

## КВАЛІФІКАЦІЙНА РОБОТА

Кіберфізична система збору, аналізу якості борошна на підставі машинного

навчання

Назва теми

Рівень вищої освіти другий (магістерський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Шифр КВРКІ 240111.24.01.10 ПЗ

Виконав здобувач II курсу, група КІ2М-24-1

Керівник

канд.-техн. наук, доцент  
Науковий ступінь, учене звання

Нормоконтролер

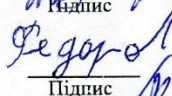
д. техн. наук, професор  
Науковий ступінь, учене звання

До захисту допускаю:  
завідувач кафедри КІС  
« 5 » травня 2026 р.

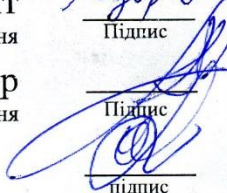
дата

  
Підпис

Артем Гуцалюк  
Ініціали, прізвище

  
Підпис

Євген Федоров  
Ініціали, прізвище

  
Підпис

Сергій ЛИСЕНКО  
Ініціали, прізвище

Ольга ПАВЛОВА  
Ініціали, прізвище

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ДРУГИЙ (МАГІСТЕРСЬКИЙ)

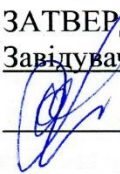
Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КІС

 Ольга ПАВЛОВА

“ 12 ” 01 2026 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Гуцалюк Артем Олексійович

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система збору, аналізу якості борошна на підставі машинного навчання

Керівник проекту (роботи) Федоров Євген Євгенович, д.т.н., проф

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 12.01.2026 р. № 6

2. Термін подання здобувачем роботи на кафедру 01.05.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

Аналіз предметної області та вимог до кіберфізичної системи контролю якості борошна

Розробка методу комплексного оцінювання якості борошна на основі машинного навчання

Моделювання, програмна реалізація та аналіз ефективності кіберфізичної системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) \_\_\_\_\_

Архітектура ПЗ проекту

Архітектура ПЗ для кіберфізичної системи

Апаратне забезпечення проекту

6. Консультанти розділів кваліфікаційної роботи

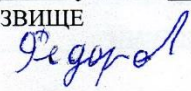
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання « 12 » 01 2026 р.

**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики КвРМ з керівником	12.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	12.01.2026	виконано
3	Робота над розділом 1 - аналіз відомих моделей, методів за темою; постановка задачі	20.01.2026	виконано
4	Робота над розділом 2 - розробка моделей для вирішення поставленої задачі	01.02.2026	виконано
5	Робота над науковою статтею	01.03.2026	виконано
6	Робота над розділом 3 - розробка методів для вирішення поставленої задачі	15.03.2026	виконано
7	Робота над розділом 4 - проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.2026	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2026	виконано
9	Попередній захист ДРМ	29.04.2026	виконано
10	Захист ДРМ на засіданні ЕК	До 15.05.2026	

Здобувач  Артем Гуцалюк  
Підпис Імя, ПРИЗВИЩЕ

Керівник кваліфікаційної роботи  Євген Федоров  
Підпис Імя, ПРИЗВИЩЕ

## РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Кіберфізична система збору, аналізу якості борошна на підставі машинного

Автор роботи: Гуцалюк Артем Олексійович

Керівник роботи: Федоров Євген Євгенійович

Пояснювальна записка: 81 с., 17 рис., 8 табл., 2 дод., 81 джерел.

КІБЕРФІЗИЧНА СИСТЕМА, ЯКІСТЬ БОРОШНА, МАШИННЕ НАВЧАННЯ, КОМП'ЮТЕРНИЙ ЗІР, НЕЙРОННІ МЕРЕЖІ, ВИЯВЛЕННЯ АНОМАЛІЙ, ОБРОБКА ЗОБРАЖЕНЬ, КОНТРОЛЬ ЯКОСТІ

Об'єктом дослідження є процес автоматизованого контролю показників якості борошна.

Предметом дослідження є метод комплексного оцінювання якості борошна, моделі машинного навчання та апаратно-програмні засоби кіберфізичної системи для збору й аналізу даних.

Метою кваліфікаційної роботи магістра є підвищення ефективності контролю якості борошна шляхом розробки кіберфізичної системи, для забезпечує автоматизований збір та аналіз візуальних і технологічних даних із використанням методів машинного навчання.

Для розв'язання поставлених задач використовувалися методи машинного навчання, комп'ютерного зору, математичного моделювання та методи побудови кіберфізичних систем.

Наукова новизна отриманих результатів полягає в тому, що:

- набув подальшого розвитку метод комплексного оцінювання якості борошна шляхом поєднання візуальних даних і параметрів технологічного процесу в межах єдиної ML-моделі;
- набула подальшого розвитку інформаційна технологія автоматизованого збору, комплексування та інтелектуального аналізу даних для контролю якості борошна в режимі реального часу.

На основі проведених досліджень розроблено архітектуру кіберфізичної системи та компоненти програмного забезпечення для автоматизованого збору, обробки, аналізу й логування даних, що забезпечують функціонування системи в режимі реального часу.

Практична значимість отриманих результатів полягає у можливості створення доступної кіберфізичної системи експрес-аналізу якості борошна, яка дозволяє зменшити вплив людського фактора, підвищити оперативність контролю та забезпечити безперервний моніторинг якості продукції.

У першому розділі проведено аналіз сучасного стану автоматизації контролю якості у борошномельній промисловості. Розглянуто підходи до збору та обробки даних, включаючи методи машинного навчання та виявлення аномалій. Встановлено обмеження традиційних лабораторних методів і складність інтеграції існуючих рішень, що обґрунтовує доцільність використання гібридної архітектури в межах кіберфізичних систем.

У другому розділі сформовано теоретичну основу дослідження шляхом побудови математичних і концептуальних моделей. Запропоновано модель представлення вхідних даних, що інтегрує візуальні та технологічні параметри, а також моделі класифікації, виявлення аномалій і часових обмежень, що забезпечують роботу системи в режимі реального часу.

У третьому розділі розроблено метод комплексного оцінювання якості борошна на основі машинного навчання. Метод базується на інтеграції двох обчислювальних трактів — класифікації та виявлення аномалій. Запропоновано підхід до комплексування даних, механізм прийняття рішення, а також реалізовано логування та адаптацію системи до змін умов експлуатації.

У четвертому розділі виконано програмно-апаратну реалізацію системи та проведено її тестування. Отримані результати підтвердили точність у стабільних умовах і виявили вплив зовнішніх факторів на якість роботи. Показано, що використання механізмів адаптації дозволяє відновлювати ефективність системи в змінних умовах.

## ЗМІСТ

Скорочення та умовні позначки .....	5
Вступ.....	6
1 Аналіз відомих моделей, методів та засобів автоматизованого збору та аналізу якості на основі машинного навчання .....	8
1.1 Аналіз моделей машинного навчання, що застосовуються в задачах автоматизованого аналізу якості .....	8
1.2 Аналіз методів автоматизованого контролю та збору даних якості з використанням машинного навчання.....	13
1.3 Аналіз апаратних та програмних засобів реалізації підсистеми збору даних та аналізу.....	19
1.4 Аналіз українських наукових та прикладних ініціатив у сфері автоматизації контролю якості.....	25
1.5 Постановка задачі дослідження.....	27
1.6 Висновки .....	28
2 Моделювання процесів функціонування кіберфізичної системи контролю якості борошна.....	30
2.1 Концептуальна модель та формалізовані критерії адекватності системи.....	30
2.2 Математична модель задачі ідентифікації якості борошна.....	32
2.3 Модель простору ознак і вхідних даних .....	35
2.4 Модель нейромережевої класифікації .....	38
2.5 Модель виявлення аномалій .....	41
2.6 Модель часових обмежень і критерії адекватності .....	43
2.7 Висновки .....	47
3 Метод оцінювання якості борошна на основі машинного навчання.....	48
3.1 Основи методу комплексного оцінювання якості борошна.....	48
3.2 Представлення та комплексування вхідних даних .....	51
3.3 Оцінювання якості борошна .....	53
3.4 Механізм прийняття рішення щодо якості продукту .....	56

3.5 Механізм логування та адаптації даних у системі контролю якості .....	59
3.6 Теоретичне обґрунтування та аналіз ефективності методу .....	61
3.7 Висновки .....	63
4 Програмно-апаратна реалізація та експериментальне тестування розробленої системи .....	64
4.1 Програмно-апаратна реалізація та архітектурна інтеграція компонентів.....	64
4.2. Опис функціонування системи та сценарії роботи кіберфізичної системи .....	67
4.3. Технічне планування та підготовка апаратно-програмного комплексу до експерименту .....	69
4.4 Підготовка та навчання нейромережових моделей .....	70
4.5. Проведення експериментальних досліджень системи в динамічних умовах..	72
4.6 Висновок .....	82
Висновки .....	84
Перелік джерел посилань .....	86
ДОДАТОК А .....	96
ДОДАТОК Б .....	97
ДОДАТОК В .....	100
ДОДАТОК Г .....	107

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ЗНМ - Згорткова нейронна мережа

КФС - кіберфізична система

МН - машинне навчання

КЗ - комп'ютерний зір

## ВСТУП

Сучасний стан галузі переробки зерна та виробництва борошна характеризується значною залежністю від лабораторних методів контролю, які, попри свою нормативну необхідність, мають суттєві експлуатаційні обмеження. Аналіз показує, що традиційні підходи до оцінювання якості базуються на періодичному ручному відборі проб та їх подальшому аналізі, що робить процес дискретним і розтягнутим у часі. Така затримка між моментом виробництва та отриманням результату ускладнює оперативне реагування на зміни властивостей сировини, зумовлюючи потребу у впровадженні локальних кіберфізичних засобів для безперервного моніторингу якості продукції безпосередньо в потоці.

Проблема дослідження полягає у тому, що наявні на ринку промислові аналізатори є високовартісними та складними для інтеграції, а також спостерігається дефіцит доступних рішень, які б поєднували бюджетні IoT-сенсори з потужними алгоритмами глибокого навчання.

Актуальність теми роботи зумовлена важливістю забезпечення стабільної якості борошна для конкурентоспроможності української продукції. Розробка адаптивної системи дозволить мінімізувати вплив людського фактора та забезпечити безперервний моніторинг, що робить обраний напрям дослідження своєчасним.

Метою роботи є підвищення ефективності контролю якості борошна шляхом розробки КФС, що забезпечує автоматизований збір та аналіз візуальних і технологічних даних із використанням методів МН.

Поставлена мета досягається розв'язанням таких основних завдань:

- аналіз сучасних методів і засобів автоматизованого контролю якості борошна;
- обґрунтування структури кіберфізичної системи та вибір моделей МН;
- розробка методу комплексного оцінювання якості борошна;
- розробка підсистеми збору даних та формування вхідної інформації;
- дослідження ефективності розробленого методу та системи.

Об'єктом дослідження виступає процес автоматизованого контролю показників якості борошна.

Предметом дослідження є метод, моделі МН та апаратно-програмні засоби кіберфізичної системи для збору та аналізу даних.

Наукова новизна отриманих результатів полягає в удосконаленні підходу до оцінювання якості борошна шляхом комплексування візуальних даних та параметрів технологічного процесу в межах єдиної ML-моделі.

