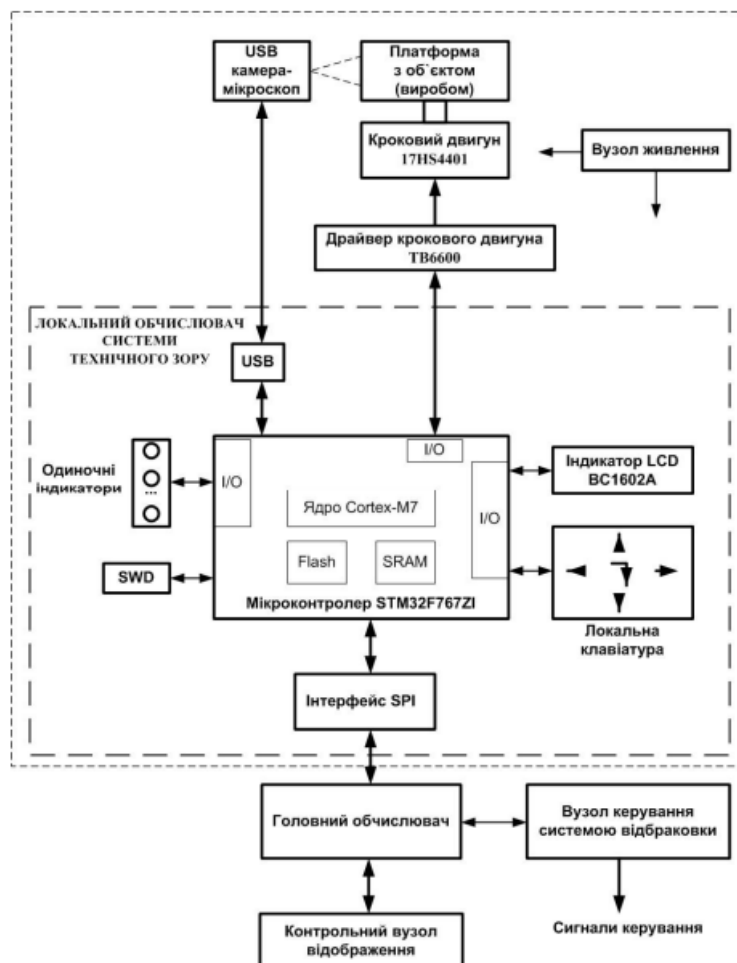


Рисунок 1.2. Структурна схема системи технічного зору для проведення досліджень дефектоскопії [4]

Описана в статті [4] система включає у своєму складі спеціальну установку для розміщення досліджуваного об'єкту з можливістю його обертання в горизонтальній площині, модуль отримання відеозображень за допомогою цифрової камери, а також підсистеми попередньої обробки відеокадрів на локальному обчислювачі з метою підвищення їх якості. Після цього оброблені зображення передаються на центральну обчислювальну систему, де виконується подальше виявлення та ідентифікація можливих дефектів, а також прийняття рішення щодо відбракування зразку продукції.

У структурі системи використовується кроковий двигун 17HS4401, який використовується як виконавчий механізм для обертання досліджуваного зразку. Керування двигуном забезпечується через драйвер ТВ6600 з допомогою локального обчислювального ядра на базі мікроконтролера Cortex-M7

Введення відеопотоку відбувається з використанням USB-камери, яка згідно технічних характеристик забезпечує достатню роздільну здатність для можливості виявлення дефектів розміром від 50 мікрон та більше. Введення зображення в локальну обчислювальну систему відбувається або у форматі безперервного відеопотоку або покадрово.

В свою чергу мікроконтролер обладнаний системою індикації, що складається з драйверу керування індикацією та світлодіодами, також має клавіатуру у якості системи введення та перемикання режимів керування двигуном, порт USB для підключення відеокамери та портом SWD для програмування Flash-пам'яті МК з відлагодженням програми у реальному часі.

Після попередньої обробки, покращення якості кадрів та відповідних перетворень, зображення передається на основний обчислювальний блок для подальшого аналізу та виявлення дефектів з прийняттям рішення про відбракування.

Також автор зазначає, що було розроблено спеціальне ПЗ для локальної обчислювальної системи, що включає у собі керування двигуном,

отримання зображень ймовірної дефектної зони у вигляді відеопотоку чи окремих кадрів, покращення якості зображення та передачі.

Розглянувши запропоновану систему у статті [4] можна виділити ряд недоліків, а саме:

- потенційно обмежена швидкодії, що в свою чергу зумовлено необхідністю попередньої обробки зображення та подальшої передачі на центральну обчислювальну систему для аналізу;

- описана архітектура системи не орієнтована на обробку великого потоку даних у реальному часі, що в свою чергу обмежує можливість її застосування у складі великих виробничих ліній з відносно високою швидкістю руху продукції.

Наступна розробка, яка буде розглянута у даному розділі - система розпізнавання дефектів на моделях 3D-друку [5].

У статті автор зазначає, що задача автоматизованого виявлення дефектів у 3D-друці є досить актуальною через обмеження традиційних методів контролю якості, зокрема таких, як візуальний огляд та використання спеціального обладнання. Ці підходи характеризуються високою вартістю, значними витратами часу та залежністю від людського фактора.

У зв'язку з цим доцільно реалізувати систему, яка буде використовувати сучасні алгоритми та методи глибокого навчання, що дозволить підвищити ефективність, швидкість та точність виявлення дефектів на об'єктах. Особливу увагу приділено архітектурі ResNet, як одній з найбільш ефективних моделей для аналізу зображень у задачах комп'ютерного зору.

Таке рішення обґрунтовано тим, що дана архітектура завдяки своїм перевагам здатна досягати необхідної якості та швидкодії. До таких переваг відносять [5]: використання залишкових зв'язків, що дозволяє мережі ефективно передавати інформацію між шарами, що, у свою чергу, забезпечує збереження важливих ознак навіть при дуже глибокій

архітектурі. Зазначається, що дана перевага особливо критична для розпізнавання тонких дефектів у зображеннях 3D-друк, де кожен дрібний сегмент має важливе значення.

Згідно інформації з джерела [5] Архітектура ResNet-50 використовує залишкові блоки, що в свою чергу сприяє ефективному проходженню інформації через усі шари нейронної мережі, що значно мінімізує проблему зникнення градієнтів. Це дозволяє моделі навчатися без втрати важливих ознак. Завдяки модифікаціям, таким як відмова від агресивного зменшення роздільної здатності у першому згортковому шарі та використанню адаптивного середнього пулінгу, модель здатна зберігати важливу інформацію про дрібні об'єкти, що дозволяє розпізнавати дефекти, які можуть проявлятися як невеликі або точкові області аномалій на поверхні друкованих деталей.

У підсумку можна зазначити, що незважаючи на високу ефективність описаної архітектури для вирішення задачі виявлення дефектів, запропонований підхід має ряд потенційних недоліків. Зокрема використання даної моделі характеризується високою обчислювальною складністю, що в свою чергу може негативно впливати на швидкодію. Особливо це може мати критичні наслідки у разі необхідності роботи в режимі реального часу.

Окрім того застосування такого типу складних моделей є дуже обмеженим на вбудованих платформах із нижчою обчислювальною потужністю, наприклад Raspberry Pi, що досить широко використовуються у виробничих системах.

У роботі також не розглянуто можливості оптимізації моделі, наприклад спрощення архітектури, що в свою чергу могло б підвищити ефективність її використання. У деяких випадках застосування таких складних моделей може бути надлишковим для певного спектру задач, що призводить до нераціонального використання обчислювальних ресурсів.

Після проведеного аналізу описаних аналогів доцільно зробити акцент на тому, що наявні рішення або орієнтуються на спеціалізовані дослідницькі установки, або потребують використання ресурсомістких алгоритмів глибокого навчання.

У першому випадку основними недоліками є складність інтеграції у виробничий процес та значно обмежена швидкодія у зв'язку з багатоступеневою обробкою даних.

У другому випадку проблемою є достатньо високі вимоги до обчислювальної потужності, що сильно ускладнює можливість використання таких систем на вбудованих платформах і як наслідок підвищує вартість реалізації повноцінної системи.

Відповідно, проведений аналіз показав відсутність універсального рішення, яке б могло одночасно забезпечити достатню точність виявлення дефектів, можливість роботи в режимі реального часу, оптимальні вимоги до апаратних ресурсів та простоту інтеграції у склад виробничих систем. У зв'язку з цим виникає необхідність розробки автоматизованої системи виявлення дефектів, яка буде поєднувати переваги сучасних методів машинного навчання та комп'ютерного зору з можливістю ефективної роботи та реалізації на доступних вбудованих обчислювальних платформах.

1.4. Висновки до розділу 1

У розділі було розглянуто загальні принципи автоматизації виробничих процесів, зокрема структуру та особливості функціонування автоматизованих систем керування. Описано значення технічних та програмних засобів у забезпеченні ефективності роботи таких систем.

Окрему увагу приділено сучасним методам штучного інтелекту, зокрема машинному навчанню та комп'ютерному зору, які відіграють ключову роль у задачах автоматизації аналізу даних та прийняття рішень.

Розглянуто ключові підходи до обробки візуальної інформації та визначено їх значення у системах автоматизованого контролю якості продукції.

В межах частини огляду та аналізу існуючих рішень було проаналізовано декілька підходів до побудови систем виявлення дефектів. Виявлено, що традиційні системи технічного зору мають обмеження, які в більшості пов'язані зі швидкістю, складністю архітектури та недостатньою адаптивністю до умови роботи у реальному часі. У свою чергу сучасні підходи на основі глибокого навчання забезпечують високу точність, проте складні моделі характеризуються значною обчислювальною складністю та обмеженою придатністю для використання на платформах з нижчою обчислювальною потужністю.

Таким чином, аналіз існуючих підходів показав необхідність розробки системи автоматизованого виявлення дефектів, яка буде об'єднувати необхідний рівень точності та достатньо високу швидкість з можливістю ефективного використання у реальних виробничих умовах.

2. АВТОМАТИЗОВАНА СИСТЕМА ВИЯВЛЕННЯ ДЕФЕКТІВ. СТРУКТУРНА СХЕМА СИСТЕМИ. АЛГОРИТМИ КОМП'ЮТЕРНОГО ЗОРУ

2.1. Автоматизована система виявлення дефектів з використанням комп'ютерного зору

Сучасні методи та підходи до вирішення задачі контролю якості, що базуються на візуальному огляді або використанні спеціалізованого обладнання, мають ряд обмежень, зокрема високу вартість, складність масштабування, обмеження швидкодії та ефективності. У зв'язку з цим доцільним є застосування методів комп'ютерного зору, які дозволяють автоматизувати процес візуальної оцінки якості.

Значна увага приділяється питанню забезпечення балансу між точністю виявлення об'єктів та аналізу на наявність дефектів, швидкодією та обчислювальною ефективністю, що дозволить інтегрувати таку систему у склад сучасного виробничого процесу.

Розроблювана система призначена для автоматизованого виявлення дефектів продукції на основі візуальної оцінки стану об'єкта. Основною метою є підвищення ефективності контролю якості.

До основного переліку функцій цієї системи можна віднести такі:

- 1) отримання зображення з камери;
- 2) попередня обробка відеопотоку;
- 3) виявлення об'єктів;
- 4) аналіз об'єктів на наявність відхилень від норми;
- 5) формування результатів перевірки;
- 6) прийняття рішення про відбракування;
- 7) передача результатів до центральної системи.

Така автоматизована система повинна містити у своїй структурі перелік спеціалізованих модулів, які будуть відповідати за виконання

функцій отримання, збору та обробки даних. Узагальнено структуру системи можна описати наступним чином:

- 1) модуль отримання зображення;
- 2) модуль обробки вхідних даних;
- 3) модуль аналізу (визначена модель машинного навчання);
- 4) модуль прийняття рішення;
- 5) модуль виконавчих механізмів;
- 6) модуль інтерфейсу користувача.

Розроблювана система орієнтована на обробку візуальних даних у режимі реального часу, із забезпечення достатньої точності та ефективності використання обчислювальних ресурсів. Особлива увага приділяється можливості для подальшої інтеграції системи у виробничі процеси та її адаптації до умов обмежених апаратних ресурсів.

2.2. Розробка структурної схеми автоматизованої системи виявлення дефектів

Структурна схема автоматизованої системи виявлення дефектів з використанням комп'ютерного зору зображена на рис. 2.1.

Система складається з функціональних блоків, які у поєднанні забезпечують повний цикл автоматизованого виявлення, аналізу та усунення дефектного об'єкта з виробничої лінії.