На основі проведених досліджень розроблено метод оптимізації продуктивності систем інтелектуальних мереж з пристроями IoT, сервером з множинним доступом та БПЛА для додаткового зв'язку.

Практичне значення роботи визначається можливістю створення доступної системи експрес-аналізу.

Для розв'язання поставлених задач використовувалися методи МН, комп'ютерного зору, математичного моделювання та методи побудови кіберфізичних систем.

За темою кваліфікаційної роботи опубліковано дві публікації: збірник наукових праць за матеріалами XVII Міжнародної студентської науково-технічної конференції «Перспективні мережеві та комп'ютерні технології». (Харків- 2026). [82]; збірник наукових праць за матеріалами 3-ї міжнародної науково-практичної конференції «Глобальні тенденції в науці: дослідження, інновації та розвиток» (27–29 квітня 2026 р., Варна, Болгарія) [83]

# 1 АНАЛІЗ ВІДОМИХ МОДЕЛЕЙ, МЕТОДІВ ТА ЗАСОБІВ АВТОМАТИЗОВАНОГО ЗБОРУ ТА АНАЛІЗУ ЯКОСТІ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

1.1 Аналіз моделей машинного навчання, що застосовуються в задачах автоматизованого аналізу якості

Інтеграція методів штучного інтелекту в системи управління якістю харчової промисловості зумовлює перехід від детермінованих алгоритмів до ймовірнісних моделей, здатних адаптуватися до природної варіативності біологічної сировини [1][2]. У контексті КФС збору та аналізу якості борошна математична модель виступає центральним елементом обробки даних, який трансформує вхідні сигнали від сенсорів (оптичних, спектральних, хімічних) у кінцеве рішення про відповідність продукції стандартам. На відміну від жорсткої логіки, де правила задаються експертом вручну, моделі МН автоматично формують правила прийняття рішень шляхом оптимізації внутрішніх параметрів на основі навчальної вибірки. Такий підхід дозволяє виявляти складні нелінійні залежності між фізико-хімічними властивостями зерна та якісними показниками готового борошна, які неможливо описати аналітичними формулами [3].

Значна частина задач контролю якості базується на обробці структурованих даних, отриманих у результаті лабораторних вимірювань або зчитування показників технологічних датчиків (температура, вологість, білизна). Для таких умов, де простір ознак є чітко визначеним, найбільш ефективними залишаються класичні алгоритми навчання з учителем.

Одним із найпоширеніших методів є метод опорних векторів. Його математична сутність полягає у пошуку оптимальної гіперплощини, що максимізує відстань між класами даних у багатовимірному просторі ознак [4]. Завдяки використанню ядерних функцій, метод ефективно працює з лінійно нероздільними даними, трансформуючи їх у простір вищої розмірності[5]. Дослідження, підтверджують стійкість методу опорних векторів до перенавчання на малих вибірках, що є характерним для лабораторних наборів даних, де отримання

кожного зразка вимагає тривалого хімічного аналізу[6]. Водночас час навчання моделі суттєво зростає при роботі з великими масивами даних, що обмежує її використання у системах високошвидкісного моніторингу.

Альтернативним підходом є алгоритм k-найближчих сусідів. Цей метод не створює узагальнюючої математичної моделі, а класифікує нові зразки борошна шляхом пошуку аналогів у наявній базі даних [7]. Відсутність складного етапу навчання та простота програмної реалізації роблять даний метод привабливим для вбудованих обчислювальних систем. Проте точність методу критично залежить від обраної метрики відстані та наявності шумів у вхідних сигналах, що вимагає ретельної попередньої обробки даних.

Окремий клас моделей, що набув значного поширення для обробки табличних даних у промисловості, становлять ансамблеві методи. Їхня методологія базується на об'єднанні прогнозів великої кількості базових моделей, що дозволяє отримати результат, точність і стабільність якого перевищують показники будь-якого окремого алгоритму [8].

Найвідомішим представником цього класу є алгоритм Випадкового лісу (рисунок 1.1). Метод полягає у побудові множини незалежних дерев рішень, кожне з яких навчається на випадковій підвибірці даних. Під час класифікації зразка борошна фінальне рішення приймається шляхом голосування більшості дерев, а для задач регресії (прогнозування числових параметрів) результати усереднюються [9].

Архітектура ансамблю забезпечує високу стійкість до зашумлених даних, що надходять із сенсорів КФС. Помилка окремого датчика вплине лише на частину дерев, тоді як більшість ансамблю сформує коректний прогноз на основі інших ознак.

Важливою перевагою ансамблевих методів, зокрема градієнтного бустингу, є можливість кількісної оцінки важливості кожної вхідної ознаки. Це дозволяє інженерам визначити, які параметри технологічного процесу (температура, струм двигуна) мають найбільший вплив на якість борошна, та виключити з моніторингу малоінформативні показники [10, 11].

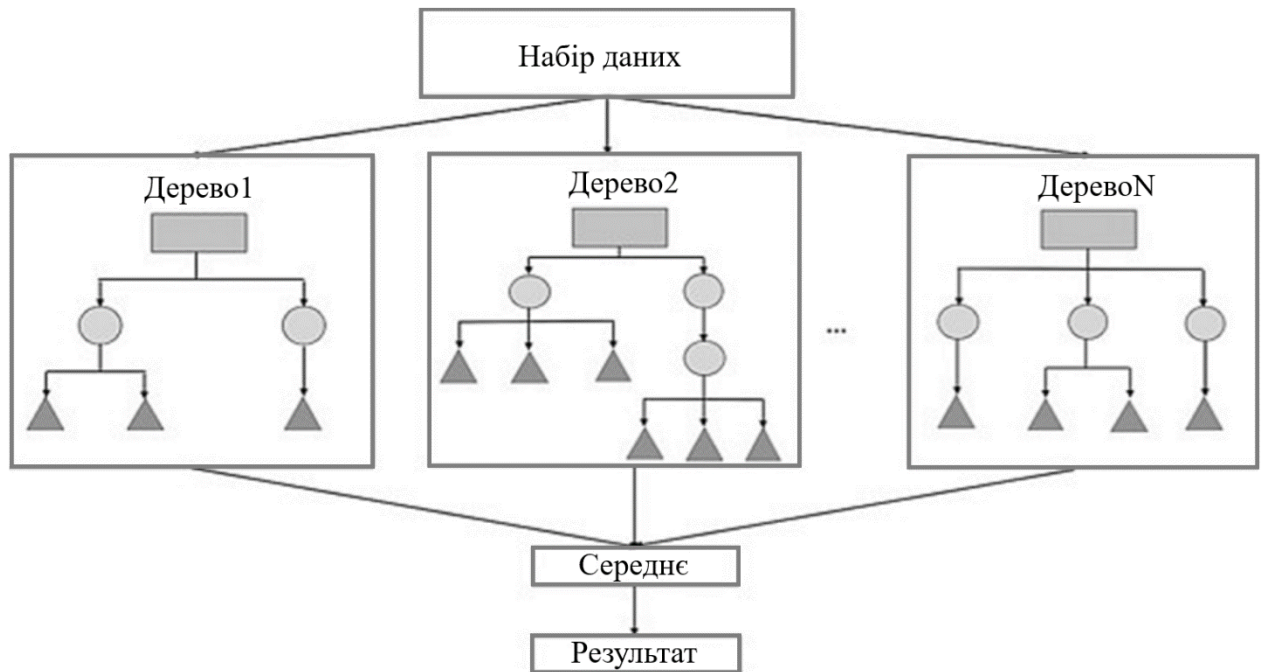


Рисунок 1.1 - Схематичне зображення принципу роботи методу випадкового лісу [75]

З розвитком технологій КЗ фокус досліджень змістився в бік глибокого навчання. На відміну від класичних алгоритмів, які вимагають попереднього ручного конструювання ознак, глибокі нейронні мережі здатні автоматично формувати ієрархічні представлення даних безпосередньо з первинного сигналу [12]. Це є важливим для аналізу зображень борошна, де інформація про якість міститься у складних текстурних візерунках, які важко формалізувати математично.

Основним інструментом у цій сфері є ЗНМ. Їх архітектура використовує операцію математичної згортки для виявлення локальних особливостей зображення. Перші шари мережі реагують на прості примітиви (лінії, точки), тоді як глибші шари розпізнають комплексні образи дефектів, наприклад, специфічну форму частинок висівок [13]. Сучасні архітектури (ResNet, EfficientNet) дозволяють будувати глибокі моделі з високою точністю класифікації, вирішуючи проблему згасання градієнта за допомогою залишкових зв'язків [14].

Для задач потокового контролю застосовують алгоритми об'єктної детекції (наприклад, сімейства YOLO). Ці моделі здатні в реальному часі локалізувати та ідентифікувати окремі об'єкти на конвеєрній стрічці, що дозволяє підраховувати

кількість сторонніх домішок на одиницю площі [15]. Також варто відзначити застосування рекурентних нейронних мереж (RNN/LSTM) для аналізу спектральних даних [16]. Враховуючи кореляційні зв'язки між сусідніми ділянками спектра, ці моделі підвищують точність визначення хімічного складу (білок, волога) порівняно з лінійними методами [17].

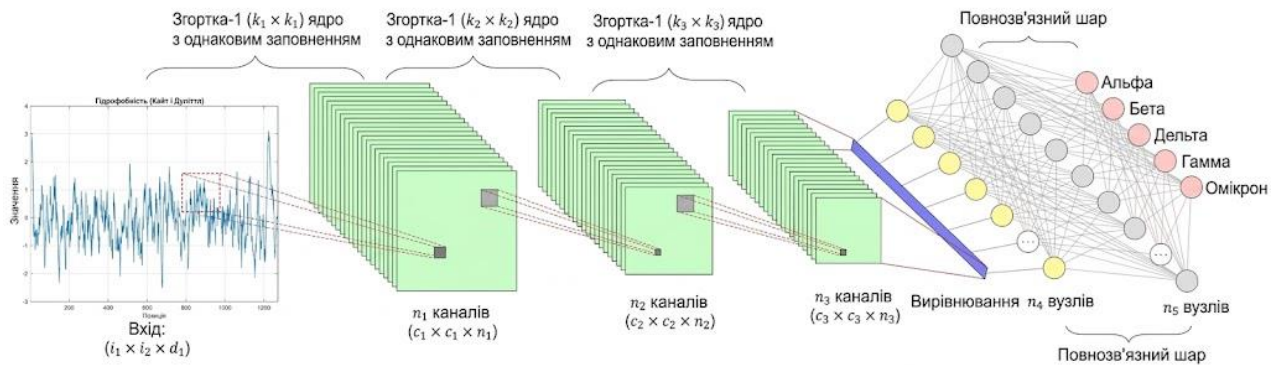


Рисунок 1.2 - Архітектура ЗНМ для задачі класифікації зображень зразків продукції. Місце для вставки: після опису CNN та ResNet [76]

Специфікою промислового контролю якості є сильний дисбаланс класів: кількість якісної продукції зазвичай у тисячі разів перевищує кількість браку. Навчання класифікаторів на таких незбалансованих вибірках часто призводить до того, що модель ігнорує рідкісні класи дефектів [18]. Для вирішення цієї проблеми застосовуються моделі навчання без учителя, зокрема методи виявлення аномалій [19].

Ключовим підходом тут є використання автоенкодерів - нейронних мереж, які навчаються стискати вхідні дані у компактний вектор прихованого стану, а потім відновлювати їх з максимальною точністю. Модель тренується виключно на зображеннях якісного борошна [20]. При подачі на вхід зображення з дефектом автоенкодер не може коректно його відновити, що призводить до високої помилки реконструкції. Величина цієї помилки слугує індикатором аномалії, що дозволяє системі реагувати на будь-які, навіть раніше невідомі види дефектів, не потребуючи збору великої бази прикладів браку [21].

Вибір конкретної моделі МН для інтеграції в КФС є результатом багатокритеріального аналізу, що враховує не лише метрологічні характеристики, але й обчислювальну складність алгоритмів. У таблиці 1.1 наведено порівняльну характеристику основних класів моделей у контексті їх застосування для аналізу якості борошна.

Таблиця 1.1 - Порівняльний аналіз моделей МН для задач контролю якості

Клас моделей	Типові алгоритми	Переваги	Недоліки	Сфера застосування
1	2	3	4	5
Класичні (навчання з учителем)	SVM, Random Forest, k-NN	Висока інтерпретованість, робота з малими даними, низькі вимоги до CPU	Потребують складного Feature Engineering, низька ефективність на зображеннях	Аналіз лабораторних даних, табличні дані, прості сенсори
Глибинне навчання	ЗНМ (ResNet, VGG), LSTM	Автоматичне виділення ознак, найвища точність для візуальних даних	Потребують великих датасетів та GPU, принцип «чорної скриньки»	КЗ, спектральний аналіз, класифікація текстур
Детектори об'єктів	YOLO, SSD, Faster R-CNN	Локалізація дефектів, робота в реальному часі	Висока обчислювальна складність	Потоковий контроль, виявлення домішок

Кінець таблиці 1.1

Навчання без учителя	Autoencoders, One-Class SVM	Виявлення нових/невідомих дефектів, не потребують розмітки браку	Складність налаштування порогу спрацювання, менша точність класифікації	Пошук аномалій, контроль стабільності процесу
----------------------	-----------------------------	--	---	---

Аналіз показує, що для побудови комплексної системи збору та аналізу якості доцільним є використання гібридної архітектури. На нижньому рівні ефективно застосовувати легковагові класичні моделі або оптимізовані ЗНМ для швидкого реагування в реальному часі. Натомість складні моделі глибокого навчання можуть бути розгорнуті на серверному рівні для детального аналізу трендів та періодичного донавчання системи. Такий підхід дозволяє збалансувати вимоги до точності, швидкодії та вартості апаратного забезпечення.

## 1.2 Аналіз методів автоматизованого контролю та збору даних якості з використанням машинного навчання

Створення КФС для борошномельної промисловості вимагає перегляду традиційних підходів до метрологічного забезпечення виробництва[22]. Якщо класична лабораторна практика базується на дискретному відборі проб та їх руйнівному хімічному аналізі, то парадигма цифрового виробництва передбачає використання методів безперервного неруйнівного контролю [23, 24].

Ефективність кінцевої системи залежить не стільки від обраної архітектури нейронної мережі, скільки від якості сформованого методологічного підходу до отримання первинних даних, їх репрезентативності та стійкості до виробничих завад [25].

Ключовим аспектом проєктування підсистеми збору даних є вибір топології вимірювання відносно технологічного потоку. У науковій літературі та стандартах

автоматизації виділяють три основні методи організації контролю, кожен з яких накладає специфічні вимоги до алгоритмів обробки даних [26].

Метод In-line передбачає розміщення сенсорів безпосередньо у транспортній магістралі (наприклад, у самопливній трубі або пневмотранспорті), де сенсор контактує з продуктом постійно. Перевагою методу є відсутність затримки часу вимірювання, що дозволяє реалізувати миттєве керування виконавчими механізмами. Однак цей підхід є найбільш складним з точки зору збору даних, оскільки вимагає від методів МН стійкості до вібрацій, запилення оптики та зміни швидкості потоку продукту [27, 28].

Метод On-line базується на автоматичному відборі частини потоку борошна в бічну вимірювальну камеру, проведення аналізу та повернення продукту в основний потік. Цей метод забезпечує кращі умови для стабілізації зразка (наприклад, ущільнення борошна перед спектральним аналізом), що підвищує точність вимірювання, але вносить часову затримку, яка може бути критичною для швидкоплинних процесів [29].

Метод At-line передбачає використання автоматичних пробовідбірників, які доставляють зразок до локального аналізатора, розташованого в цеху. Хоча цей метод забезпечує найвищу точність, близьку до лабораторної, його дискретність не дозволяє виявляти короткочасні флуктуації якості, що підтверджується теорією репрезентативного пробовідбору П. Есбенсена [30, 31].

Для розроблюваної КФС найбільш перспективним є комбінування методів In-line (для візуальної дефектоскопії) та On-line (для спектрального аналізу), що дозволяє досягти балансу між оперативністю та точністю даних.

Візуальна оцінка залишається базовим методом визначення сортності борошна, оскільки колір продукту має високу кореляцію із зольністю - вмістом периферійних частинок зерна. Методи КЗ трансформують суб'єктивне сприйняття кольору людським оком у об'єктивні цифрові метрики [32, 33].

Основою методу є перетворення зображення з простору RGB, який є апаратно-залежним і чутливим до змін освітлення, у перцептульно-рівномірні колірні простори, такі як CIE Lab\* або HSV. Канал *L* у цьому просторі математично

корелює зі стандартними показниками вимірювачів білизни [34]. Проте просте вимірювання середнього кольору є недостатнім.

Сучасні методи включають етап сегментації зображення для відділення пікселів борошна від фону та тіней, що виникають через нерівність поверхні насипу. Для цього застосовують адаптивні порогові методи (наприклад, метод Оцу) або моделі семантичної сегментації, що дозволяє працювати в умовах змінного освітлення [35, 36].

Загальна структура методу обробки візуальних даних представлена на рисунку 1.3. На схемі відображено послідовність етапів перетворення інформації: процес розпочинається з введення зображення та його оцифрування. Далі виконується блок попередньої обробки, де усуваються шуми камери та коригується баланс білого. Ключовим етапом є сегментація, яка дозволяє виділити області інтересу - окремі частинки або плями забруднень. На виході блоку «Вилучення ознак» формується вектор числових характеристик (текстурні дескриптори, геометричні параметри), який передається в систему координат для подальшої класифікації або навігації робототехнічних маніпуляторів.

Окремим напрямом є метод морфологічного аналізу вкраплень. Борошно не є гомогенним середовищем; воно містить частинки висівок, які відрізняються контрастом. Методика збору даних у цьому випадку передбачає використання макрозйомки з високою роздільною здатністю. Алгоритм аналізу базується на виявленні локальних градієнтів яскравості (детектор граней Кенні або оператор Собеля) з подальшим розрахунком геометричних характеристик кожного об'єкта: площі, периметра, фактора форми (сферичності) [37]. Це дозволяє не лише підрахувати кількість вкраплень (що нормується стандартом на «крупність»), а й класифікувати їх тип - висівки, частинки металу або інші домішки [38, 39].

Важливим елементом методології є використання текстурного аналізу. Оскільки візуальна структура борошна вищого та першого ґатунку відрізняється мікрорельєфом, застосовуються методи на основі матриць суміжності рівнів сірого. Розрахунок таких статистичних ознак, як ентропія, енергія, контраст та кореляція, дозволяє сформулювати вектор ознак, який подається на вхід класифікатора МН.

Використання текстурного аналізу значно підвищує роздільну здатність системи порівняно з простим аналізом гістограми яскравості [40, 41].

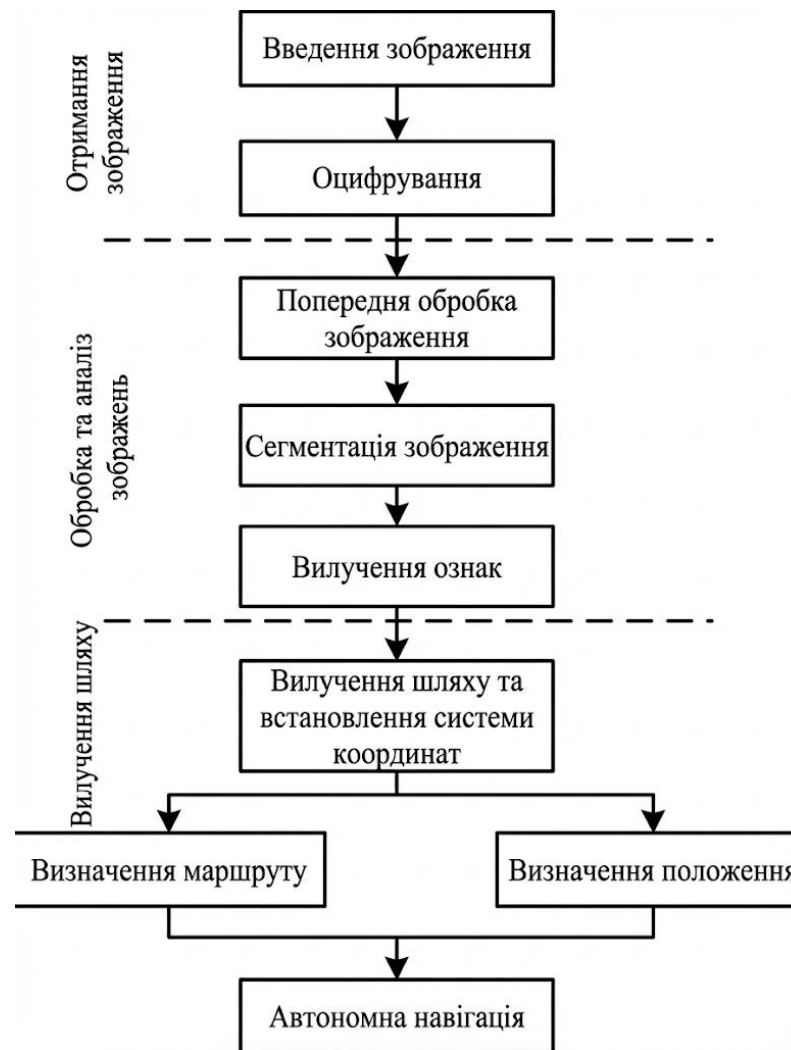


Рисунок 1.3 - Етапи методу КЗ: від отримання сирого зображення до сегментації краплень та аналізу текстури [77]

КЗ аналізує поверхню, а методи спектроскопії дозволяють отримати інформацію про хімічний склад (волога, білок, клейковина) без використання реагентів. Метод NIR-спектроскопії базується на реєстрації спектрів поглинання електромагнітного випромінювання в діапазоні 780-2500 нм. Органічні молекули, що містять зв'язки С-Н, N-H та О-Н, поглинають енергію на специфічних частотах, що формує унікальний «спектральний відбиток» зразка [42, 43].