Блок центральної обчислювальної системи (БЦОС) - це центральний елемент системи, який виконує функції керування, збору та обробки даних, а також прийняття рішень. Даний блок може бути реалізований з використанням мікрокомп'ютера, наприклад, Raspberry Pi. До БЦОС підключаються усі ключові блоки, такі як модуль виконавчих механізмів (МВМ), модуль камери (МК), модуль введення/виведення інформації (МВВД).

Центральна обчислювальна система виконує загальні операції керування та прийняття рішень, формує керуючі сигнали для підключених модулів, а також централізовано збирає від них інформацію для обробки.

Наступний ключовий блок системи - модуль камери (МК). Даний модуль є одним із ключових для реалізації системи з комп'ютерним зором, та фізично підключається до мікрокомп'ютера. Важливим параметром даного блоку є роздільна здатність камери та особливості передачі даних, так як це безпосередньо впливає на можливість точного виявлення дефектів.

Модуль камери напряму підключається до БЦОС через спеціальний інтерфейс. Після чого виконується ініціалізація та налаштування параметрів камери і після успішного завершення камера буде готовою до роботи.

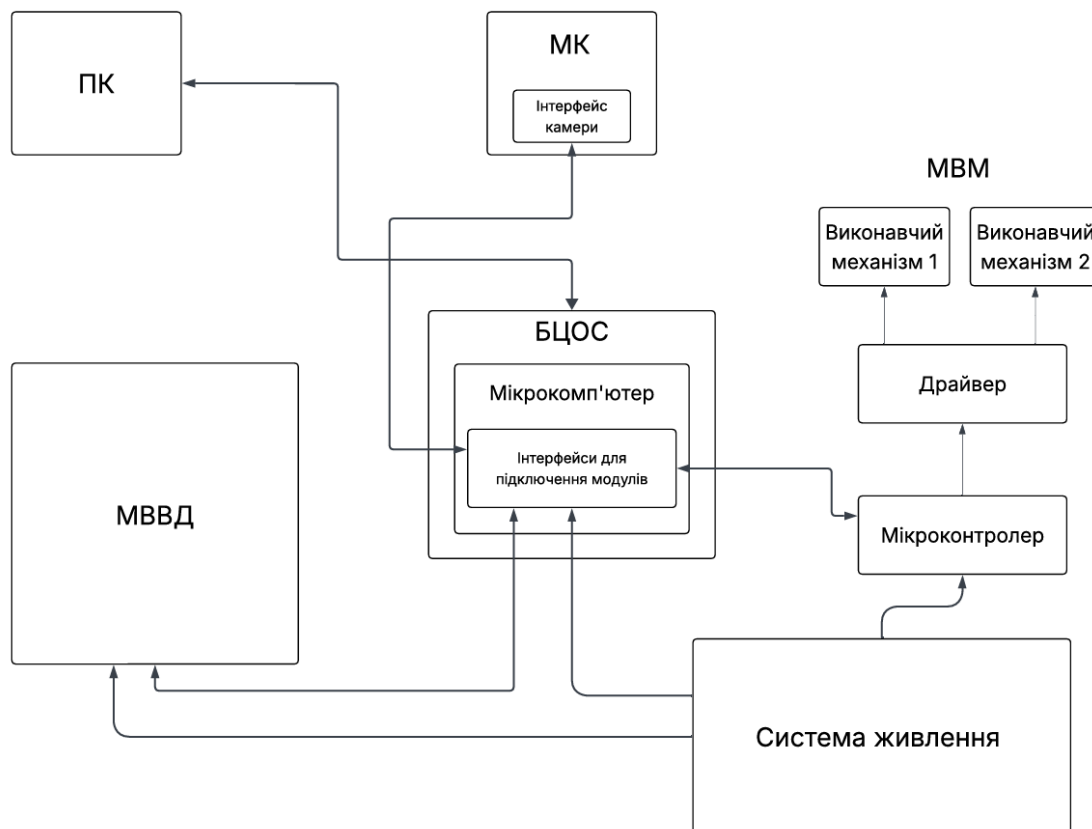


Рисунок 2.1. Структурна схема автоматизованої системи виявлення дефектів методами комп'ютерного зору

Наступний блок системи - модуль виконавчих механізмів (МВМ). Даний модуль підключається до БЦОС та призначений для виконання

механічних операцій з об'єктом дослідження, наприклад забезпечує функцію прибирання дефектних об'єктів з виробничої лінії. Цей модуль може бути реалізований шляхом використання спеціального мікроконтролера, який отримує команди від БЦОС, модуля керування виконавчими механізмами, наприклад драйвера двигунів, а також самих виконавчих механізмів.

Мікроконтролер у складі МВМ виконує функцію посередника між БЦОС та драйвером. МК централізовано отримує команди від БЦОС, наприклад, результат перевірки продукції - непридатний продукт, після чого МК виконує один із декількох варіантів алгоритму керування виконавчими механізмами для того, щоб усунути даний об'єкт з виробничої лінії. Таким чином забезпечується роздільна логіка керування, а саме, БЦОС дає команду і отримує результат про виконання без інформації про процеси, як саме це відбувається, а вся відокремлена логіка операцій реалізована тільки на МК.

Драйвер у складі МВМ виконує функцію посередника між виконавчими механізмами, такими як двигуни та сервоприводи, а МК. Це є необхідним, так як МК напряду не може керувати великою кількістю виконавчих механізмів, але окрім цього драйвер забезпечує можливість реалізації роздільного енергозабезпечення для.

Для забезпечення можливості виконання налаштування системи на місці та перегляду стану її роботи передбачено модуль введення/виведення даних (МВВД). Даний модуль може бути реалізовано у вигляді сенсорного дисплея для виведення та введення, або дисплея та модуля клавіатури МВВД підключається до БЦОС для приймання та передачі даних. Таким чином забезпечується можливість передачі команд від оператора до системи, та зворотне виведення параметрів на дисплей.

Для забезпечення функціонування системи передбачено два джерела живлення: для обчислювального блоку та для виконавчих механізмів. Така реалізація є важливою, так як блок виконавчих механізмів споживає значно

більше енергії та може створювати зворотній ефект, що може впливати на стабільність роботи всієї системи, а в окремих випадках може навіть призводити до пошкодження обладнання.

У складі системи передбачено інтерфейс для підключення до комп'ютера, що може бути корисним для аналізу результатів роботи, або для налаштування.

2.3. Огляд алгоритмів аналізу зображень методами комп'ютерного зору. Бібліотека комп'ютерного зору OpenCV

Не існує єдиного загальноприйнятого підходу до чіткого розмежування обробки зображень та суміжних галузей, таких як аналіз зображень та комп'ютерний зір. У ряді випадків це розмежування здійснюється на основі припущень, що обробка зображень - це процес, у якому як вхідні, так і вихідні дані мають вигляд зображень. Проте даний підхід є умовним та обмеженим. Зокрема, відповідно до цього навіть відносно проста задача, наприклад, обчислення середнього значення інтенсивності зображення, формально не відноситься до обробки зображень, що свідчить про недосконалість подібного поділу [7].

З іншого боку, існують такі галузі, як машинний зір, де кінцевою метою є комп'ютерна імітація людського зору, включаючи навчання, здатність до висновків та дій на основі спостережуваної інформації. Ця область сама по собі утворює лише одне із напрямів штучного інтелекту, метою якого є імітація інтелектуальної діяльності. Область, пов'язана з аналізом зображень (інакше звана «інтерпретація» зображень), займає проміжне положення між обробкою зображень і машинним зором [7].

У задачах автоматизованого виявлення дефектів продукції важливого значення набувають алгоритми аналізу зображень, що дозволяють забезпечити функції обробки, інтерпретації та класифікації візуальних даних. Методи які реалізує комп'ютерний зір забезпечує

можливість автоматичного виявлення та виділення інформативних ознак досліджуваних об'єктів та виявляти типові відхилення від заданих значень.

Сучасні алгоритми аналізу зображень можна поділити на декілька ключових груп, а саме:

- класичні методи обробки зображень, які ґрунтуються на математичних перетвореннях та фільтрації. До таких алгоритмів відносяться згладжування, фільтрацію шумів, підвищення контрастності та нормалізація зображень;

- алгоритми виділення ознак, серед яких можна виділити детектори контурів, методи знаходження ключових точок, а також сегментацію зображень. Ці підходи надають можливість отримувати структурну інформацію про об'єкти та їхні геометричні характеристики.

Для задач, які передбачають функцію виявлення особливих ознак, до яких можна віднести дефекти, широко використовуються методи порогової обробки, морфологічні операції, а також аналіз текстур та форм об'єктів. Такі методи є відносно простими, та можуть ефективно використовуватися у системах реального часу, так як не потребують значних обчислювальних ресурсів.

Але класичні методи мають ряд обмежень, які в більшості пов'язані з чутливістю до шумів, змін освітлення та складності обробки складних візуальних елементів. У зв'язку з цим значно поширюються методи машинного навчання, зокрема нейронні мережі, які в свою чергу дозволяють автоматично визначати ознаки, та забезпечують високий показник точності аналізу зображень.

Для реалізації алгоритмів комп'ютерного зору широко використовується бібліотека OpenCV, яка є однією з найпоширеніших відкритих платформ для роботи з аналізом зображень та відео.

Дана бібліотека містить великий набір інструментів для виконання простих та складних операцій, зокрема фільтрації, трансформації, сегментації, виявлення об'єктів на зображеннях та роботи з відеопотоками.

Функціонал бібліотеки OpenCV реалізовано декількома мовами програмування, зокрема Python та C++, що робить її універсальним інструментом для розробки системи комп'ютерного зору.

Таким чином, використання сучасних алгоритмів цифрової обробки зображень та комп'ютерного зору дозволяє створювати ефективні системи автоматизованого аналізу зображень, які будуть здатні забезпечити достатній рівень ефективності у задачах контролю якості продукції.

2.4. Висновки до розділу 2

У розділі було розглянуто підходи до побудови автоматизованої системи виявлення дефектів об'єктів, використовуючи методи комп'ютерного зору. Обґрунтовано доцільність застосування методів машинного навчання та комп'ютерного зору у задачах контролю якості, що зумовлено їх ефективністю та швидкодією.

Розроблено та описано структурну схему автоматизованої системи, визначено її ключові функціональні модулі та описано принцип їх взаємодії. Запропонована структура дозволяє забезпечити повний цикл обробки візуальних даних від отримання до прийняття рішення щодо класифікації та усунення дефектної продукції.

Було проведено огляд основних алгоритмів аналізу зображень, включаючи як класичні методи обробки, так і сучасні підходи, що ґрунтуються на використанні машинного навчання. Описано їх переваги та обмеження у контексті інтеграції у системи обробки даних у реальному часі.

Окрему увагу приділено питанню використання функціоналу бібліотеки OpenCV, як інструменту для реалізації алгоритмів комп'ютерного зору, що забезпечує необхідний функціонал та рівень продуктивності для вирішення задачі класифікації дефектів.

3. АПАРАТНА РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ВИЯВЛЕННЯ ДЕФЕКТІВ. РОЗРОБКА ЕЛЕКТРИЧНОЇ СХЕМИ

3.1. Обґрунтування вибору апаратних елементів системи

3.1.1. Блок обчислювальної системи

Блок обчислювальної системи є ключовим елементом, який забезпечує виконання функцій керування, збору, обробки та аналізу даних. У зв'язку з цим вибір апаратної платформи повинен ґрунтуватися на відповідності її технічних характеристик вимогам поставленої задачі.

Так як для даної системи передбачається інтеграція алгоритму розпізнавання об'єктів та виявлення дефектів методами комп'ютерного зору, апаратна платформа повинна мати достатній рівень обчислювальної потужності.

Першим вибором для порівняння є одноплатний мікрокомп'ютер Raspberry Pi (див. рис. 3.1)



Рисунок 3.1. Одноплатний мікрокомп'ютер Raspberry Pi [8]

Даний комп'ютер є повноцінною обчислювальною системою, яка функціонує на базі операційної системи (ОС) Linux та підтримує можливість виконання прикладного ПЗ для обробки даних, включно з алгоритмами комп'ютерного зору.

Raspberry Pi має ряд особливостей, що робить його використання для задачі автоматизованого аналізу та обробки досить зручним, а саме:

- 1) має компактні габаритні розміри та характеризується досить низьким енергоспоживанням;
- 2) підтримує поширені інтерфейси для підключення зовнішніх пристроїв та модулів, наприклад: USB; HDMI; GPIO; CSI;
- 3) передбачено можливість підключення камери, датчиків та виконавчих механізмів.