Типовий спектр борошна (рисунок 1.4) має складну хвилеподібну структуру. На графіку можна виділити характерні піки поглинання, що відповідають основним компонентам: широкі смуги в районі 1450 нм та 1940 нм вказують на присутність води (О-Н зв'язки), тоді як область 2100-2200 нм містить інформацію про амінокислоти та білкові структури (N-H зв'язки). Інтенсивність цих піків прямо пропорційна концентрації речовини, що дозволяє будувати калібрувальні моделі для кількісного аналізу.

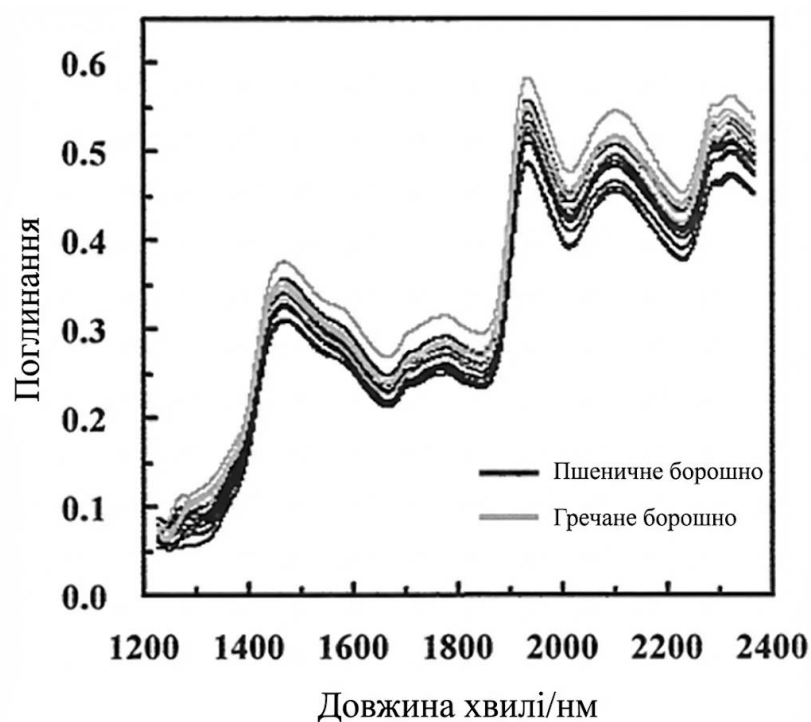


Рисунок 1.4 - Типовий NIR-спектр борошна з виділеними зонами поглинання [78]

Специфікою методу при зборі даних з борошна є сильне розсіювання світла на частинках продукту. Це призводить до того, що корисний сигнал становить лише малу частку від загальної інтенсивності, а базовий рівень спектра постійно дрейфує залежно від відстані до датчика та щільності насипу [44]. Тому невіддільною частиною методу є хемометрична обробка даних. Вона включає застосування алгоритмів згладжування (фільтр Савіцького-Голея) для усунення високочастотного шуму сенсора, а також методи корекції розсіювання [45, 46]. Ці перетворення є обов'язковим етапом перед подачею даних у моделі МН, оскільки

вони нормалізують вхідний вектор і роблять систему інваріантною до фізичних умов вимірювання [47].

Сучасною еволюцією методу є гіперспектральний імідинг, який поєднує просторову інформацію (зображення) зі спектральною. На відміну від точкового NIR-датчика, що дає усереднене значення по об'єму, гіперспектральний імідинг дозволяє побудувати карту розподілу вологості або клейковини по поверхні потоку. Це дає можливість виявляти локальні неоднорідності змішування та фальсифікацію продукції іншими видами крохмалю, що є критичним для забезпечення стабільності хлібопекарських властивостей [48, 49].

В умовах, коли пряме вимірювання якісних показників є неможливим у реальному часі (наприклад, визначення показника числа падіння або реологічних властивостей тіста), застосовується метод віртуальних сенсорів [50]. Цей методологічний підхід базується на припущенні, що якість готового продукту є функцією від параметрів технологічного процесу його виробництва.

Збір даних для віртуальних сенсорів передбачає агрегацію інформації з промислових контролерів та IoT-пристроїв, що керують млиновим обладнанням. До інформативних параметрів належать: сила струму на вальцьових верстатах (що корелює з твердістю зерна та ступенем пошкодження крохмалю), температура підшипників, вібрація розсівів, вологість повітря в цеху та витрати води на зволоження зерна [51, 52].

Методологія побудови віртуального сенсора включає етап синхронізації часових рядів. Оскільки проходження зерна від приймального бункера до виходу готового борошна займає певний час (транспортне запізнення), дані з датчиків на початку лінії повинні бути зіставлені з результатами лабораторного аналізу борошна з відповідним часовим зсувом. Використання рекурентних нейронних мереж дозволяє враховувати цю динаміку та інерційність процесу, прогнозуючи якість борошна на основі історії режимів роботи обладнання [53]. Такий підхід дозволяє реалізувати предиктивний контроль, попереджаючи оператора про можливе відхилення якості ще до завершення помелу партії, що є основою концепції «Digital Twin» (цифровий двійник) у харчовій промисловості [54, 55].

Ефективність КФС досягається не окремим використанням описаних методів, а їх комплексуванням. Методологія мультимодального аналізу передбачає одночасну обробку візуальних даних (для оцінки сортності та чистоти), спектральних даних (для хімічного складу) та технологічних параметрів (для непрямой оцінки реології) [56].

Злиття даних може відбуватися на трьох рівнях: рівень даних, рівень ознак та рівень рішень. Для задач контролю якості борошна найбільш доцільним є злиття на рівні ознак, що дозволяє нейронній мережі знаходити приховані кореляції між візуальним виглядом борошна та його спектральними характеристиками, підвищуючи загальну надійність ідентифікації сорту [57, 58].

### 1.3 Аналіз апаратних та програмних засобів реалізації підсистеми збору даних та аналізу

Апаратні та програмні засоби вимагають використання комплексного підходу, здатного функціонувати в специфічних умовах борошномельного виробництва. На відміну від лабораторного обладнання, промислова система збору даних повинна відповідати жорстким вимогам щодо вібростійкості, пилозахисту (клас захисту IP65/IP67) та електромагнітної сумісності. Ключовим завданням є створення надійного тракту отримання первинної інформації від фізичного сенсора до обчислювального ядра, де відбувається виконання моделей МН [59].

Підсистема збору даних є фундаментом КФС, оскільки якість вхідного сигналу визначає граничну теоретичну точність роботи будь-якого алгоритму. Враховуючи, що основними методами контролю обрано КЗ та спектроскопію, центральним елементом апаратної частини виступають оптичні сенсори промислового класу [60].

Для задач візуальної інспекції потоку борошна критичним параметром є тип затвора матриці камери. Використання бюджетних сенсорів із рядковим затвором, які послідовно зчитують рядки пікселів, є неприпустимим для об'єктів, що рухаються на конвеєрі, оскільки це призводить до геометричних спотворень

зображення. Тому в КФС реального часу застосовуються виключно камери з глобальним затвором, які експонують усі пікселі матриці одночасно, фіксуючи чіткий «зліпок» стану продукту в конкретний момент часу. Окрім того, важливу роль відіграє спектральна чутливість матриці: для виявлення сторонніх домішок часто використовують монохромні камери з розширеним динамічним діапазоном, які забезпечують кращий контраст між частинками ендосперму та висівками порівняно зі звичайними RGB-сенсорами [61].

Для реалізації спектрального аналізу в компактних системах дедалі частіше використовуються MEMS-спектрометри (мікроелектромеханічні системи). На відміну від громіздких лабораторних приладів з рухомими дифракційними решітками, MEMS-чіпи інтегруються безпосередньо в плату пристрою збору даних. Вони забезпечують вимірювання спектра відбиття в ближньому інфрачервоному діапазоні (900-1700 нм), що корелює з вмістом вологи та клейковини. Хоча їхня роздільна здатність є нижчою за лабораторні аналоги, вона є достатньою для експрес-аналізу в поєднанні з алгоритмами хемометрики [62].

Узагальнена структурна схема сенсорного вузла наведена на рисунку 1.5. Вона демонструє взаємодію ключових компонентів: оптичний модуль (камера або спектрометр) передає дані через високошвидкісний інтерфейс (наприклад, MIPI CSI-2) на блок попередньої обробки. Важливим елементом є керована система підсвічування, синхронізована з моментом експозиції, що забезпечує стабільність умов зйомки незалежно від зовнішнього освітлення в цеху.

Обробка потокового відео та спектральних даних вимагає значних обчислювальних ресурсів, які не можуть бути забезпечені традиційними мікроконтролерами загального призначення. Передача «сирих» даних у хмару для аналізу є недоцільною через високу затримку та вимоги до пропускної здатності мережі. Тому сучасна архітектура КФС базується на концепції граничних обчислень (Edge Computing), що передбачає виконання розпізнавання нейронних мереж безпосередньо на пристрої збору даних [63, 64].

Для реалізації цієї концепції використовуються спеціалізовані одноплатні комп'ютери з апаратними прискорювачами штучного інтелекту. Найпоширенішим

класом таких пристроїв є платформи на базі графічних процесорів, зокрема серія NVIDIA Jetson (рисунок 1.6). Їхня архітектура містить сотні ядер CUDA, оптимізованих для паралельних матричних обчислень, що лежать в основі глибокого навчання. Це дозволяє виконувати обробку зображень з частотою 30-60 кадрів на секунду, що є достатнім для управління сортувальними механізмами [65].

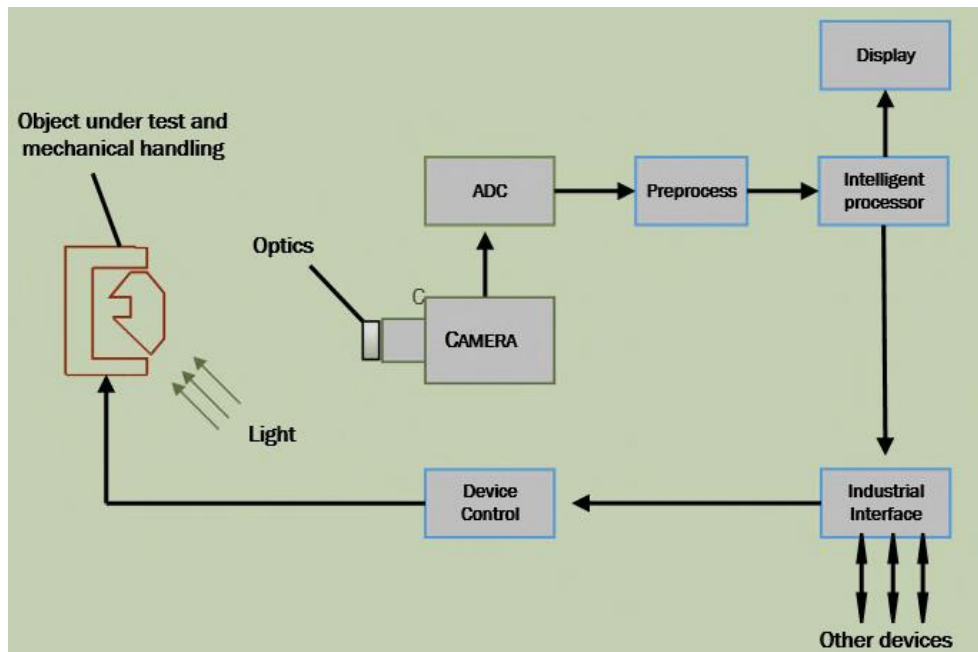


Рисунок 1.5 - Структурна схема апаратного вузла збору даних КФС [79]



Рисунок 1.6 - одноплатний комп'ютер NVIDIA Jetson [80]

Альтернативним підходом є використання тензорних процесорів, таких як Google Coral. Ці спеціалізовані інтегральні схеми розроблені виключно для виконання операцій з тензорами і демонструють вищу енергоефективність порівняно з GPU. TPU ідеально підходять для вбудованих систем з обмеженим живленням, однак їхня гнучкість обмежена підтримкою специфічних форматів моделей. Порівняльний аналіз, наведений у таблиці 1.2, дозволяє оцінити придатність різних платформ для задач контролю якості борошна.

В таблиці 1.2 вказуються порівняльна характеристика обчислювальних платформ. Враховуючи характеристики, платформи на базі GPU забезпечують найкращий баланс між продуктивністю та зручністю розробки, тоді як рішення на базі FPGA доцільні лише для задач із наджорсткими вимогами до затримки (менше 1 мс).

Таблиця 1.2 - Порівняльна характеристика обчислювальних платформ для Edge AI

Платформа	Тип прискорювача	Продуктивність (TOPS)	Енергоспоживання (Вт)	Підтримка ПЗ
NVIDIA Jetson Nano	GPU (128 CUDA cores)	0.5	5-10	PyTorch, TensorFlow, TensorRT
Raspberry Pi 5 + Coral TPU	CPU + TPU (ASIC)	4.0	8-15	TensorFlow Lite
STM32MP1	Neural Processing Unit	1.2	< 3	STM32 Cube.AI
FPGA (Xilinx/Intel)	Програмована логіка	Варіативна	Висока	VHDL/Verilog, Custom IP cores

Інтеграція розробленої системи в загальну інфраструктуру підприємства вимагає надійних каналів зв'язку. У промислових умовах, де присутні сильні електромагнітні завади від потужних електродвигунів млинів, пріоритет надається дротовим інтерфейсам. Для передачі відеопотоку високої роздільної здатності стандартом є інтерфейси GigE Vision або USB 3.0, які забезпечують необхідну пропускну здатність без втрати пакетів даних [66].

Для обміну даними телеметрії та керуючими сигналами (наприклад, команда на відбраковку партії) використовуються протоколи промислового Інтернету речей. Традиційний протокол Modbus TCP поступово витісняється сучасними рішеннями, такими як MQTT та OPC UA. Протокол MQTT, що працює за моделлю «видавець-підписник», дозволяє ефективно передавати дані від багатьох сенсорів до центрального сервера навіть за умов нестабільного з'єднання. Водночас OPC UA є стандартом для Індустрії 4.0, забезпечуючи не лише транспорт даних, а й їх семантичний опис та шифрування, що дозволяє безпосередньо інтегрувати аналізатор якості з SCADA-системами підприємства [67, 68].

Програмна складова КФС охоплює не лише алгоритми аналізу, а й інструменти для їх адаптації під обмежені ресурси вбудованих систем. Розробка та тренування моделей зазвичай виконуються на потужних робочих станціях з використанням фреймворків глибокого навчання, таких як TensorFlow або PyTorch. Однак прямий перенос моделі з сервера на Edge-пристрій часто є неможливим через брак оперативної пам'яті [69].

Для вирішення цієї проблеми застосовують методи оптимізації, зокрема квантування. Цей процес передбачає зменшення розрядності ваг нейронної мережі з 32-бітних чисел із плаваючою комою (FP32) до 8-бітних цілих чисел (INT8). Дослідження показують, що квантування дозволяє зменшити розмір моделі в 4 рази та прискорити її виконання на Edge-пристроях у 2-3 рази з мінімальною втратою точності (менше 1%). Для автоматизації цього процесу використовуються спеціалізовані конвертери, такі як TensorFlow Lite Converter або NVIDIA TensorRT [70].

Важливу роль відіграють також бібліотеки КЗ, насамперед OpenCV. Вони забезпечують базові операції попередньої обробки зображень (фільтрація, корекція кольору, зміна розміру) ще до подачі даних на вхід нейромережі. Ефективне використання цих бібліотек дозволяє розвантажити нейронний прискорювач і підвищити загальну швидкодію системи.

На рисунку 1.7 наведено діаграму потоків даних у програмному забезпеченні КФС. Процес починається з захоплення кадру драйвером камери, далі відбувається його попередня обробка засобами центрального процесора, передача тензора в пам'ять прискорювача для розпізнавання, і, нарешті, пост-обробка результатів та їх відправка через мережевий інтерфейс.

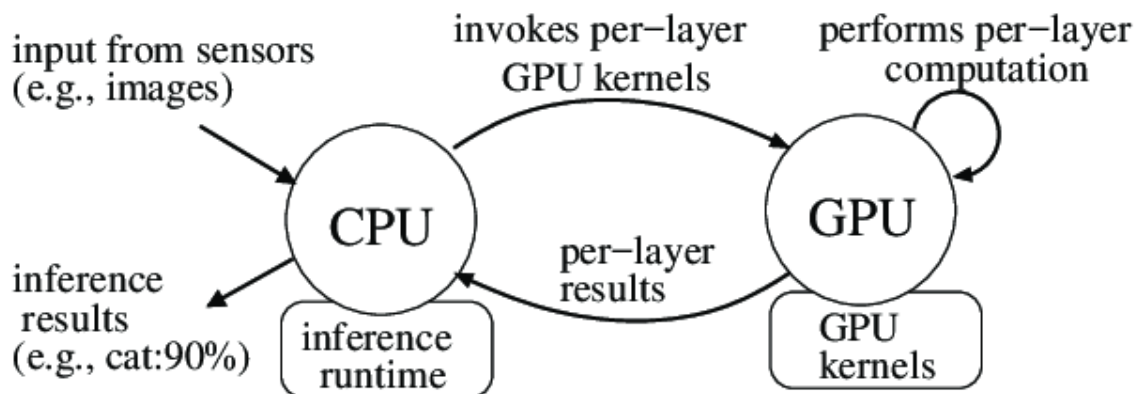


Рисунок 1.7 - Діаграма потоків даних у програмному забезпеченні системи аналізу [81]

Підсумовуючи аналіз апаратних та програмних засобів, можна стверджувати, що сучасний ринок пропонує достатній набір компонентів для створення ефективної КФС. Комбінація камер з глобальним затвором, обчислювачів класу Edge AI та оптимізованих нейромережевих моделей дозволяє реалізувати автоматизований контроль якості борошна в режимі реального часу, забезпечуючи інтеграцію з існуючими виробничими лініями.

#### 1.4 Аналіз українських наукових та прикладних ініціатив у сфері автоматизації контролю якості

Розвиток вітчизняного агропромислового комплексу та переробної галузі характеризується стратегічною переорієнтацією з сировинного експорту на виробництво продукції з високою доданою вартістю. В умовах гармонізації українських стандартів з регламентами Європейського Союзу, вимоги до стабільності якісних показників борошна суттєво зростають. Це формує об'єктивний запит на впровадження автоматизованих систем об'єктивного контролю, здатних мінімізувати вплив «людського фактора» та корупційних ризиків при визначенні сортності продукції. Аналіз стану галузі дозволяє виділити декілька векторів розвитку вітчизняних ініціатив у цій сфері.

Основний науковий потенціал у сфері автоматизації харчових виробництв зосереджений на базі провідних технічних університетів (зокрема, НУХТ, ОНТУ, КПІ ім. Ігоря Сікорського). Діяльність вітчизняних наукових шкіл традиційно спрямована на математичне моделювання тепломасообмінних процесів та розробку первинних перетворювачів.

Вагомим здобутком українських вчених є розвиток теорії та практики вологометрії. Вітчизняні розробки ємнісних та НВЧ-сенсорів для потокового контролю вологості зерна вирізняються високою надійністю та адаптованістю до фізичних властивостей українських сортів пшениці. Проте, функціонал таких систем часто обмежується вимірюванням одного-двох параметрів (волога, температура) і не дозволяє комплексно оцінювати інтегральні показники якості, такі як білизна чи наявність специфічних домішок, що потребує залучення оптичних методів [71].

Останнім часом активізувалися дослідження у напрямі КЗ та кваліметрії. Наукові групи досліджують застосування цифрової обробки зображень для визначення геометричних параметрів зернівок, їх склоподібності та фракційного складу продуктів розмелу. Водночас, більшість цих робіт залишається на рівні лабораторних експериментів або теоретичних моделей. Впровадження методів

глибокого навчання та нейронних мереж у реальні системи керування технологічними процесами українських млинзаводів поки що носить фрагментарний характер, що пов'язано з відсутністю доступних апаратних платформ вітчизняного виробництва для реалізації Edge Computing.

На прикладному рівні український ринок зернопереробного обладнання представлений потужними машинобудівними підприємствами (завод «ОЛІС», «Хорольський механічний завод», KMZ Industries). Ці виробники успішно конкурують із закордонними компаніями у сегменті механічного технологічного обладнання: виробляють високопродуктивні сепаратори, розсів, трієри та ситовийні машини, що відповідають сучасним вимогам енергоефективності.

Однак у сегменті сенсорики та інтелектуальних систем управління спостерігається суттєва залежність від імпорту. Для лабораторного аналізу якості більшість експорторієнтованих підприємств змушені закуповувати дорогі аналізатори провідних брендів (Pertin Instruments, Chopin Technologies, Brabender). Вартість таких комплексів є критичним бар'єром для малих та середніх виробників, які складають значну частку внутрішнього ринку. Як наслідок, на таких підприємствах досі домінують застарілі, ручні методи контролю, що унеможливорює оперативне реагування на зміну якості сировини [72].

Специфічною проблемою української галузі є високий ступінь зношеності основних фондів та наявність великої кількості млинів, спроектованих ще за радянських часів. Повна заміна технологічних ліній на сучасні автоматизовані комплекси є економічно невідомою для більшості власників.

Це породжує попит на модернізації старого обладнання шляхом встановлення сучасних сенсорів та локальних систем автоматизації. Українські компанії активно працюють у цьому напрямі, впроваджуючи системи автоматизованого керування електроприводами та дозуванням. Проте ніша автоматизованого контролю якості в рамках ретрофітінгу залишається незаповненою. Відсутні бюджетні, компактні модулі, які можна було б легко інтегрувати в існуючі самопливні труби старого млина для моніторингу якості борошна в реальному часі без суттєвої реконструкції лінії.

Україна має потужний IT-сектор, що стимулює розвиток стартапів у сфері агротехнологій. Існуючі вітчизняні рішення (системи супутникового моніторингу полів, ERP-системи для елеваторів, дрони для обприскування) демонструють високий рівень інноваційності. Проте фокус цих розробок переважно зміщений на етап вирощування та зберігання врожаю.

Узагальнюючи аналіз вітчизняних ініціатив, можна констатувати наявність технологічного вакууму:

1. Відсутність комплексних рішень середнього цінового сегмента. Ринок поляризований між дешевими ручними приладами та наддорогими імпортними автоматизованими лабораторіями.