Завдяки цим особливостям цей комп'ютер набув широкого поширення у вбудованих системах, автоматизації та задачах комп'ютерного зору, де важливими параметрами є компактність, доступність та достатня обчислювальна потужність.

До основних характеристик моделі Raspberry Pi , яка на поточний момент є останньою з доступних, можна віднести [7]:

- 1) центральний процесор Quad Arm Cortex-A76 на 2,4 ГГц;
- 2) оперативна пам'ять 2/4/8/16 ГБ LPDDR4X-4267 SDRAM;
- 3) подвійний вихід дисплея HDMI 4kp60 із підтримкою HDR
- 4) графічне ядро VideoCore VII з OpenGL-ES 3.1, Vulkan 1.2
- 5) наявність процесора датчика зображення Raspberry Pi (ISP);
- 6) 802.11ac дводіапазонний Wi-Fi
- 7) Bluetooth 5.0 (з підтримкою BLE)
- 8) 2x USB 3.0 (з можливістю одночасної повної пропускної здатності) та 2x USB 2.0;
- 9) подвійні 4-канальні трансивери MIPI CSI/DSI, підтримка 2x дисплеїв; або 2x камер; або 1x дисплей + 1x камера;
- 10) 40-контактний роз'єм GPIO Raspberry Pi;

11) Годинник реального часу (RTC).

Всі вище описані характеристики роблять даний комп'ютер одним з кращих виборів з доступних.

Наступний варіантом є зменшена версія Raspberry Pi, саме - Raspberry Pi Zero W (див. рис. 3.2).



Рисунок 3.2. Мікрокомп'ютер Raspberry Pi Zero W [9]

Даний комп'ютер належить до сімейства Raspberry Pi але у свою чергу відрізняється малими габаритними розмірами і ще нижчим енергоспоживанням. Його цільовим призначенням є використання у складі вбудованих систем, де важливим є малі розміри та можливість бездротового підключення.

До його особливостей можна віднести [9]:

- 1) компактні розміри, що дозволяє інтегрувати плату в обмежений простір;
- 2) низьке енергоспоживання;
- 3) підтримка бездротових технологій передачі даних;
- 4) підтримка підключення камери.

До його основних характеристик відноситься:

- 1) ядро Broadcom BCM2835 з CPU і GPU:

- процесор CPU: ARM1176JZ-F з тактовою частотою 1 ГГц;

- графічний співпроцесор GPU: VideoCore IV з тактовою частотою 400 МГц;

- 2) оперативна пам'ять: 512 МБ;
- 3) наявність портів Mini HDMI і Micro USB з OTG;
- 4) подача живлення від Micro USB;
- 5) стандартний 40-піновий роз'єм GPIO для плат розширення;
- 6) композитний відео-роз'єм;
- 7) підтримка камери через інтерфейс CSI.

Відповідно варто зазначити, що дана платформа має обмежені обчислювальні ресурси, зокрема однопоточний процесор та невеликий обсяг оперативної пам'яті, що значною мірою обмежує можливість використання цієї платформи для ресурсоємних задач, зокрема обробка відеопотоку в реальному часі або застосування складних моделей машинного навчання.

Таким чином, Raspberry Pi Zero W може використовуватись для простих або допоміжних задач, але для реалізації повноцінних систем аналізу та комп'ютерного зору доцільніше використовувати більш продуктивні платформи. Для зручності порівняння описаних платформ сформовано порівняльну таблицю характеристик 3.1.

Табл. 3.1. Порівняння характеристик апаратних платформ

Характеристика	Raspberry Pi 5	Raspberry Pi Zero W
Процесор	Cortex-A76, 4 ядра, 2.4 ГГц	Cortex-A7, 1 ядро, 1 ГГц
Оперативна пам'ять	4 / 8 ГБ LPDDR4X	512 МБ LPDDR2
Обробка відео	4Kp60 (апарат. декод.)	1080p30 (обмежено)
Підтримка камери	2× MIPI CSI роз'єми	1× MIPI CSI роз'єм
Інтерфейси	USB 3.0, HDMI, PCIe, GPIO	micro-USB, mini-HDMI, GPIO
Бездротовий зв'язок	Wi-Fi 802.11ac, BT 5.0	Wi-Fi 802.11n, BT 4.1
Габарити	85 × 56 мм	65 × 30 мм
Енергоспоживання	~5–10 Вт	~0.5–1 Вт
Придатність для обробки сигналу в реальному часі	Повноцінна	Обмежена

Альтернативною платформою для забезпечення центральної обчислювальної системи може бути Orange Pi (див. рис. 3.3).



Рисунок 3.3. Мікрокомп'ютер Orange Pi [10]

Цей одноплатний мікрокомп'ютер також призначений для виконання ресурсоємних обчислювальних задач та використовується у складі вбудованих систем, зокрема і у сферах штучного інтелекту та комп'ютерного зору.

Завдяки високому рівню продуктивності, дана платформа може використовуватись для реалізації системи комп'ютерного зору, яка буде працювати в режимі реального часу та виконувати задачі виявлення дефектів продукції, але варто зазначити, що вона має вищий показник енергоспоживання, характеризується складністю налаштування ПЗ та меншою підтримкою у порівнянні з більш поширеними платформами.

Цей мікрокомп'ютер має ряд наступних технічних характеристик[10]:

1) центральний процесор: 8-ядерний 64-бітний 4xCortex-A76 та 4xCortex-A55, велике ядро: підтримує частоту 2.4 ГГц та мале ядро - 1.8 ГГц;

2) відеоядро Arm Mali-G610 MP4 "Odin" GPU, яке має сумісність з: OpenGL ES1.1/2.0/3.2, OpenCL 2.2 та Vulkan 1.2;

3) нейромережевий процесор (NPU): вбудований прискорювач AI NPU з підтримкою INT4/INT8/INT16 змішаної операції до 6 TOPS;

- 4) оперативна пам'ять: 4 ГБ/8 ГБ/16 ГБ/32 ГБ (LPDDR4/4x);
- 5) пам'ять: QSPI Nor FLASH 16 МБ та підтримка MicroSD (TF) картки;
- 6) підтримка камери: MIPI CSI 4Lane.

3.1.2. Вибір модуля камери

Одним із ключових елементів для реалізації автоматизованої системи виявлення дефектів є модуль камери, який забезпечує можливість отримання відео та зображень для подальшої обробки. Відповідно, від характеристик камери безпосередньо залежать точність та ефективність роботи всієї системи.

При виборі модуля камери необхідно враховувати ряд таких важливих характеристик:

- 1) роздільна здатність камери;
- 2) частота кадрів;
- 3) якість сенсора та чутливість до освітлення;
- 4) тип підключення;
- 5) можливість роботи в реальному часі.

Також варто зазначити, що для реалізації системи доцільно використовувати камери, які підключаються до комп'ютера, використовуючи інтерфейс CSI. Це важливий параметр, так як даний спосіб підключення забезпечує достатньо високу швидкість передачі даних та менші затримки у порівнянні з тими, які підключаються до USB-портів.

Камери бувають аналоговими та цифровими, відповідно:

- 1) аналогові камери передають відеосигнал у вигляді безперервного електричного сигналу. Для подальшої обробки такого сигналу необхідно використовувати додаткові пристрої для перетворення аналогового сигналу у відповідний до обробки формат. До переваг аналогових камер можна віднести простоту реалізації та відносно низьку вартість, але їх

використання обмежується нижчою якістю зображення, чутливістю до завад та складністю інтеграції у склад сучасних цифрових систем;

2) цифрові камери, на відміну від аналогових, формують зображення у цифровому форматі безпосередньо на рівні сенсора, що дозволяє виконувати передачу даних без додаткових перетворень. Ці камери підтримують сучасні інтерфейси підключення, що значною мірою спрощує інтеграцію з обчислювальними системами.

Ключовими перевагами використання цифрових камер є:

- висока якість та роздільна здатність зображення;
- стійкість до шумів та завад;
- можливість обробки даних у реальному часі;
- простота інтеграції з програмними засобами комп'ютерного зору.

З урахуванням вимог до сучасних автоматизованих систем контролю якості, зокрема необхідністю обробки зображень у режимі реального часу та використанням алгоритмів комп'ютерного зору, доцільним є застосування саме цифрових камер.

Розглянемо детальніше варіанти доступних камер для мікрокомп'ютера Raspberry Pi (див. рис. 3.4).

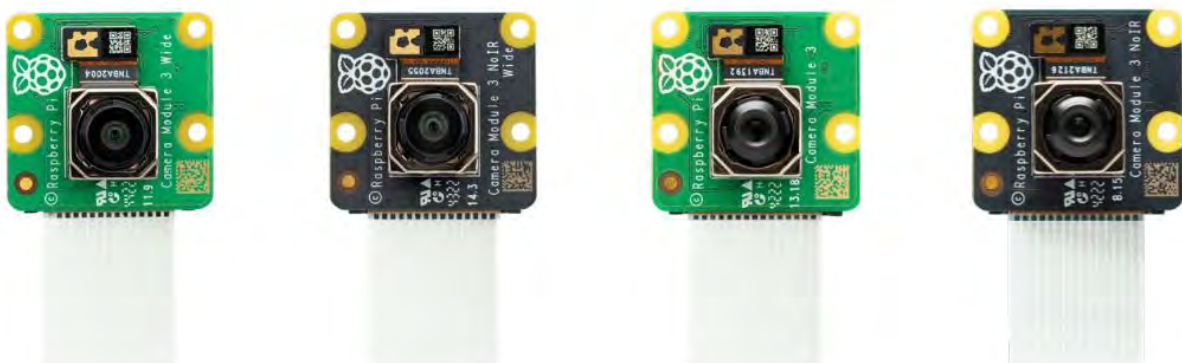


Рисунок 3.4. Модулі камери для Raspberry Pi [10]

На рис. 3.4 показано чотири варіанти камери, а саме:

- Camera module 3 Wide;
- Camera module 3 NoIR Wide;

- Camera Module 3;
- Camera Module 3 NoIR.

Зазначені модулі відрізняються між собою характеристиками об'єктива, зокрема кутом огляду, а також наявністю або відсутністю інфрачервоного фільтра. Версії Wide мають збільшений кут огляду, що надає можливість охоплювати більшу область спостереження, однак це може спричиняти спотворення кадру по краях.

У свою чергу моделі NoIR не містять у своєму складі інфрачервоного фільтра, що дозволяє їх використовувати в умовах недостатнього освітлення або разом з інфрачервоним підсвічуванням. Але при звичайному освітленні такі камери можуть передавати кольору з певним викривленням.

Звичайні моделі Camera Module 3 здатні забезпечити точнішу передачу кольорів та є придатними для використання при достатньому рівні освітленості, що є важливим фактором для реалізації функції виявлення дефектів в автоматизованому режимі.

Усі перелічені модулі камери мають наступні характеристики [10]:

- 1) 12-ти мегапіксельний датчик зображення CMOS Sony IMX708 із зворотними підсвічуванням;
- 2) високе співвідношення сигнал\шум (SNR);
- 3) вбудована 2D динамічна корекція пікселів дефектів (DPC);
- 4) фазове автофокусування (PDAF) для швидкого автофокусування;
- 5) функція QBC Re-mosaic;
- 6) режим HDR (до 3 мегапікселів);
- 7) послідовний вихід даних (CSI-2).

3.1.3. Вибір складових модуля виконавчих механізмів

Модуль виконавчих механізмів представляє собою окрему підсистему. Такий модуль для автоматизованої системи усунення браку з лінії може складатися з:

1) мікроконтролера - виконує обчислювальні операції і керує виконавчими механізмами на основі результату виявлення браку;

2) драйвера виконавчих механізмів - виконує функцію інтерфейса для підключення та керування механізмами та забезпечує розподілене живлення;

3) виконавчий механізм - сервоприводи, двигуни та інші механізми які призначені для виконання механічного впливу.

У якості обчислювального ядра можна використовувати різні платформи розробки, у даному випадку доречно використовувати платформу Arduino Nano на базі 8-ми бітного мікроконтролера ATmega328 (див. рис. 3.5).



Рисунок 3.5. Платформа розробки Arduino Nano [11]

Розглянемо детальніше технічні характеристики даної платформи для розробки [11]:

- 1) мікроконтролер: ATmega328
- 2) робоча напруга: 5В;
- 3) Flash пам'ять: 32 кБ з яких 2 кБ використовується як завантажувач;
- 4) SRAM: 2 кБ;
- 5) робоча частота: 16 МГц;
- 6) кількість аналогових каналів на вхід: 8;
- 7) EEPROM: 1 кБ;

- 8) граничне значення постійного струму на 1 вихідний пін: 40 мА;
- 9) рекомендована вхідна напруга: 7 - 12В;
- 10) кількість цифрових пінів: 22 (6 з яких ШІМ);
- 11) енергоспоживання: 19 мА.

Підсумовуючи описані характеристики даної платформи для розробки, можна зазначити, що Arduino Nano може ефективно використовуватись для забезпечення виконання функцій керування виконавчими механізмами, обробки сигналів від сенсорів, та забезпечення взаємодії між окремими складовими модулями автоматизованої системи.

У якості драйвера двигунів можна використати 16-ти канальний 12-ти бітний ШІМ модуль для серводвигунів (див. рис. 3.6).



Рисунок 3.6. 16-ти канальний 12-ти бітний ШІМ модуль для серводвигунів [12]

Даний модуль надає можливість керувати сервоприводами використовуючи інтерфейс I2C, що, у свою чергу, дозволяє зменшити кількість пінів для підключення та реалізувати відокремлене живлення для сервоприводів.

Розрядність 12 біт є достатньою для того, щоб забезпечити достатню точність формування сигналів керування, що є важливим для реалізації плавного переміщення та стабільного керування.

Також варто зазначити, що використання драйвера дозволяє підключати декілька сервоприводів з роздільним керуванням, що дозволяє збирати великі та складні конструкції.

Даний модуль має наступні характеристики [12]:

- реалізує керування використовуючи інтерфейс I2C;
- на платі драйвера встановлені додаткові резистори по кожному з виходів для зручного підключення світлодіодів;
- надає можливість каскадування до 62-х виконавчих механізмів;
- живлення від 3.3В з можливістю підключення до 5В пристроїв;
- надає можливість підключати 6В сервоприводи;
- має 6-ти розрядну I2C адресу;
- регульована частота ШІМ до 1.6 кГц;
- окремий вхід для джерела живлення.

У підсумку можна зазначити, що даний модуль є хорошим вибором для реалізації драйвера виконавчих механізмів для автоматизованої системи виявлення та усунення бракованої продукції.

У якості виконавчого механізму можна використати сервопривід MG90S (див. рис. 3.7).



Рисунок 3.7. Сервопривід MG90S [13]

Даний сервопривід має невеликі габаритні розміри, достатню швидкодію, а також металевий редуктор, що значно підвищує надійність та стійкість до механічних навантажень та зношення. Управління сервоприводом реалізується за допомогою сигналів ШІМ.

Сервопривід MG90S має володіє наступними характеристиками [13]:

- 1) робоча напруга: 4.8 - 6В / 1 Ват;
- 2) розміри: 32.2 x 31.7 x 12.3 мм;
- 3) висота від фланця до полиці кріплення: 8.5 мм;
- 4) кут обертання: 90о;
- 5) крутний момент: 1.9 кг/см (при живленні 4.8В), 2.2 кг/см (при живленні 6В);
- 6) діапазон робочої температури: 0 - 55 оС;
- 7) швидкість обертання 0.15с/60о (при 4.8В), 0.12с/60о (при 6В);
- 8) вага: 14 гр.

3.1.4. Вибір складових модуля введення\виведення інформації

Для введення та виведення інформації можна використовувати різноманітні пристрої: сенсорні дисплеї, клавіатуру, LCD-дисплей, набір модуля кнопок тощо. Для даної задачі використання складної системи введення та виведення інформації є недоречною, так як більшість налаштувань виконується через підключення до ПК.

Тому є доречним використати звичайний LCD-дисплей для виведення даних та модуль кнопок для введення простих команд.

У даному випадку розглядається LCD-дисплей 20x4 (див. рис. 3.8)



Рисунок 3.8. LCD-дисплей 20x4 [14]

Даний дисплей має достатній розміри екрану і кількість символів, для того, щоб забезпечити відображення всієї корисної інформації для користувача.

Даний дисплей володіє наступними характеристиками:

- 1) кількість символів в рядку: 20;
- 2) кількість рядків: 4;
- 3) інтерфейс підключення: I2C;
- 4) адреса пристрою на шині I2C: 0x27 або 0x3f;
- 5) чіп конвертера I2C: PCF8574A;
- 6) живлення: 5V;
- 7) розміри: 98x60x20 мм.

Дисплей може використовуватись для відображення поточного режиму роботи системи, результатів перевірки продукції, повідомлень про помилки, стан підключення складових модулів, а також для виведення службової інформації про процес роботи системи загалом.

У якості блоку введення інформації доцільно використати мембранну 16-ти кнопкову клавіатуру (див. рис. 3.9)



Рисунок 3.9. 16-ти кнопкова мембранна клавіатура [15]

Даний модуль призначений для введення команд керування, щоб забезпечити взаємодію користувача з системою.

Мембранна клавіатура характеризується компактними розмірами, простотою підключення та достатнім рівнем функціональності для реалізації ключових операцій керування та налаштування.

Даний модуль мембранної клавіатури володіє рядом наступних технічних характеристик:

- 1) розмір клавіатури: 77x70x0.8 мм;
- 2) довжина шлейфу: 86 мм;
- 3) маса: 10 гр.;
- 4) роз'єм: 8 пінів;
- 5) максимальне значення напруги: 35В, 100мА;
- 6) опір ізоляції: 100МОм, 100В;
- 7) діелектрична стійкість: 250VRms (50-60 Гц, 1 хв);
- 8) брязкіт контактів: менше 5 мс;
- 9) кількість натискань: 1 мільйон;

10) робоча температура: 0 - 70оС;

11) допустимий рівень вологості: 90-95%, 240 год.

Клавіатура у складі автоматизованої системи виявлення візуальних дефектів може використовуватись для запуску або зупинки роботи системи, перемикання режимів функціонування, навігації в меню, введення службових команд тощо.

3.2. Електрична схема автоматизованої системи виявлення дефектів. Розрахунок загальної потужності системи

У даному підрозділі розглядається електрична схема автоматизованої системи візуального виявлення дефектів з використанням методів комп'ютерного зору (див. рис. 3.10).

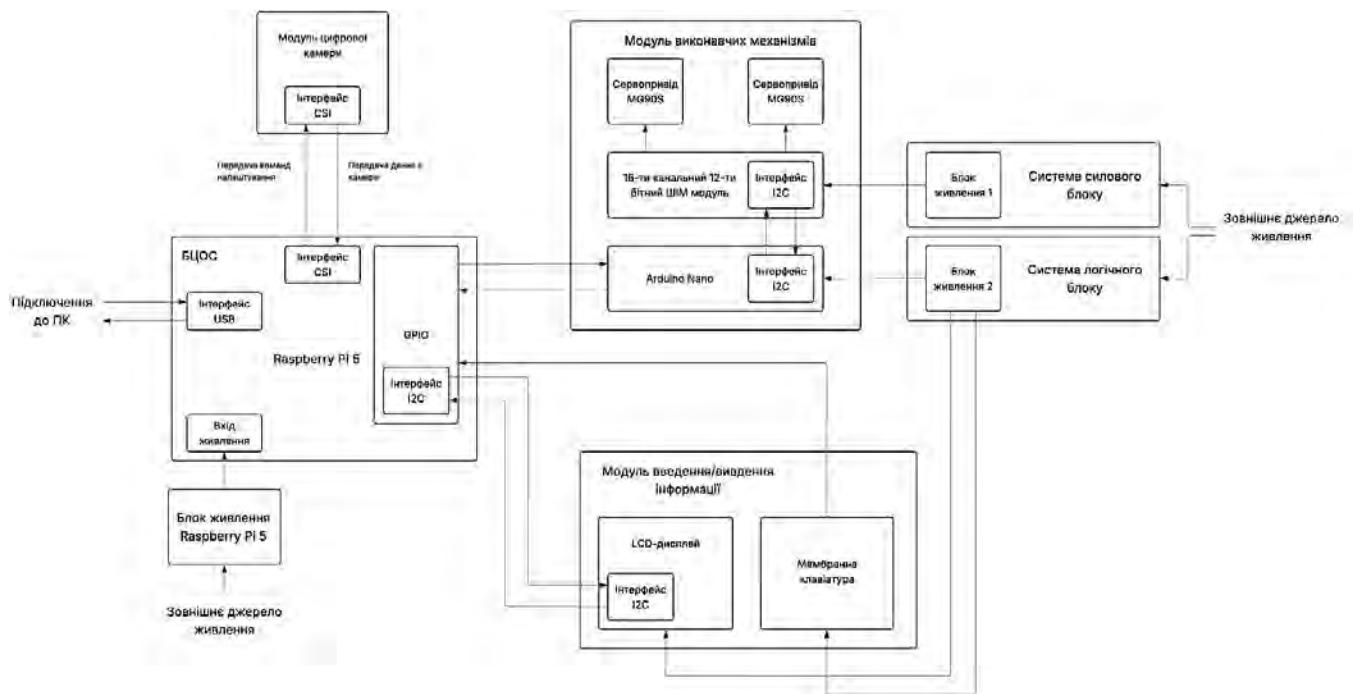


Рисунок 3.10. Електрично-структурна схема автоматизованої системи візуального виявлення дефектів продукції з використання комп'ютерного зору

На схемі зображено усі складові блоки автоматизованої системи та зв'язки між ними. Центральним керуючим блоком системи є БЦОС,

функцію якого виконує Raspberry Pi 5. Даний мікрокомп'ютер містить у своєму складі набір інтерфейсів, які дозволяють підключати різноманітні модулі з різними протоколами комунікації.

До БЦОС підключаються такі модулі:

1) модуль камери через інтерфейс CSI, що дозволяє забезпечити взаємодію з камерою напряду з мінімальними затримками та повним доступом до налаштувань;

2) модуль виконавчих механізмів через GPIO. Окремі порти GPIO мають призначення для приймання та передачі даних, наприклад, підтримують взаємодію між пристроями, використовуючи інтерфейс UART, сама комунікація відбувається за допомогою двох ліній - RX (приймання) та TX (передача). Підключення таким чином дозволяє мінімізувати затримки передачі даних та легко передавати команди на мікроконтролер модуля виконавчих механізмів;

3) модуль введення\виведення інформації складається з двох блоків, а саме: модуля дисплея, який підключається до БЦОС через інтерфейс I2C, що в свою чергу дозволяє передавати дані через дві лінії: SCL (шина синхронізації) та SDA (шина даних). Кожен пристрій, підключений до даного інтерфейсу, має власну унікальну адресу, що дозволяє підключати велику кількість модулів до одного фізичного інтерфейсу I2C. Другим складовим компонентом є – мембранна клавіатура, яка підключається до портів GPIO напряду.

Живлення БЦОС реалізовано з використанням спеціального блока живлення для обраної моделі Raspberry Pi.

Модуль виконавчих механізмів складається з чотирьох частин:

1) мікроконтролера Arduino Nano, який приймає команди від БЦОС та керує виконавчими механізмами через драйвер. Драйвер підключається до мікроконтролера використовуючи вже згаданий інтерфейс I2C;

2) драйвера виконавчих механізмів, який дозволяє забезпечити підключення великої кількості виконавчих механізмів та реалізувати

роздільну логіку керування кожним механізмом окремо. Також використання такого драйвера дає можливість забезпечити окреме живлення для кращого показника стабільності роботи системи;

3) а також два серводвигуна MG90S, які виконують функції механічного усунення відбракованого продукту з виробничої лінії. Сервоприводи підключаються до драйвера, використовуючи три контакти, а саме сигнальний, через який виконується передача команд, а також два контакти живлення.

Модуль введення та виведення інформації в свою чергу складається з двох частин, а саме:

1) LCD-дисплей, який завдяки спеціальній мікросхемі реалізовує можливість підключення до БЦОС через інтерфейс I2C;

2) мембранна клавіатура, яка напряду підключається до контактів GPIO, що дозволить обробляти кожен натиск кнопки незалежно одна від одної.

Також відображено систему живлення у вигляді двох блоків, а саме: блок живлення для логічної частини системи та силової частини системи. Така реалізація дозволяє забезпечити стабільність та захист системи від впливу силових компонентів.

Знаючи перелік та технічні характеристики всіх складових автоматизованої системи, виконаємо розрахунок загальної максимальної споживаної потужності.

Загальна формула для розрахунку потужності має наступний вигляд та вимірюється у ватах (Вт):

$$P = U \cdot I \quad (3.1)$$

де:

P - потужність, Вт;

U - напруга живлення, В;

I - струм споживання, А.