2. Потреба в адаптивних алгоритмах так, як Закордонні системи часто постачаються із закритими калібруваннями, які не завжди враховують специфіку українських сортів зерна, що вимагає розробки відкритих систем з можливістю донавчання на локальних даних.

3. Необхідність бюджетної автоматизації для створення недорогих IoT-рішень, доступних для малого бізнесу, які б дозволили наблизити рівень контролю якості до європейських стандартів.

Саме ці фактори визначають доцільність розробки КФС збору та аналізу якості на основі МН, яка поєднує доступну елементну базу з передовими методами обробки інформації.

### 1.5 Постановка задачі дослідження

На основі проведеного аналізу стану галузі, методів контролю та існуючих апаратних засобів можна сформулювати основну концепцію магістерської роботи. Виявлена суперечність між необхідністю суцільного контролю якості борошна та дискретністю існуючих лабораторних методів вимагає створення автоматизованої системи, що працює в режимі реального часу.

Метою роботи є підвищення ефективності та оперативності процесу контролю якості борошна шляхом розробки КФС, що забезпечує автоматизований збір даних та їх інтелектуальний аналіз на основі моделей МН.

Для досягнення мети необхідно вирішити наступну сукупність задач:

1. Розробка архітектури системи: Обґрунтувати структуру КФС, яка поєднує рівень фізичного збору даних та рівень обробки і прийняття рішень, забезпечуючи баланс між вартістю та швидкістю.

2. Побудова підсистеми збору даних: Розробити схему сенсорного вузла, що включає оптичні засоби реєстрації та забезпечити умови для отримання репрезентативних даних (стабільне освітлення, синхронізація з потоком продукту).

Вибір та адаптація моделей МН:

1. Для задачі візуальної інспекції (визначення сортності, наявності висівкових краплень) обрати архітектуру ЗНМ, оптимізовану для роботи на вбудованих пристроях.

2. Для прогнозування непрямих параметрів якості реалізувати модель на базі ансамблевих методів або регресійних нейромереж.

3. Створення алгоритмічного забезпечення. Розробити алгоритми попередньої обробки зображень та програмний модуль для розпізнавання моделі на апаратній платформі Edge AI.

4. Експериментальне дослідження. Провести тестування розробленої системи на реальних зразках борошна, оцінити метрики точності класифікації та часові характеристики обробки даних.

Вимоги до системи: безперервний або з мінімальною дискретністю режим роботи, час реакції не більше 1-2 секунд на обробку одного кадру/зразка, точність класифікації не нижче 90% для основних товарних сортів, можливість передачі даних за протоколами ІоТ.

## 1.6 Висновки

Виконано комплексний аналіз сучасного стану проблеми автоматизації контролю якості у борошномельній промисловості. Основні результати аналізу дозволяють зробити наступні висновки:

Встановлено, що сучасна парадигма харчової промисловості передбачає перехід від лабораторних руйнівних методів до методів неруйнівного контролю в потоці (in-line/on-line). Найбільш перспективними для цієї задачі є методи КЗ (для оцінки білизни та геометричних параметрів частинок) та NIR-спектроскопії (для аналізу хімічного складу).

Аналіз математичних моделей показав, що для задач зі складною, нелінійною природою даних (зображення, спектри) використання детермінованих алгоритмів є неефективним. Доведено доцільність застосування ЗНМ для візуального аналізу та ансамблевих методів для обробки структурованих даних сенсорів.

Апаратна реалізація. Розвиток технологій Edge Computing зробив доступними потужні обчислювальні платформи (NVIDIA Jetson, Google Coral, STM32 AI), які дозволяють виконувати складні алгоритми штучного інтелекту безпосередньо на місці вимірювання. Це вирішує проблему затримок передачі даних та дозволяє створювати компактні «розумні» сенсори.

Актуальність для України. Аналіз вітчизняного ринку виявив дефіцит доступних автоматизованих рішень для контролю якості. Більшість українських підприємств використовують застарілі методи або залежать від дороговартісного імпортного обладнання. Створення бюджетної КФС на базі IoT та ML є актуальним науково-прикладним завданням, що має значний потенціал для впровадження на малих та середніх млинах.

На основі проведеного аналізу обґрунтовано необхідність та доцільність розробки КФС збору та аналізу якості борошна, яка поєднує методи КЗ та МН, що і є предметом подальших розділів даної роботи.

## 2 МОДЕЛЮВАННЯ ПРОЦЕСІВ ФУНКЦІОНУВАННЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ КОНТРОЛЮ ЯКОСТІ БОРОШНА

### 2.1 Концептуальна модель та формалізовані критерії адекватності системи

Для досягнення мети підвищення ефективності контролю якості борошна шляхом переходу до безперервного автоматизованого моніторингу сформовано формалізоване описання КФС, що реалізує функції збору та інтелектуального аналізу даних. З цією метою побудовано концептуальну модель системи, яка встановлює відповідність між фізичним процесом переміщення продукту та його цифровим представленням у вигляді потоку даних для подальшої обробки. Реалізований підхід дозволяє узагальнити процес контролю якості як задачу перетворення вхідної інформації про стан продукту у формалізовані рішення щодо його відповідності встановленим критеріям.

Концептуально система може представлена як оператор перетворення вхідних даних у множину рішень, що описується відображенням вигляду:

$$S: X(t) \rightarrow Y(t), \quad (2.1)$$

де  $X(t)$  - множина вхідних даних, що характеризують стан продукту в момент часу  $t$ ;

$Y(t)$  - множина вихідних оцінок якості;

$S$  - оператор перетворення, що реалізує функцію інтелектуального аналізу. Введене відображення формалізує загальний принцип функціонування системи як процес переходу від вимірюваних характеристик до прийняття рішень.

Вхідні дані формуються на основі цифрового відеопотоку та допоміжних параметрів середовища і можуть формалізовані як:

$$X(t) = \{I(t), Z(t)\}, \quad (2.2)$$

де  $I(t)$  - цифрове зображення потоку борошна в момент часу  $t$ ;

$Z(t)$  - множина допоміжних параметрів, що характеризують умови формування зображення, зокрема освітлення, геометрію сцени та параметри сенсорного обладнання.

Представлення дозволяє врахувати вплив зовнішніх факторів на процес аналізу та підвищити адекватність моделі.

Вихід системи включає результати класифікації та індикатор наявності аномалій і подається у вигляді:

$$Y(t) = \{ C(t), A(t) \}, \quad (2.3)$$

де  $C(t)$  - результат класифікації якості, що належить скінченній множині класів;

$A(t)$  - бінарна змінна, яка відображає наявність або відсутність морфологічної аномалії в продукті. Вихід системи поєднує як оцінку відповідності стандартам, так і детекцію дефектів.

Вимогою КФС є її робота в умовах жорстких часових обмежень, що визначаються динамікою технологічного процесу. Загальний час реакції системи визначається як сума окремих етапів обробки:

$$T_{response} = T_{capture} + T_{processing} + T_{decision}, \quad (2.4)$$

де  $T_{capture}$  - час отримання зображення сенсором;

$T_{processing}$  - час обчислювальної обробки даних;

$T_{decision}$  - час формування рішення щодо якості продукту. Сумарний час визначає швидкодію системи та її здатність працювати в умовах реального часу.

З позиції системного аналізу розглянута система може бути інтерпретована як замкнений контур обробки інформації, в якому вхідні дані перетворюються у рішення, що потенційно можуть формувати керуючі впливи на фізичний процес. Взаємодія описується композицією відображень:

$$X(t) \xrightarrow{F} Y(t) \xrightarrow{G} U(t), \quad (2.5)$$

де  $F$  - функція інтелектуального аналізу вхідних даних, яка реалізує перетворення зображення та супутніх параметрів у оцінку якості продукту;

$G$  - функція формування керуючого впливу на основі отриманих результатів;

$U(t)$  - керуючий сигнал, що визначає подальшу обробку продукту. Наведена форма запису відображає узагальнену структуру КФС як послідовність перетворень інформації від сенсорного рівня до рівня прийняття рішень.

Концептуальна модель визначає загальну структуру перетворення інформації у КФС та встановлює зв'язок між фізичним об'єктом спостереження і цифровими методами його оцінювання. Подальша деталізація цієї моделі потребує формалізації задачі ідентифікації якості борошна та побудови відповідних математичних залежностей, що розглядається у наступному підрозділі.

## 2.2 Математична модель задачі ідентифікації якості борошна

У межах побудованої концептуальної моделі КФС задача аналізу якості борошна формалізована як задача відображення вхідних даних у дискретний простір класів якості. Така постановка базується на припущенні про існування функціональної залежності між характеристиками зображення продукту та його фізико-хімічними властивостями, що визначають сортність і наявність дефектів. Процес контролю якості розглядається як задача класифікації, у якій на основі спостережуваних даних необхідно визначити належність продукту до одного з наперед визначених класів.

У кожний момент часу система формує вектор ознак, що описує поточний стан продукту. Тоді задачу ідентифікації представлено у вигляді відображення:

$$Y = F(X), \quad (2.6)$$

де  $X$  - вектор ознак, що характеризує об'єкт;

$Y \in \{c_1, c_2, \dots, c_n\}$  - множина класів якості;

$F$  - функція класифікації. Введене відображення визначає загальний принцип прийняття рішень у системі, при якому кожному набору ознак ставиться у відповідність певний клас якості.

Вектор ознак сформовано на основі аналізу зображення і має вигляд:

$$X = (x_1, x_2, \dots, x_m), \quad (2.7)$$

де  $x_i$  - окрема ознака, що відображає певну властивість продукту, зокрема колірні характеристики, текстуру або просторову структуру часток. Застосований перехід дозволив перейти від неструктурованого зображення до формалізованого числового представлення, придатного для математичної обробки. При цьому вибір складу ознак безпосередньо впливає на точність і стійкість класифікації.

З огляду на варіативність умов виробництва та наявність шумів у даних, задачу класифікації розглянуто у ймовірнісній постановці. Функцію класифікації визначено як правило вибору найбільш імовірного класу:

$$F(X) = \arg \max P(c_i|X), \quad (2.8)$$

де  $P(c_i|X)$  - умовна ймовірність належності об'єкта до класу  $c_i$  за наявності вектора ознак  $X$ . Такий підхід дозволяє враховувати невизначеність і неоднозначність вхідних даних, що є характерним для реальних технологічних процесів.

Для практичної реалізації цього відображення використано параметризовану модель, подану у вигляді:

$$Y_{pred} = F(X, \theta), \quad (2.9)$$

де  $\theta$  - множина параметрів моделі;

$Y_{pred}$  - передбачене значення класу. Таким чином, задача ідентифікації зводиться до визначення таких параметрів  $\theta$ , які забезпечують мінімальну похибку класифікації на заданій вибірці даних.

Оцінювання якості моделі здійснено за допомогою функції втрат, що характеризує розбіжність між істинними та передбаченими значеннями. Для багатокласової класифікації використана крос-ентропійна функція:

$$L = - \sum_{i=1}^n y_i \log(p_i), \quad (2.10)$$

де  $y_i$  - істинне значення належності до класу;

$p_i$  - оцінена моделлю ймовірність належності до класу;

$n$  - кількість класів.