Відповідно, якщо система складається з декільком складових компонентів, то загальне значення струму можна визначити наступним чином:

$$I_{\Sigma} = I_1 + I_2 + I_3 + \dots + I_n \quad (3.2)$$

Після цього розраховується значення загальної потужності за допомогою наступного виразу:

$$P_{\Sigma} = U \cdot I_{\Sigma} \quad (3.3)$$

Або виконується розрахунок значення потужності споживання кожного окремого компонента незалежно, а після сумують потужності окремих елементів за допомогою наступного виразу:

$$P_{\Sigma} = P_1 + P_2 + P_3 + \dots + P_n \quad (3.4)$$

Але варто наголосити, що у разі, якщо в системі передбачено живлення окремих складових компонентів з різними значеннями напруги, наприклад, для сервоприводів від 6В, а логічні компоненти від 3,3В чи 5В, то розрахунок виконується виключно для окремих складових компонентів.

У даному випадку розрахунок буде виконуватися для кожного компонента окремо, для того, щоб мати дані про те, які компоненти споживають найбільше. Остаточний загальний вираз буде мати наступний вигляд:

$$P_{\Sigma} = P_{\text{БЦОС}} + P_{\text{МВМ}} + P_{\text{МВВД}} + P_{\text{МК}}$$

де:

$P_{\text{БЦОС}}$ - розрахункове значення потужності блоку центральної обчислювальної системи;

$P_{\text{МВМ}}$ - розрахункове значення потужності модуля виконавчих механізмів;

$P_{\text{МВВД}}$ - розрахункове значення потужності модуля введення\виведення даних;

$P_{\text{МК}}$ - розрахункове значення потужності модуля камери.

Блок центральної обчислювальної системи реалізовано з використанням мікрокомп'ютера Raspberry Pi, відповідно маючи технічні характеристики модуля, можемо виконати приблизний розрахунок споживаної потужності, відповідно:

$$P_{БЦОС} = U_{БЦОС} \cdot I_{БЦОС} = 5B \cdot 2,5A = 12,5 [Вт]$$

В свою чергу модуль камери споживає близько 0,25А при напрузі 5В, відповідно можемо визначити значення споживаної потужності:

$$P_{МК} = U_{МК} \cdot I_{МК} = 5B \cdot 0,25A = 1,25 [Вт]$$

Модуль введення та виведення інформації складається з двох компонентів, але враховуючи що для реалізації даного блоку були використані простий LCD-дисплей та мембранну клавіатуру, то загальне споживання такого модуля буде досить незначним. Визначимо орієнтовне значення споживаної потужності даного модуля:

$$P_{МВВД} = U_{LCD} \cdot I_{LCD} + U_{КЛАВ} \cdot I_{КЛАВ} = 5B \cdot 0,08A + 5B \cdot 0,01A = 0,45 [Вт]$$

Розрахунок споживаної потужності для модуля виконавчих механізмів буде включати більшу кількість складових, тому вираз для розрахунку буде мати наступний вигляд:

$$P_{МВМ} = P_{АРД} + P_{РРСА} + 2 \cdot P_{МГ90S}$$

Отримавши ключові показники з переліку даних складових можемо виконати розрахунок споживаної потужності модуля, відповідно:

$$\begin{aligned} P_{МВМ} &= P_{АРД} + P_{РРСА} + 2 \cdot P_{МГ90S} = I_{АРД} \cdot U_{АРД} + I_{РРСА} \cdot U_{РРСА} + 2 \cdot P_{МГ90S} = [Вт] \\ &= 5B \cdot 0,019A + 5B \cdot 0,01A + 2 \cdot 1Bm = 0,095Bm + 0,05Bm + 2Bm = 2,145 \end{aligned}$$

Маючи розрахункові значення всіх складових компонентів, можемо визначати значення загальної споживаної потужності системи:

$$P_{\Sigma} = P_{БЦОС} + P_{МВМ} + P_{МВВД} + P_{МК} = 12,5 + 2,145 + 0,45 + 1,25 \approx 16,35 [Вт]$$

У результаті розрахунку було отримано значення загальної споживаної потужності на рівні 16,35 Вт, але варто зазначити, що цей показник значною мірою залежить від режиму роботи, частоти виявлення браку, швидкості виконання операцій, і це в загальній сукупності може впливати на зміну значення потужності.

3.3. Висновки до розділу 3

У розділі було проведено аналіз можливих технічних рішень для побудови автоматизованої системи виявлення візуальних дефектів методами комп'ютерного зору, описано апаратні складові, а також їх технічні характеристики.

Важливу увагу було приділено вибору центральної обчислювальної системи, так як цей вибір має безпосередній вплив на можливість системи працювати з обробкою зображень в режимі реального часу.

Було розглянуто модуль камери, що є ключовим модулем для роботи системи візуального аналізу об'єктів. Виконано оцінку можливості застосування різних модифікацій камери, з зазначенням критично важливих параметрів для максимально коректного забезпечення роботи системи,

Було описано блок модуля виконавчих механізмів, який складається з мікроконтролера, драйвера виконавчих механізмів та сервоприводів. Даний блок забезпечить можливість фізичного усунення невідповідних об'єктів з виробничої лінії.

Описано блок введення та виведення даних для доступу до налаштування системи безпосередньо на місці. Даний блок включає у собі простий дисплей для відображення інформації, а також мембранну клавіатуру з можливістю введення набору простих команд.

Побудовано та описано електричну схему систему та її окремих складових модулів, а також проведено розрахунок значення загальної споживаної потужності, в результаті чого було отримано значення 16,35 Вт.

4. РОЗПІЗНАВАННЯ ВІЗУАЛЬНИХ ДЕФЕКТІВ. ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ

4.1. Опис використаних програмних засобів

Для реалізації алгоритму автоматизованого виявлення дефектів використовуються сучасні програмні засоби та бібліотеки комп'ютерного зору, що забезпечують функції обробки, аналізу та необхідної інтерпретації візуальних даних.

Зазвичай для реалізації подібних алгоритмів використовується мова програмування Python, яка отримала широке поширення у сфері задач машинного навчання та комп'ютерного зору, що в свою чергу пов'язана з простотою синтаксису, великій кількості спеціалізованих бібліотек та засобів для роботи із зображеннями та відеопотоком.

Одним із ключових програмних засобів є бібліотека OpenCV, яка зазвичай використовується для реалізації основних етапів обробки зображень. Функціонал бібліотеки забезпечує наступний функціонал:

- 1) отримання та обробку вхідного кадру;
- 2) зміна кольорових моделей;
- 3) фільтрацію шумів;
- 4) покращення якості зображення;
- 5) порогову обробку;
- 6) виділення контурів та ключових ознак;
- 7) сегментацію об'єктів;
- 8) аналіз форми та геометричних параметрів об'єктів.

Для реалізації функцій математичної обробки даних та роботи з багатовимірними масивами, що особливо важливо при роботі з обробкою кадрів, використовується бібліотека NumPy. Її застосування дозволяє ефективно та швидко виконувати операції з матрицями зображень та оптимізовувати швидкодію алгоритмів.

У випадку використання методів машинного навчання можуть використовуватись і інші спеціалізовані бібліотеки, а саме TensorFlow або PyTorch, які забезпечують можливість створення, навчання та використання нейронних мереж для задач класифікації та виявлення об'єктів.

Загальний принцип роботи даного алгоритму можна описати як послідовність декількох етапів:

1) здійснюється отримання вхідного зображення або кадру з відеопотоку. Після виконується попередня обробка зображення, яка зазвичай включає мінімізацію шумів, зміну кольорової моделі, нормалізацію яскравості та контрастності;

2) наступним кроком виконується виділення інформативних ознак об'єкта. Для виконання цієї задачі можуть використовуватись алгоритму контурного аналізу, сегментації або методи пошуку ключових точок. Отримані характеристики аналізуються для виявлення можливих відхилень від еталонного стану об'єкта.

3) у випадку реалізації алгоритмів машинного навчання оброблене зображення або набір ознак передається моделі класифікації, яка виконує аналіз та формує результат щодо наявності або відсутності дефекту. Після цього система формує результат для подальшої обробки або відображення.

4.2. Підготовка та обробка вхідних даних

Перед початком навчання моделі штучної нейронної мережі необхідно проводити попередню обробку та підготовку вхідних даних.

Підготовка вхідних навчальних даних є одним із найважливіших етапів, так як якість та структуру даних безпосередньо впливає на остаточний результат, точність, стабільність та ефективність роботи моделі. Будь-яка нейронна мережа, будь-якої складності та структури не зможе забезпечити якісного результату у разі використання попередньо не оброблених або недостатньо підготовлених даних.

Процес підготовки навчальних даних може включати:

- 1) фільтрацію даних, усунення помилок та шумів;
- 2) нормалізацію значень;
- 3) балансування набору даних;
- 4) виділення та формування інформативних ознак.

Зокрема для моделей які передбачають аналіз зображень, можуть бути застосовані наступні етапи підготовки:

- 1) зміна розміру кадру;
- 2) корекція яскравості та контрастності;
- 3) фільтрація шумів;
- 4) перетворення кольорових моделей.

Також важливим етапом підготовки даних є аугментація. Аугментація дозволяє штучно збільшити кількість навчальних даних шляхом виконання різних перетворень зображень, до яких можна віднести такі:

- 1) обертання зображення;
- 2) масштабування;
- 3) віддзеркалення;
- 4) зміна яскравості та додавання шумів;

Для забезпечення об'єктивної оцінки якості роботи моделі набори даних зазвичай розділяють на навчальну, тестову та валідаційну вибірки. Навчальна вибірка використовується безпосередньо для процесу навчання моделі; валідаційна - для контролю процесу навчання та підбору параметрів; тестова - для оцінки точності та ефективності роботи навченої моделі на нових вхідних даних.

Окрему увагу необхідно приділяти процесу маркування даних, оскільки правильність та точність позначення об'єктів або дефектів безпосередньо впливає на якість навчання ШНМ. Наявність помилок у розмітці може призводити до формування неправильних закономірностей та погіршення точності процесу класифікації.

Для забезпечення ефективного навчання, модель повинна отримати на вхід достатню кількість різноманітних прикладів для навчання, що будуть містити як нормальні об'єкти, так і приклади з дефектами та відхиленнями від норм.

Недостатня кількість вхідних даних, або нерівномірний розподіл класів може призвести до погіршення точності класифікації та перенавчання моделі. Окрім того, якісна підготовка даних дозволяє мінімізувати час навчання ШНМ, а також покращити її здатність до узагальнення результатів на нових даних.

Відповідно, етап підготовки вхідних даних навчальної та тестової вибірки є критично важливим для побудови ефективних моделей.

Для навчання системи та тестування алгоритму виявлення візуальних дефектів використовується набір даних Weakly Supervised Learning for Industrial Optical Inspection [16]. Даний датасет був створений з метою проведення досліджень оцінки алгоритмів автоматизованого оптичного контролю якості в промислових системах.

Особливістю даного датасету є те, що він містить у своєму складі набори зображень текстурованих поверхонь з наявністю або відсутністю дефектів. Дефекти представлені у вигляді локальних аномалій різної форми, розмірів та інтенсивності, що дозволяє використовувати цей набір даних для проведення досліджень у сфері задач класифікації, локалізації та сегментації дефектів.

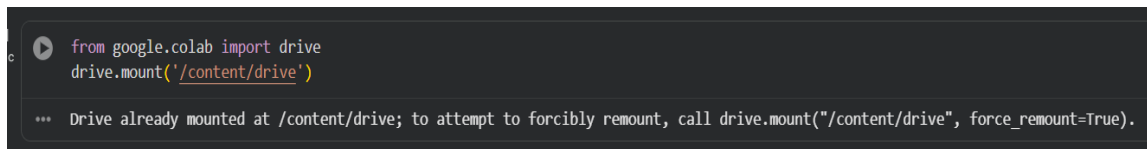
Датасет складається з набору класів, кожен з яких містить власний тип текстури поверхні та характерні дефекти. Для кожного класу сформовано набір зображень, що включають набори з дефектами та без дефектів. Дефектні зразки також мають маски розмітки, що дозволяє визначити область дефекту на поверхні.

Всі зображення представлені у відтінках сірого та збережені у форматі png. Для навчальних наборів кожного класу передбачено приблизно

1000 зображень без дефектів та 150 зображень з дефектами, тоді як тестові набори містить більшу кількість прикладів.

Варто зазначити, що даний датасет є штучно сформованим, але його структура та характер дефектів наближені до реальних задач промислового контролю якості.

Виконання роботи виконано з використанням середовища Google Colab та гугл диску. Для початку необхідно виконати підключення до гугл диску (див. рис. 4.1).

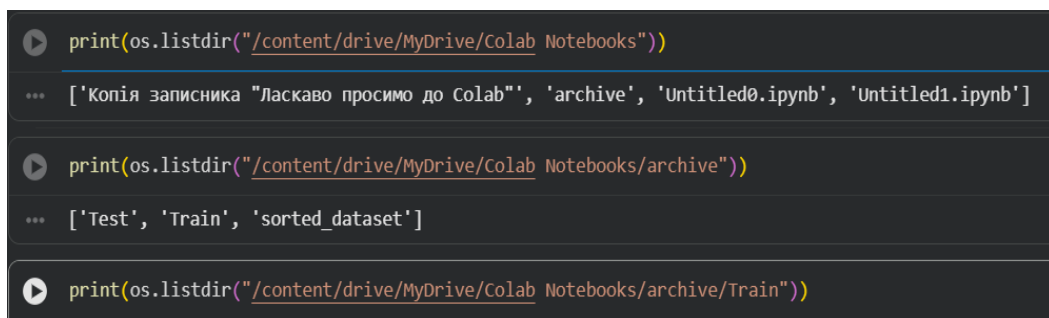


```
from google.colab import drive
drive.mount('/content/drive')

... Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Рисунок 4.1. Підключення гугл диску для отримання доступу до даних

Після вдалого підключення до гугл диску необхідно вказати шлях доступу до відповідних папок (див. рис. 4.2).



```
print(os.listdir("/content/drive/MyDrive/Colab Notebooks"))

... ['Копія записника "Ласкаво просимо до Colab"', 'archive', 'Untitled0.ipynb', 'Untitled1.ipynb']

print(os.listdir("/content/drive/MyDrive/Colab Notebooks/archive"))

... ['Test', 'Train', 'sorted_dataset']

print(os.listdir("/content/drive/MyDrive/Colab Notebooks/archive/Train"))
```

Рисунок 4.2. Вказання шляху доступу до папок з набором даних

Після налаштування доступу виконуємо підключення необхідних для попереднього сортування даних бібліотеки та завершуємо налаштовувати доступи до папок з набором даних (див. рис. 4.3).

```
import os
import shutil

# ROOT
DATA_ROOT = "/content/drive/MyDrive/Colab Notebooks/archive"

# SOURCE
IMAGES_DIR = os.path.join(DATA_ROOT, "Train")
LABEL_FILE = os.path.join(DATA_ROOT, "Train/Labels (1).txt")

# OUTPUT
OUTPUT_DIR = os.path.join(DATA_ROOT, "sorted_dataset")

OK_DIR = os.path.join(OUTPUT_DIR, "ok")
DEFECT_DIR = os.path.join(OUTPUT_DIR, "defect")

os.makedirs(OK_DIR, exist_ok=True)
os.makedirs(DEFECT_DIR, exist_ok=True)
```

Рисунок 4.3. Підключення бібліотек та завершення налаштування доступу до набору даних

Наступним кроком є безпосередньо процес сортування набору даних та розподіл між папками (див. рис. 4.4).

```
with open(LABEL_FILE, "r", encoding="utf-8") as f:
    lines = f.readlines()

    # пропускаємо перший рядок "1"
    lines = lines[1:]

    copied_ok = 0
    copied_defect = 0

    for line in lines:
        parts = line.strip().split()

        # перевірка
        if len(parts) < 3:
            continue

        defect_flag = parts[1]
        image_name = parts[2]

        source_path = os.path.join(IMAGES_DIR, image_name)

        # якщо файлу немає
        if not os.path.exists(source_path):
            print(f"NOT FOUND: {source_path}")
            continue

        # NORMAL
        if defect_flag == "0":
            destination = os.path.join(OK_DIR, image_name)
            shutil.copy(source_path, destination)

            copied_ok += 1

        # DEFECT
        else:
            destination = os.path.join(DEFECT_DIR, image_name)
            shutil.copy(source_path, destination)

            copied_defect += 1
```

Рисунок 4.4. Алгоритм сортування набору даних

Після успішного завершення сортування даних можна виконати перевірку на успішність виконання, та визначити кількість зображень у відповідних папках (див. рис. 4.5.).

```
print("DONE!")
print("OK IMAGES:", copied_ok)
print("DEFECT IMAGES:", copied_defect)

DONE!
OK IMAGES: 1000
DEFECT IMAGES: 150
```

Рисунок 4.5. Відображення результату сортування

Аналогічний процес сортування виконується і для набору тестових даних. Після успішного завершення розподілу та сортування можна переходити до процесу навчання нейронної мережі.

4.3. Опис алгоритму та програмних блоків

Перед початком роботи з побудови структури моделі ШНМ необхідно додати перелік необхідних бібліотек. Нижче продемонстровано їх перелік та спосіб підключення (див. рис. 4.6).

```
import torch
import torch.nn as nn

from torchvision import datasets, transforms
from torch.utils.data import DataLoader

import timm

import os
```

Рисунок 4.6. Підключення ключових бібліотек для навчання ШНМ

Даний перелік бібліотек містить основний необхідний функціонал, що дозволить виконати налаштування параметрів моделі та провести навчання.

Наступним кроком виконуємо вказання шляхів до необхідних наборів даних, що були попередньо посортовані, зазначаємо початкові параметри та визначаємо цільовий пристрій для навчання (див. рис. 4.7).

```
# =====  
# ШЛЯХИ  
# =====  
  
TRAIN_DIR = "/content/drive/MyDrive/Colab Notebooks/archive/sorted_dataset"  
TEST_DIR = "/content/drive/MyDrive/Colab Notebooks/archive/sorted_dataset_test"  
  
# =====  
# НАЛАШТУВАННЯ  
# =====  
  
IMAGE_SIZE = 224  
  
BATCH_SIZE = 32  
  
EPOCHS = 10  
  
LEARNING_RATE = 0.0001  
  
# =====  
# ВИЗНАЧЕННЯ ПРИСТРОЮ  
# =====  
  
DEVICE = "cuda" if torch.cuda.is_available() else "cpu"  
  
print("DEVICE:", DEVICE)
```

Рисунок 4.7. Виконання початкових налаштувань шляхів та параметрів ПНМ

Наступним кроком виконуємо початкові трансформації до зображень, а саме: масштабуємо кожне зображення до розмірів 224 на 224 пікселі; іноді виконуємо віддзеркалення з метою аугментації; випадково виконуємо обертання зображення, також з метою аугментації; та приводимо

набір даних до відповідного формату, з яким буде працювати модель (див. рис. 4.8).

```
from torchvision import transforms

train_transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomRotation(10),
    transforms.ToTensor(),
])

test_transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
])
```

Рисунок 4.8. Виконання попередньої обробки та перетворень зображень

Наступним кроком створюємо набори датасетів для тренування мережі та тестування (див. рис. 4.9).

```
train_dataset = datasets.ImageFolder(
    TRAIN_DIR,
    transform=train_transform
)

test_dataset = datasets.ImageFolder(
    TEST_DIR,
    transform=test_transform
)
```

Рисунок 4.9. Створення навчального та тестового датасетів

Наступним кроком формуємо DataLoader для тренувального та тестового датасетів (див. рис. 4.10).

```
from torch.utils.data import DataLoader

train_loader = DataLoader(
    train_dataset,
    batch_size=32,
    shuffle=True
)

test_loader = DataLoader(
    test_dataset,
    batch_size=32,
    shuffle=False
)
```

Рисунок 4.10. Створення DataLoader для навчального та тестового датасетів

Цей крок є необхідним, так як виконує розбиття набору даних на батчі - невеликі групи зображень, на яких виконується навчання та тестування. А також виконується випадкове перемішування.

У підсумку можна зазначити, що цей крок є необхідним для того, щоб навчання штучної нейронної мережі проходило більш ефективно, забезпечувало правильну аугментацію та стабільну оцінку на тестових даних.

Наступним кроком виконується створення та завантаження моделі для навчання (див. рис. 4.11).

```
model = timm.create_model(
    "efficientnet_b0",
    pretrained=True,
    num_classes=2
)

model = model.to(DEVICE)
```

Рисунок 4.11. Створення та завантаження моделі

Визначаємо функцію втрат (див. рис. 4.12). Дана функція втрат є звичайною для моделей, що працюють з задачами класифікації, і вона необхідна для того, щоб оцінювати якість передбачення.

```
criterion = nn.CrossEntropyLoss()
```

Рисунок 4.12. Визначення функції втрат

Наступним кроком є визначення оптимізатора (див. рис. 4.13).

```
optimizer = torch.optim.Adam(  
    model.parameters(),  
    lr=LEARNING_RATE  
)
```

Рисунок 4.13. Визначення оптимізатора

Оптимізатор представляє собою механізм, основною функцією якого є - оновлення ваг моделі під час навчання з метою зменшення помилки, яку визначає функція втрат.

Наступним кроком описуємо повний цикл алгоритму процесу навчання. Частина алгоритму показано нижче (4.14).

Даний алгоритм описує тренувальний цикл, який необхідний для того, щоб модель мережі поступово навчалася робити передбачення, обчислювала помилку яка виникла у процесі навчання, оновлювала ваги через оптимізатор та перевіряла отриману точність на тестових даних і при цьому зберігала ті параметри моделі, які забезпечили найкращий результат у порівнянні з попередніми.

```
best_accuracy = 0

print("Training started...\n")

for epoch in range(EPOCHS):

    # =====
    # TRAIN
    # =====

    model.train()

    running_loss = 0

    for images, labels in train_loader:

        images = images.to(DEVICE)
        labels = labels.to(DEVICE)

        # FORWARD

        outputs = model(images)

        loss = criterion(outputs, labels)

        # BACKPROP

        optimizer.zero_grad()

        loss.backward()

        optimizer.step()

        running_loss += loss.item()

    # =====
    # TEST
    # =====

    model.eval()

    correct = 0
    total = 0

    with torch.no_grad():

        for images, labels in test_loader:

            images = images.to(DEVICE)
            labels = labels.to(DEVICE)

            outputs = model(images)

            _, predicted = torch.max(outputs, 1)

            total += labels.size(0)

            correct += (predicted == labels).sum().item()

    accuracy = correct / total
```

Рисунок 4.14. Основний алгоритм процесу навчання ШНМ

Після завершення усіх епох процесу навчання буде отримано остаточну версію моделі з оптимальними параметрами, на якій можна буде проводити практичне застосування для класифікації.

4.4. Аналіз результатів роботи алгоритму

У процесі навчання моделі на кожній епосі формується результат навчання, що містить дані про номер епохи, значення втрат та отриману точність (див. рис. 4.15).

```
... Training started...

Epoch 1/10
Loss: 38.5502
Accuracy: 0.8713

Best model saved!

Epoch 2/10
Loss: 16.7951
Accuracy: 0.9113

Best model saved!

Epoch 3/10
Loss: 8.3193
Accuracy: 0.9487

Best model saved!

Epoch 4/10
Loss: 4.9760
Accuracy: 0.9687

Best model saved!

Epoch 5/10
Loss: 4.2513
Accuracy: 0.9713

Best model saved!
```

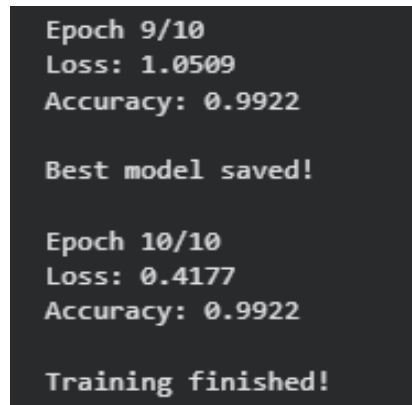
Рисунок 4.15. Виведення результатів процесу навчання ШНМ

Дані результати навчання відображають стабільний процес навчання зі зменшенням помилки. На початку помилка досить велика, а точність відносно низька, але з кожним кроком навчання оптимізатор виконує корекцію параметрів, що впливає на поступове зменшення показника втрат.

Одночасно з цим зростає показник точності, так як моделі внаслідок корекції параметрів виконує передбачення більш точно. Коли показник

втрат зменшується до досить низьких результатів, а точність стає високою, це є ознакою, що модель навчилася добре відтворювати закономірності у даних.

У результаті навчання ШНМ було отримано наступний результат (див. рис. 4.16).



```
Epoch 9/10  
Loss: 1.0509  
Accuracy: 0.9922  
  
Best model saved!  
  