Мінімізація цієї функції дозволяє забезпечити узгодженість між моделлю та експериментальними даними.

Окрім задачі визначення сортності, система виконує виявлення морфологічних дефектів, що формалізовано як задача бінарної класифікації:

$$A = G(x), \quad (2.11)$$

де  $A \in \{0,1\}$  - ознака наявності дефекту;

$G$  - функція детекції аномалій. У цьому випадку рішення базується на визначенні відхилення поточного стану від нормального.

Таким чином, загальна задача ідентифікації якості представлено у вигляді поєднання двох функцій:

$$Y = (F(X), G(X)), \quad (2.12)$$

де  $F$  - виконує класифікацію сортності;

$G$  - виявлення аномалій.

Інтегрована математична модель дозволяє одночасно враховувати відповідність продукту стандартам та наявність дефектів для повноцінного контролю якості.

Отримана математична модель визначає базові принципи перетворення вхідних даних у результати аналізу, однак не деталізує структуру самих ознак. Модель простору ознак і способи його формування розглянуто у наступному підрозділі.

### 2.3 Модель простору ознак і вхідних даних

У задачі ідентифікації якості борошна ключовим етапом є формування інформативного представлення вхідних даних. Первинні дані, що надходять із сенсорної підсистеми, мають вигляд цифрового зображення і характеризуються високою розмірністю та значною надлишковістю. У такому вигляді вони є складними для безпосереднього використання в математичних моделях, тому необхідно виконати їх перетворення у компактний числовий опис. Це досягається шляхом побудови простору ознак, який відображає суттєві властивості продукту.

Нехай вхідні дані системи задаються зображенням, описаним як функція:

$$I = f(x, y, c), \quad (2.13)$$

де  $x, y$  - координати пікселя;

$c$  - індекс кольорового каналу.

У наведеному представленні кожна точка зображення відповідає певному значенню інтенсивності, а все зображення розглянуто як сукупність цих значень.

Безпосереднє використання масиву пікселів є неефективним, виконано попереднє перетворення зображення з метою підвищення інформативності ознак. Процес формалізовано у вигляді відображення:

$$I' = T(I), \quad (2.14)$$

де  $T$  - функція перетворення;

$I'$  - модифіковане представлення зображення.

Перетворення дозволяє зменшити вплив зовнішніх факторів, зокрема освітлення, та виділити більш стабільні характеристики продукту.

Наступним етапом стало формування вектора ознак:

$$X = H(I'), \quad (2.15)$$

де  $(H)$  - функція виділення ознак;

$X = (x_1, x_2, \dots, x_n)$  - вектор ознак.

Складне зображення замінено набором числових характеристик, які відображають суттєві властивості продукту.

Для детального аналізу розділено вектор ознак на складові частини:

$$X = (X_c, X_t, X_s), \quad (2.16)$$

де  $X_c$  - колірні ознаки;

$X_t$  - текстурні ознаки;

$X_s$  - структурні ознаки.

Поділ дозволяє розглядати різні аспекти зображення окремо, що підвищує інтерпретованість моделі.

Колірні ознаки визначаються через статистичні характеристики розподілу інтенсивності пікселів. Середнє значення інтенсивності обчислюється як:

$$\mu = \frac{1}{N} \sum_{x,y} I'(x, y), \quad (2.17)$$

де  $\mu$  - середнє значення інтенсивності;

$I'(x, y)$  - значення інтенсивності пікселя;

$N$  - загальна кількість пікселів.

Ця величина характеризує загальний рівень яскравості зображення і використовується як оцінка білизни продукту.

Для оцінки однорідності використане стандартне відхилення:

$$\sigma = \sqrt{\frac{1}{N} \sum_{x,y} (I'(x,y) - \mu)^2}. \quad (2.18)$$

Цей параметр відображає ступінь варіації значень інтенсивності. Низьке значення  $\sigma$  відповідає однорідному продукту, тоді як його збільшення свідчить про наявність неоднорідностей.

Сумісний аналіз параметрів  $\mu$  та  $\sigma$  дозволив оцінити як рівень яскравості, так і структуру продукту. Наприклад, поєднання високого середнього значення та малого стандартного відхилення відповідає якісному борошну без домішок.

Текстурні ознаки описують локальні зміни інтенсивності. Одним із базових інструментів виступає градієнт зображення  $G(x,y)$ , який характеризує швидкість зміни інтенсивності. Виявлені переходи можуть відповідати границям об'єктів або дефектам. Для обчислення градієнту використовуємо формулу:

$$G(x,y) = \sqrt{\left(\frac{I'}{\partial x}\right)^2 + \left(\frac{I'}{\partial y}\right)^2}. \quad (2.19)$$

Для узагальнення текстурної інформації здійснено перехід від локальних значень до інтегральної характеристики. Це можна зробити, обчисливши середнє значення градієнта  $G_{avg}$ :

$$G_{avg} = \frac{1}{N} \sum_{x,y} G(x,y) \quad (2.20)$$

Чим більше значення цієї величини, тим більш “зернистою” є структура зображення.

Структурні ознаки  $S$  характеризують геометричні властивості об'єктів, що виділяються на зображенні. Їх описані через сумарну площу:

$$S = \sum_{k=1}^K A_k, \quad (2.21)$$

де  $S$  - сумарна площа об'єктів;

$A_k$  - площа окремого об'єкта;

$K$  - кількість об'єктів.

Додатково для оцінки розподілу об'єктів вводиться щільність  $d$ , яка відображає кількість об'єктів на одиницю площі. Ця характеристика оцінює рівномірність розподілу часток:

$$d = \frac{K}{A_{total}}. \quad (2.22)$$

Сформована модель простору ознак визначає спосіб переходу від первинних даних до формалізованого числового представлення, використаного у задачі класифікації. Вона виступає зв'язуючою ланкою між сенсорними даними та математичною моделлю прийняття рішень. Подальший розвиток цієї моделі пов'язаний із визначенням структури функції класифікації, що розглянуто у наступному підрозділі.

## 2.4 Модель нейромережевої класифікації

Задача ідентифікації якості борошна сформовано як відображення, що встановлює відповідність між вектором ознак та класом якості. У реальних умовах така залежність є складною і не може бути описана простою аналітичною формулою. У зв'язку з цим використовуються моделі, які здатні адаптуватися до даних і відтворювати складні залежності. Одним із найбільш ефективних підходів є використання нейронних мереж.

У загальному вигляді нейромережева модель виконує перетворення вхідних даних у результат класифікації:

$$Y_{pred} = F(X, \theta), \quad (2.23)$$

де  $X$  - вектор ознак або зображення;

$\theta$  - параметри моделі;

$Y_{pred}$  - результат класифікації. У цьому випадку функція  $F$  визначається структурою мережі та її параметрами.

Оскільки вхідні дані мають вигляд зображення, доцільно використовувати згорткову модель, яка враховує просторову структуру даних. Обробка зображення відбувається поетапно, причому на кожному етапі формуються нові ознаки на основі попередніх. Процес подано як послідовність тензорних перетворень:

$$\begin{aligned} X_1 &= H_1(I), \\ X_2 &= H_2(X_1), \\ &\dots \\ X_n &= H_n(X_{n-1}), \end{aligned} \quad (2.24)$$

де  $H_i$  - окремий етап обробки;

$X_i$  - проміжне представлення даних.

На початкових етапах виділяються базові ознаки (контури, границі), а на глибших шарах - комплексні просторові структури.

Базовою операцією ЗНМ виступає згортка, яка забезпечує аналіз локальних областей зображення. Математично операцію записано у вигляді:

$$X_k(x, y) = \sum_{i=1}^a \sum_{j=1}^b I(x+i, y+j) * W_k(i, j) \quad (2.25)$$

де  $X_k(x, y)$  - результат згортки в точці  $(x, y)$ ;

$W_k(i, j)$  - коефіцієнти ядра згортки;

$a, b$  - розміри ядра;

$I(x + i, y + j)$  - значення інтенсивності в локальній області зображення.

Процедура згортання гарантує виділення характерних елементів мікрорельєфу продукту.

Після виконання згортки отримані тензори проходять через функцію активації, що вводить у модель нелінійність. У реалізованій архітектурі застосовано функцію ReLU, яка задається перетворенням  $X'$ :

$$X' = \max(0, X). \quad (2.26)$$

Зазначена функція обнуляє від'ємні значення тензора і залишає додатні без змін, фільтруючи шуми та виділяючи найбільш значущі піксельні активації.

Для зменшення просторового обсягу даних і агрегації ключових характеристик застосовано операцію субдискретизації:

$$X_{pool}(x, y) = \max \{X(x + i, y + j)\} \quad (2.27)$$

де  $i, j$  задають межі локальної області. Така операція дозволяє зберегти найбільш значущі значення та зменшити розмірність даних.

Після проходження через усі згорткові шари формується компактний вектор ознак, на основі якого повнозв'язний шар генерує фінальне рішення. Для цього використано нормування значень у вигляді:

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}} \quad (2.28)$$

де  $p_i$  - оцінена ймовірність належності до класу;

$z_i$  - значення, сформоване моделлю;

$n$  - кількість класів.

Нормування перетворює виходи мережі у дискретний розподіл імовірностей для визначення найбільш достовірного класу.

Реалізована нейромережева модель послідовно перетворює вхідні дані, автоматично виділяючи інформативні ознаки під час навчання без необхідності їх ручного програмування

Розглянута архітектура повністю визначає спосіб реалізації функції  $F(X, \theta)$ , що відповідає за ідентифікацію сортності продукту. Застосований підхід гарантує ефективну обробку зображень та відтворення складних нелінійних залежностей. Подальший розвиток алгоритмічної бази пов'язаний із формалізацією задачі виявлення морфологічних аномалій, що розглянуто у наступному підрозділі.

## 2.5 Модель виявлення аномалій

Складовою КФС виступає виявлення морфологічних аномалій, зокрема сторонніх домішок або порушення структури продукту. На відміну від класифікації, де всі класи визначені заздалегідь, у даному випадку необхідно визначати відхилення від нормального стану без повного переліку можливих дефектів. Це зумовило використання підходів, що базуються на аналізі відмінностей між поточним станом та еталонною моделлю.

У межах реалізованої моделі вектор ознак  $X = (x_1, x_2, \dots, x_m)$ , сформований на попередньому етапі, відновлюється за допомогою нейромережевої структури (автоенкодера), навченої виключно на нормальних зразках. У процесі відновлення модель формує оцінку значень ознак, яка максимально відповідає типовому стану продукту. Для еталонних даних ця оцінка є близькою до вхідних значень, тоді як для аномальних спостерігається суттєве відхилення.

Кількісною мірою відхилення виступає похибка відновлення  $D(X)$ , визначена як середнє квадратичне відхилення між початковими та відновленими значеннями:

$$D(X) = \frac{1}{m} \sum_{i=1}^m (x_i - x_i^{rec})^2, \quad (2.29)$$

де  $x_i$  - значення ознаки;

$x_i^{rec}$  - відновлене значення;

$m$  - кількість ознак.

Для якісних зразків значення  $D(X)$  залишається в межах вузького діапазону, оскільки модель коректно відтворює знайомі структури. При появі сторонніх включень похибка різко зростає, що дозволило використати її як основний критерій виявлення аномалій.