Epoch 10/10  
Loss: 0.4177  
Accuracy: 0.9922  
  
Training finished!
```

Рисунок 4.16. Остаточний результат навчання моделі ШНМ

Оцінивши отримані показники, можна зазначити, що помилка стала досить малою, а точність класифікації становить близько 99%. Але важливішим параметром оцінки результативності моделі є практичне застосування на тестових даних.

Наступним кроком виконуємо тестування на вибірці нормальних та дефектних зображень. Для цього необхідно використати додатковий алгоритм, який буде випадковим чином обирати з тестового набору зображення по 3 кожного типу (див. рис. 4.17).

В результаті було отримано наступний результат класифікації на тестових даних (див. рис. 4.18)

```
import random
import matplotlib.pyplot as plt
from PIL import Image

classes = train_dataset.classes

def predict_image(img_path):
    image = Image.open(img_path).convert("RGB")
    image_tensor = test_transform(image).unsqueeze(0).to(DEVICE)

    model.eval()
    with torch.no_grad():
        output = model(image_tensor)
        probs = torch.softmax(output, dim=1)
        pred = torch.argmax(probs, dim=1).item()
        conf = probs[0][pred].item()

    return classes[pred], conf

def show_samples(folder, class_name, n=3):
    path = os.path.join(folder, class_name)
    files = [f for f in os.listdir(path) if f.lower().endswith(".png")]

    samples = random.sample(files, n)

    plt.figure(figsize=(15, 5))

    for i, file in enumerate(samples):
        img_path = os.path.join(path, file)

        pred, conf = predict_image(img_path)

        img = Image.open(img_path)

        plt.subplot(1, n, i+1)
        plt.imshow(img)
        plt.axis("off")

        plt.title(
            f"True: {class_name}\nPred: {pred}\nConf: {conf:.2f}"
        )

    plt.tight_layout()
    plt.show()
```

Рисунок 4.17. Алгоритм формування вибірки даних для практичних тестів моделі ШНМ

Можна зазначити, що навчена модель ШНМ здатна виявляти дефекти за зображенням забезпечуючи хорошу точність, у даному випадку було класифіковано нормальні та дефектні об'єкти зі 100% точністю.

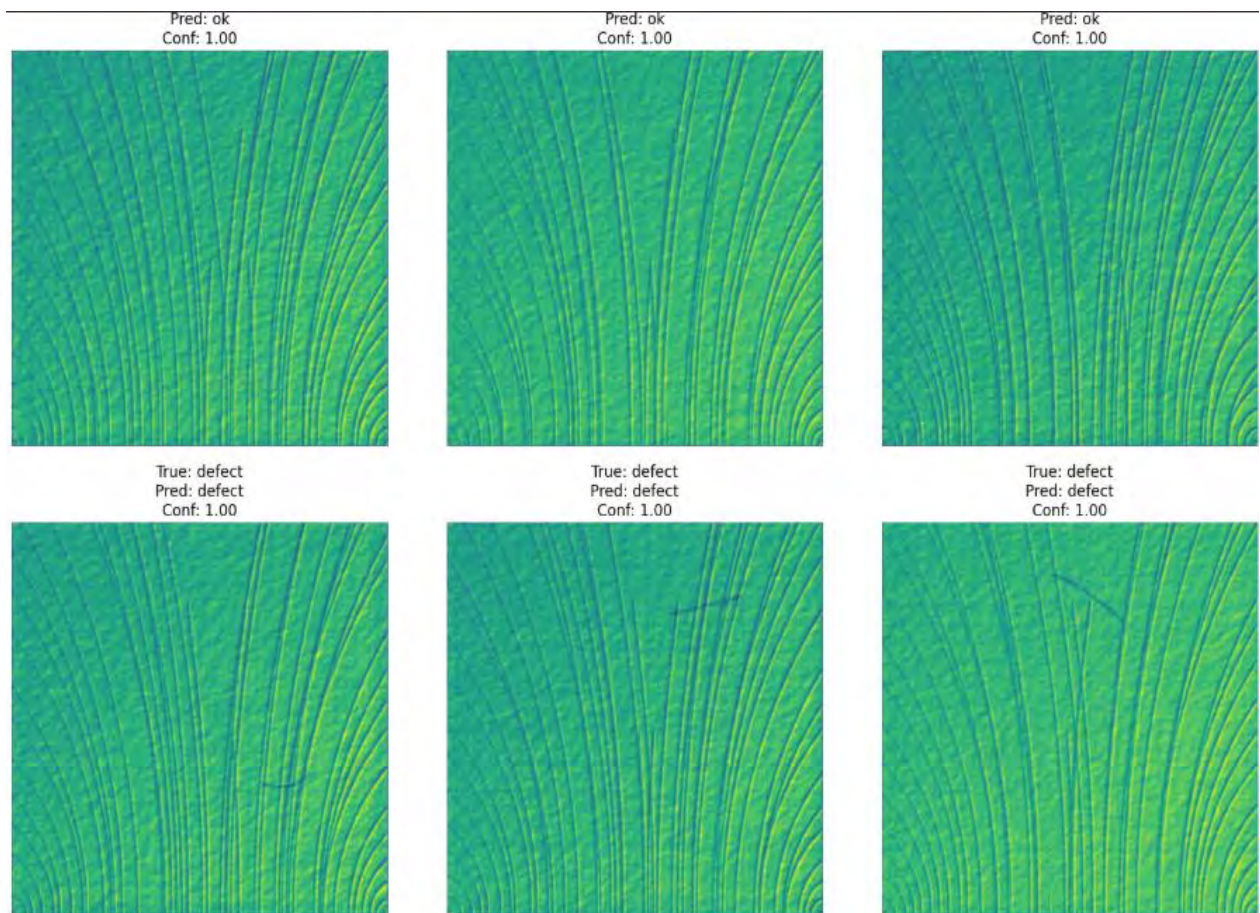


Рисунок 4.18. Результат практичного застосування навченої моделі

4.5. Блок-схема алгоритму роботи автоматизованої системи візуального контролю якості методами комп'ютерного зору

У даному підрозділі буде наведено та описано блок-схему алгоритму роботи автоматизованої системи візуального контролю якості методами комп'ютерного зору (див. рис. 4.19).

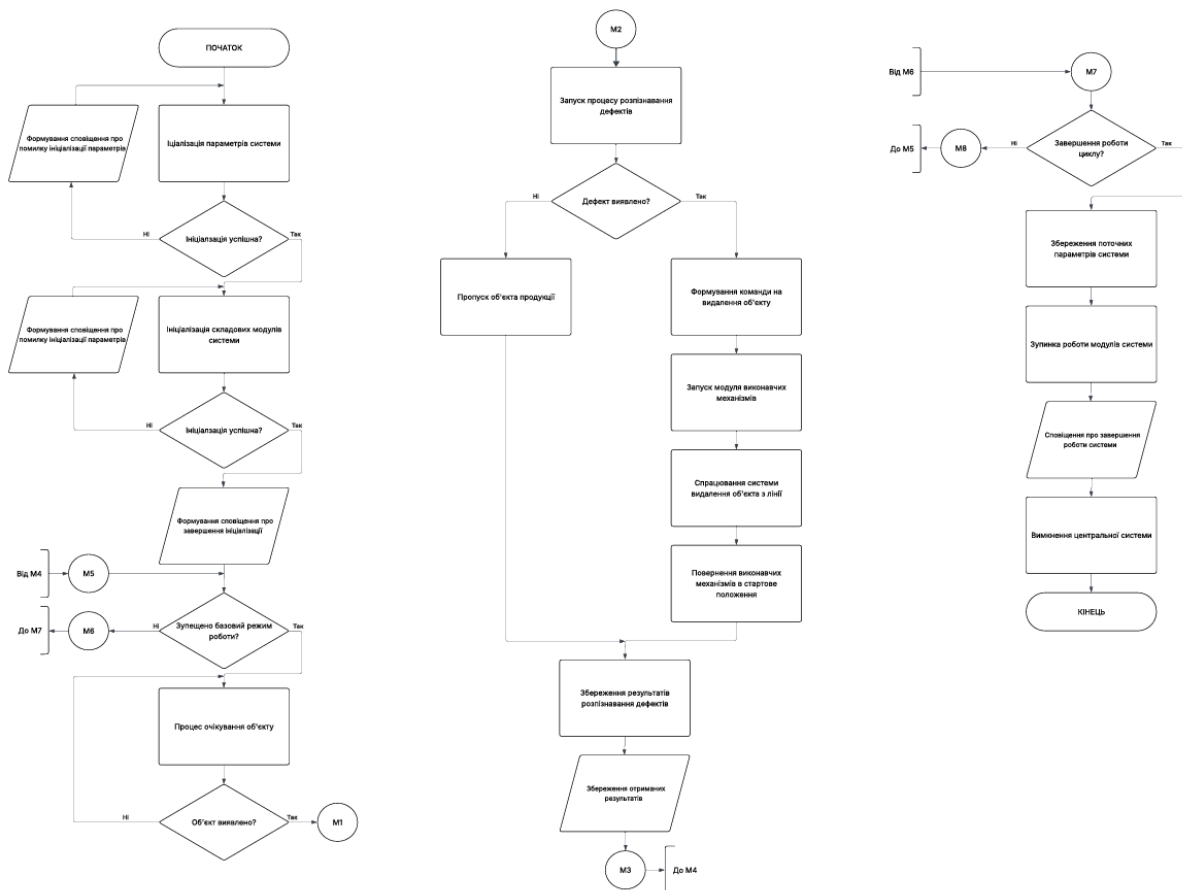


Рисунок 4.19. Блок-схема алгоритму роботи автоматизованої системи візуального контролю якості методами комп'ютерного зору

На початку алгоритму виконується базова ініціалізація. Ініціалізація - це процес попереднього налаштування параметрів роботи системи, до якого можна віднести налаштування вхідних даних, перевірка наявності та справності складових модулів, введення параметрів контролю, вибір режиму роботи тощо.

Ініціалізація є критично-важливим етапом алгоритму, так як його наявність гарантує що система розпочне свою роботу у справному вигляді.

Згідно алгоритму, якщо під час ініціалізації виникла помилка, то система сформує сповіщення про цю помилку і спробує повторно виконати ініціалізацію.

Після успішного завершення всіх етапів ініціалізації перевіряється умова запуску циклу роботи системи. Цикл роботи системи це умова за якої система циклічно виконує перелік операцій, у даному випадку це послідовність виявлення об'єкта у відповідній зоні, перевірка на наявність дефектів, формування висновку про відбракування чи ні, у разі виявлення браку фізичну усуває об'єкт, зберігає результати та повторює цей цикл знову до моменту доки він не буде перерваний чи повністю зупинений.

Якщо умова запуску циклу дійсна, то алгоритм переходить до етапу процесу виявлення об'єкта. Система повинна працювати тільки тоді, коли в зоні дії є об'єкт з яким можна працювати і відповідно у випадку коли такий об'єкт відсутній система буде очікувати його появу.

Після успішного виявлення об'єкта в робочій зоні запускається алгоритм візуальної оцінки та виявлення дефектів.

У разі якщо на об'єкті перевірки не виявлено жодних дефектів, то система пропускає його далі по лінії, але у випадку виявлення дефектів запускає ланцюг послідовних операцій, а саме: формується команда на видалення об'єкта; запускається модуль виконавчих механізмів; спрацьовує система фізичного видалення об'єкта з лінії, у даному випадку це сервоприводи; та виконує повернення виконавчих механізмів у вихідний стан для продовження роботи у звичному режимі.

У будь якому варіанті після завершення даного етапу відбувається збереження даних про результати циклу для формування статистики та перевірки якості роботи системи а також формується сповіщення про успішне збереження.

Після успішного завершення повного циклу, алгоритм повертається на етап перевірки умови запуску нового циклу і повторює вище описаний алгоритм.

У разі якщо умова запуску циклу не дійсна, алгоритм переходить від мітки M1 до M2. Там виконується перевірка чи є запит на повне завершення роботи системи. Даний етап необхідний для того, щоб система не вимикалась самостійно у разі відсутності об'єктів на лінії, а вимикалась виключно згідно введеної команди.

У разі якщо дана команда сформована, то відбувається перехід до послідовності дій коректного завершення роботи системи, що включає у собі: збереження поточних параметрів роботи системи; зупинка ключових модулів; формування сповіщення про завершення роботи автоматизованої системи; та вимкнення центральної обчислювальної системи.

У випадку коли такої команди немає алгоритм переходить від мітки M5 до M3 та повторює вище описаний перелік дій.

4.6. Висновки до розділу 4

У розділі було розглянуто та описано декілька програмних алгоритмів, а саме: алгоритм сортування та попередньої обробки навчальних та тестових даних та алгоритм опису моделі штучної нейронної мережі для класифікації зображень.

Попередня обробка даних, які будуть використовуватися для навчання моделі ШНМ є обов'язковим процесом, оскільки наявність даного етапу має безпосередній вплив на швидкість та якість навчання, а також на вихідну точність передбачення вже навченої моделі.

Також було розглянуто архітектуру моделі ШНМ, алгоритм навчання, а також аналіз отриманих результатів в процесі навчання та проведення тестування на нових вхідних даних.

Алгоритм навчання включав визначення функції втрат, вибір оптимізатора, налаштування параметрів. Під час навчання модель поступово мінімізовувала похибку на навчальній вибірці, адаптуючи вагові коефіцієнти.

Після завершення процесу навчання було проведено аналіз отриманих показників, а саме точність передбачення та показник помилки. В результаті навчання точність передбачення досягла 99,22%, зокрема, значення помилки становить менше ніж 0,5.

Була проведена перевірка навченої моделі на нових вхідних даних, задля підтвердження результатів навчання. В результаті було отримано повну відповідність між фактичним значенням та передбаченим, що в свою чергу підтверджує отримані показники в результаті навчання.

Побудовано загальну блок-схему алгоритму роботи автоматизованої системи виявлення візуальних дефектів з використанням методів комп'ютерного зору, а також наведено детальний опис кожного етапу у послідовності виконання.

ВИСНОВКИ

У ході роботи було досліджено та реалізовано підхід до автоматизації процесу виявлення дефектів використовуючи алгоритми машинного навчання та комп'ютерного зору. На основі аналізу сучасних методів та підходів до автоматизації було визначено, що використання моделей ШНМ є досить ефективним та перспективним інструментом для підвищення точності, швидкості та стабільності автоматизованих систем контролю якості продукції, зокрема і на виробничих лініях. Проведена робота включає перелік наступних розділів, у яких досліджено ключові аспекти теми:

1) розгляд загальних аспектів автоматизації виробничих процесів, де було описано загальні принципи автоматизації та застосування цих принципів для різних технічних задач. Описано загальну концепцію застосування методів та алгоритмів машинного навчання та комп'ютерного зору, що є перспективним напрямом в сучасній автоматизації. Зокрема було розглянуто перелік аналогів автоматизованих систем контролю якості продукції за допомогою автоматизації візуальної оцінки стану об'єктів;

2) описано загальну структуру автоматизованої системи виявлення дефектів, що включає у собі різноманітні модулі. Описано концепцію роботи автоматизованої системи з використанням алгоритмів комп'ютерного зору. Детально розглянуто та описано особливості розробки алгоритмів комп'ютерного зору, а також програмні засоби, які можна використовувати для реалізації даної задачі;

3) виконано аналіз апаратних складових системи з описом технічним параметрів, з метою підбору найоптимальніших складових для побудови системи. Даний розділ включає детальний аналіз усіх ключових модулів, описано переваги та недоліки кожного з обґрунтуванням їх використання;

4) реалізовано алгоритм аналізу візуальних даних для оцінки стану поверхні досліджуваних об'єктів. Для виконання цієї задачі було виконано підготовку та обробку вхідних даних для моделі ШНМ, а також розроблено

алгоритм навчання моделі. Проведено навчання моделі, тестування та аналіз отриманих результатів. Було отримано показник точності розпізнавання дефектів 99,22%, що було підтверджено практичною перевіркою на нових вхідних даних.

Загалом, розроблена система поєднує сучасні методи машинного навчання, комп'ютерного зору та апаратної інтеграції, при цьому забезпечуючи надійний, швидкий та масштабований інструмент для вирішення задачі автоматизованого візуального аналізу якості поверхні продукції.

Отримані результати підтверджують доцільність впровадження такого роду систем у виробничі процеси з метою підвищення вихідної якості продукції та мінімізації помилок внаслідок впливу людського фактора.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ ТА ДЖЕРЕЛ

1. Я. І. Проць, О. А. Данилюк, Т.Б. Лобур Автоматизація неперервних технологічних процесів. Навчальний посібник для технічних спеціальностей вищих навчальних закладів. – Тернопіль: ТДТУ ім. І.Пулюя, 2008. – 239с.;
2. Інтелектуальні системи автоматизації : монографія / Аврунін О. Г., Владов С. І., Петченко М. В., Семенець В. В., Татарінов В. В., Тельнова Г. В., Філатов В. О., Шмельов Ю. М., Шушляпіна Н. О. – Кременчук : Видавництво «НОВАБУК», 2021. – 322 с.;
3. Основи машинного навчання : навч. посіб. / В. О. Харченко. – Суми : Сумський державний університет, 2023. – 264 с.;
4. Пуйда В. Я. Система технічного зору для досліджень в області дефектоскопії матеріалів та виробів // Computer Systems and Networks. – 2022. - Vol. 4, No. 1. - С. 12-128. - DOI: <https://doi.org/10.23939/csn2022.01.122>;
5. Олейніков І. А., Срібна І. М. Назва статті // Телекомунікаційні та інформаційні технології. - 2025. - № 1(86). - С. 111–119. – DOI: 10.31673/2412-4338.2025.014545;
6. Творошенко І. С. Конспект лекцій з дисципліни «Цифрова обробка зображень» для студентів 4 курсу денної форми навчання напряму 6.080101 – Геодезія, картографія та землеустрій / І. С. Творошенко ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2017. – 75 с.;
7. Цифрова обробка сигналів та зображень. [Електронний ресурс]: навч. посіб. для здобувачів ступеня магістра за освіт. програмою «Комп'ютерно-інтегровані системи та технології в приладобудуванні» спец. 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка / КПІ ім. Ігоря Сікорського ; уклад.: В. І. Микитенко, Г. С. Тимчик. Електрон. текст. дані (1 файл). – Київ : КПІ ім. Ігоря Сікорського, 2025. – 181 с.;

8. Мікрокомп'ютер Raspberry Pi 5 Board 8GB. [Електронний ресурс] режим доступу: <https://evo.net.ua/mikrokompiuter-raspberry-pi-5-board-8gb/>;
9. Мікрокомп'ютер Raspberry Pi Zero W. [Електронний ресурс] режим доступу: <https://evo.net.ua/raspberry-pi-zero-w/>;
10. Камера Raspberry Pi Camera Module 3 Wide NoIR [Електронний ресурс] режим доступу: <https://evo.net.ua/kamera-raspberry-pi-camera-module-3-wide-noir/>;
11. Платформа розробника Arduino Nano [Електронний ресурс] режим доступу: https://evo.net.ua/arduino-nano/?srsltid=AfmBOoodUBdK6Ctf3dBJHh-owSVQbb-Ay0qIFS2jAgSTwkBNWYBK1x4_;
12. 16-ти канальний 12-ти бітний ШІМ модуль для серводвигунів [Електронний ресурс] режим доступу: <https://arduino.ua/prod1442-16-kanalnii-12-bit-pwmservo-modyl-s-i2c-interfeisom-na-pca9685>;
13. Сервопривід MG90S. [Електронний ресурс] режим доступу: <https://arduino.ua/prod229-servoprivod-mg-90s-micro>;
14. LCD-дисплей 20x4. [Електронний ресурс] режим доступу: <https://arduino.ua/prod663-lcd-2004-i2c-simvolnii-displei-20x4-sinii>;
15. 16-ти кнопкова мембранна клавіатура. [Електронний ресурс] режим доступу: <https://arduino.ua/prod167-16-ti-knopochnaya-membrannaya-klaviatyr>;
16. M. Wieler, T. Hahn, and F. A. Hamprecht, “Weakly Supervised Learning for Industrial Optical Inspection,” Heidelberg Collaboratory for Image Processing (HCI), Sep. 14, 2007. DOI: 10.5281/zenodo.12750201.
17. Спеціальні розділи математики: конспект лекцій : навч. посіб. / КПІ ім. Ігоря Сікорського ; уклад.: Ю. В. Куц, Ю. Ю. Лисенко, В. М. Сокурєнко. – Київ : КПІ ім. Ігоря Сікорського, 2024. – 192 с.
18. Муравйов О. В. Передача даних та сучасні методи обробки сигналів. Практикум: навчальний посібник / О. В. Муравйов; КПІ ім. Ігоря Сікорського. – Київ: КПІ ім.Ігоря Сікорського, 2022. – 55 с.

19. Статистичні методи визначення залежностей між випадковими величинами: навчальний посібник / Ю. В. Куц, Ю. Ю. Лисенко; КПІ ім. Ігоря Сікорського. – Київ: КПІ ім. Ігоря Сікорського, 2023. – 115 с.

20. Комп'ютерне проєктування електронних схем. Комп'ютерний практикум [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою «Комп'ютерно-інтегровані системи та технології в приладобудуванні» спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» та 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» / Р. М. Галаган ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 21,33 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2023. – 419 с.

21. Баженов В.Г. Електроніка. Лабораторний практикум: навчальний посібник / В. Г. Баженов, Є. Ф. Сулов, Ю. Ю. Лисенко, А.С. Момот; КПІ ім. Ігоря Сікорського. – Київ: КПІ ім. Ігоря Сікорського, 2022. – 70 с.

22. Куц Ю.В. Новітні системи та технології: навчальний посібник / Ю. В. Куц, Ю. Ю. Лисенко, А.С. Момот; КПІ ім. Ігоря Сікорського. – Київ: КПІ ім. Ігоря Сікорського, 2022. – 123 с.

23. Image Fusion as an Augmentation Method to Improve the Classification Efficiency of Thermal Images / D. Storozhyk, A. Protasov, I. Lysenko, Y. Steshenko, A. Alexiev // International Journal “NDT Days”, BSNDT: Bulgaria, 2025. – Volume 8, Issue 2. – 77-84 pp. ISSN: 2603-4018, eISSN: 2603-4646

24. Storozhyk D., A. Protasov, Y. Kuts, O. Muraviov, I. Lysenko, Y. Mirchev (2024) Enhancing neural network efficiency in automated image analysis for thermal nondestructive testing. Journal of Theoretical and Applied Mechanics, Vol.54, pp. 242-252.

25. Практичний досвід застосування та аналіз ефективності різних магнітних суспензій / В.С. Якотюк, С.М. Глабець, Ю.Ю. Лисенко, А.С. Момот // Технічна діагностика і неруйнівний контроль, 2025, №4, стор. 44-48.

26. Analysis of eddy current probe signal model for structural monitoring of aircraft materials / I. Lysenko, Y. Kuts, V. Uchanin, Y. Mirchev // Transactions on Aerospace Research, Institute of Aviation: Poland, 2025. –Volume 3/2025 (280). – 1-10 pp. eISSN 2545-2835, DOI: <https://doi.org/10.2478/tar-2025-0010>.

27. Studies of EDDY current probes for inspection of aluminium alloy structure welds using smartphone-based flaw detector / G. Mook, V. Uchanin, Ju. Lysenko // The Paton Welding Journal, 2024, 12, 2024, pp. 42–48.

28. Investigation of the Impact of Limited Penetration Depth on the Effectiveness of Eddy Current Testing of Aluminum Alloys / V. Uchanin, I. Lysenko, Y. Kuts, Y. Mirchev // International Journal “NDT Days”, BSNDT: Bulgaria, 2024. – Volume 7, Issue 4. – 213-217 pp. ISSN: 2603-4018, eISSN: 2603-4646

29. Effectiveness of the technology of automated eddy current flaw detection with array probe / Iu. Lysenko, Yu. Kuts, Y. Mirchev, O.E. Levchenko, S.M. Glabets // The Paton Welding Journal, 2024, 11, 2024, pp. 30–35. <https://doi.org/10.37434/tpwj2024.11.04>

30. Ефективність технології автоматизованої вихрострумової дефектоскопії з матричними перетворювачами / Лисенко Ю.Ю., Куц Ю.В., Мірчев Й., Левченко О.Е., Глабець С.М. // Технічна діагностика і неруйнівний контроль, 2024, №3, стор. 3-8.

31. Дослідження вихрострумових перетворювачів для контролю зварних швів конструкцій із алюмінієвих сплавів з використанням дефектоскопа на базі смартфона / Мок Г., Учанін В.М., Лисенко Ю.Ю. // Технічна діагностика і неруйнівний контроль, 2024, №3, стор. 32-38.

32. Evaluation Of Eddy Current Array Performance In Detecting Aircraft Component Defects / I. Lysenko, Y. Kuts, V. Uchanin, Y. Mirchev, O. Levchenko // Transactions on Aerospace Research, Institute of Aviation: Poland, 2024. – Volume 2/2024 (275). – 1-9 pp. eISSN 2545-2835, DOI: <https://doi.org/10.2478/tar-2024-0007>