Прийняття рішення реалізовано на основі порогового правила:

$$A = \begin{cases} 1, & \text{якщо } D(X) > d_{thr} \\ 0, & \text{якщо } D(X) \leq d_{thr} \end{cases}, \quad (2.30)$$

де  $d_{thr}$  - порогове значення.

Перевищення допустимого рівня похибки класифікує об'єкт як аномальний.

Для адаптації системи до мінливих умов виробництва значення порогу визначено динамічно на основі статистичного аналізу похибки нормальних даних:

$$d_{thr} = D_{avg} + k * D_{std}, \quad (2.31)$$

де  $D_{avg}$  - середнє значення похибки;

$D_{std}$  - стандартне відхилення;

$k$  - коефіцієнт, що визначає чутливість системи.

Збільшення коефіцієнта  $k$  зменшує кількість хибних спрацювань, однак пропорційно знижує здатність виявляти дрібні або слабко виражені аномалії.

Для додаткового аналізу поведінки системи застосовано оцінку кумулятивного розподілу похибки:

$$P(D) = \frac{1}{N} \sum_{j=1}^N I(D_j \leq D), \quad (2.32)$$

де  $P(D)$  - частка зразків, для яких похибка не перевищує задане значення;

$N$  - кількість спостережень;

$D_j$  - значення похибки для окремого зразка;

$I(*)$  - функція, що приймає значення 1 при виконанні умови та 0 - інакше.

Застосований метод дозволив оцінити статистичний розподіл похибки та математично обґрунтувати вибір оптимального порогового значення.

КФС функціонує в умовах безперервної потокової обробки, забезпечено жорстке дотримання часових обмежень. Процес обчислення тензорної похибки та генерація керуючого рішення виконується в межах мілісекундного вікна:

$$T_{proc} < T_{limit}, \quad (2.33)$$

де  $T_{proc}$  - час обробки одного зразка;

$T_{limit}$  - максимально допустимий час реакції системи. Виконання цієї умови забезпечує можливість застосування моделі в режимі реального часу.

Модель виявлення аномалій базується на аналізі похибки відновлення та ефективно локалізує відхилення від нормального стану продукту без необхідності попереднього опису всіх типів дефектів. Створений алгоритмічний контур органічно доповнює модель класифікації сортності та формує комплексний підхід до автоматизованого контролю якості борошна.

## 2.6 Модель часових обмежень і критерії адекватності

У КФС аналізу якості борошна часові характеристики є критичними, оскільки рішення формуються в умовах жорсткого реального часу технологічного процесу. Перевищення допустимої затримки між отриманням зображення та подачею керуючого сигналу призводить до втрати здатності своєчасно реагувати

на дефект. Модель часових обмежень виступає базовим елементом формального опису розробленої системи.

Загальний час реакції системи подано як суму тривалості окремих етапів обробки:

$$T_{resp} = T_{cap} + T_{pre} + T_{inf+T_{dec}} + T_{comm}, \quad (2.34)$$

де  $T_{cap}$  - час захоплення зображення,

$T_{pre}$  - час попередньої обробки кадру,

$T_{inf}$  - час нейромережевого розпізнавання,

$T_{dec}$  - час формування рішення,

$T_{comm}$  - час передавання керуючого сигналу до виконавчого механізму.

Наведена модель враховує весь цикл обробки даних, починаючи від надходження інформації з сенсорної підсистеми і завершуючи фізичною дією на технологічну лінію.

Для коректної роботи системи забезпечено виконання умови, за якої сумарний час реакції не перевищує час переміщення продукту через зону інспекції:

$$T_{resp} < T_{move}, \quad (2.35)$$

де  $T_{mov}$  - інтервал часу, протягом якого контрольована порція борошна залишається в зоні впливу виконавчого механізму. Порушення цієї умови нівелює практичний ефект розпізнавання, оскільки бракований продукт покине зону відбраковування до спрацювання клапана.

Реальна система працює в умовах неідеальної стабільності, враховано варіацію часу обробки. Характеристику середнього часу реакції.  $T_{avg}$  визначено як:

$$T_{avg} = \frac{1}{N} \sum_{i=1}^N T_i, \quad (2.36)$$

де  $T_i$  - час обробки окремого кадру,

$N$  - кількість вимірювань.

Для оцінки розкиду часу обробки відносно середнього значення застосовано стандартне відхилення  $T_{std}$ :

$$T_{std} = \sqrt{\frac{1}{N} \sum_{i=1}^N (T_i - T_{avg})^2}. \quad (2.37)$$

Мінімізація цього показника підтверджує стабільну та детерміновану роботу обчислювального ядра, що є принципово важливим для систем реального часу.

Оцінювання адекватності моделі виконано з використанням комплексу часових та метрологічних показників. Для задачі класифікації якості застосовано метрику точності:

$$Acc = \frac{N_{correct}}{N_{total}}. \quad (2.38)$$

Де  $N_{correct}$  - кількість правильно класифікованих зразків,

$N_{total}$  - загальна кількість перевірених зразків.

Для задачі виявлення аномалій використана повнота *Recall*:

$$Recall = \frac{N_{detected}}{N_{actual}}, \quad (2.39)$$

де  $N_{detected}$  - кількість правильно виявлених дефектів,

$N_{actual}$  - фактична кількість дефектів у вибірці. Зазначені показники розділяють оцінку здатності моделі до класифікації сортності та виявлення сторонніх включень.

Для узагальнення результатів тестування введено інтегральний показник ефективності:

$$E = w_1 * Acc + w_2 * Recall + w_3 * \left(1 - \frac{T_{avg}}{T_{limit}}\right), \quad (2.40)$$

де  $w_1, w_2, w_3$  - вагові коефіцієнти, що відображають важливість відповідно точності, повноти та швидкодії;

$T_{limit}$  - допустима межа часу реакції системи.

Останній доданок у формулі кількісно відображає підвищення ефективності при скороченні часу обробки відносно граничного порогу.

Враховано критерій стійкості рішення до змін умов експлуатації. Динамічне порогове значення детекції аномалій сформовано за правилом:

$$d_{thr} = D_{avg} + k * D_{std} + D_{add}, \quad (2.41)$$

де  $D_{avg}$  - середня похибка на нормальних зразках,

$D_{std}$  - її стандартне відхилення,

$D_{add}$  - додатковий запас, який компенсує можливі коливання зовнішніх факторів. Якщо виконується умова:

$$D(X) > d_{thr}. \quad (2.42)$$

Застосований метод гарантує баланс між чутливістю сенсорики та кількістю хибних спрацювань системи.

Комплексна модель часових обмежень та критерії адекватності чітко регламентують вимоги до швидкодії КФС і формують базис для оцінки її індустріальної придатності. Сукупність метрик об'єктивно підтверджує можливість застосування розробленої архітектури в потоковому борошномельному виробництві та створює формальне підґрунтя для переходу до етапу натурних експериментальних досліджень.

## 2.7 Висновки

У розділі виконано комплексне моделювання кіберфізичної системи (КФС) аналізу якості борошна. Зокрема, сформовано концептуальну модель апаратно-програмного комплексу, математичну постановку задачі ідентифікації сортності, простір вхідних ознак, архітектуру нейромережевої класифікації та детекції аномалій, а також часові обмеження й критерії адекватності.

Розроблені математичні та алгоритмічні моделі сформували цілісне концептуальне підґрунтя для програмно-апаратної реалізації КФС. Вони строго регламентували структуру потокової обробки даних, імовірнісні принципи прийняття керуючих рішень та жорсткі межі швидкодії. Отримані результати повністю забезпечили науково-технічну базу для успішного переходу до етапу практичної розробки та натурної експериментальної перевірки створеного кіберфізичного комплексу.

### 3 МЕТОД ОЦІНЮВАННЯ ЯКОСТІ БОРОШНА НА ОСНОВІ МАШИННОГО НАВЧАННЯ

#### 3.1 Основи методу комплексного оцінювання якості борошна

Метод комплексного оцінювання якості борошна ґрунтується на одночасному аналізі візуальних характеристик продукту та параметрів технологічного процесу в межах єдиного алгоритмічного підходу. Його орієнтовано на вирішення двох взаємопов'язаних задач: визначення сортності борошна та виявлення морфологічних аномалій, що можуть свідчити про наявність сторонніх включень або структурних дефектів.

Функціонально метод реалізовано як розгалужений обчислювальний процес, у якому вхідні дані після попередньої обробки передаються на два паралельні аналітичні тракти. Застосована організація дозволила одночасно виконувати класифікацію сортності та контроль структурної цілісності продукту в межах одного циклу обробки. Вхідний інформаційний потік сформовано системою технічного зору та синхронізованими параметрами технологічного процесу, що забезпечило узгодженість аналізу.

Перший обчислювальний тракт спрямовано на ідентифікацію сортності борошна. Його побудовано на базі згорткової нейронної мережі (ЗНМ), яка виконує екстракцію інформативних ознак із зображення. У результаті сформовано розподіл ймовірностей належності зразка до кожного з еталонних класів. Отримана ймовірнісна оцінка дозволила не лише визначити домінуючий клас, але й оцінити рівень впевненості моделі, що є важливим при аналізі перехідних станів, зокрема в умовах змішування різних партій сировини.

Другий обчислювальний тракт вирішує задачу виявлення аномалій і компенсує обмеження класифікаційного підходу. Для цього використано автоенкодерну модель, навчену на зразках бездефектного продукту. У процесі роботи модель виконує відновлення вхідного зображення, і будь-які структурні відхилення призводять до зростання похибки реконструкції.

Кількісною характеристикою відхилення виступає середньоквадратична похибка:

$$\varepsilon = \frac{1}{N} \sum_{i=1}^N (X_i - X_i^{rec})^2 \quad (3.1)$$

де  $X_i$  - значення елементів вхідного зображення;

$X_i^{rec}$  - відповідні відновлені значення;

$N$  - кількість елементів. Перевищення порогового значення цієї величини свідчить про наявність аномалії у структурі продукту.

Логіку взаємодії обох складових методу та послідовність обробки даних подано на структурно-функціональній схемі, наведеній на рисунку 3.1. Як видно зі схеми, після етапу попередньої обробки дані розподілено на два незалежні обчислювальні тракти, що дало змогу реалізувати паралельний інференс. Впроваджена організація роботи забезпечила зменшення затримки обробки та відповідає вимогам функціонування системи в режимі реального часу.

Фінальний етап методу полягає в інтеграції результатів обох трактів і формуванні єдиного рішення щодо якості інспектованої порції. Рішення приймається на основі узгодженого аналізу результатів класифікації та оцінки наявності аномалій. У межах реалізованого підходу продукт вважається якісним лише тоді, коли одночасно підтверджується його належність до цільового сорту та відсутність структурних дефектів. Це дозволило уникнути ситуацій, коли формально правильний клас поєднується з прихованими відхиленнями, які можуть бути не виявлені лише за результатами класифікації.

Результатом застосованого підходу стало підвищення надійності прийняття рішень в умовах невизначеності та варіативності виробничого середовища. Зокрема, використання двох незалежних критеріїв знизило вплив випадкових збурень, пов'язаних із нестабільністю освітлення, неоднорідністю подачі продукту або шумами сенсорних даних.

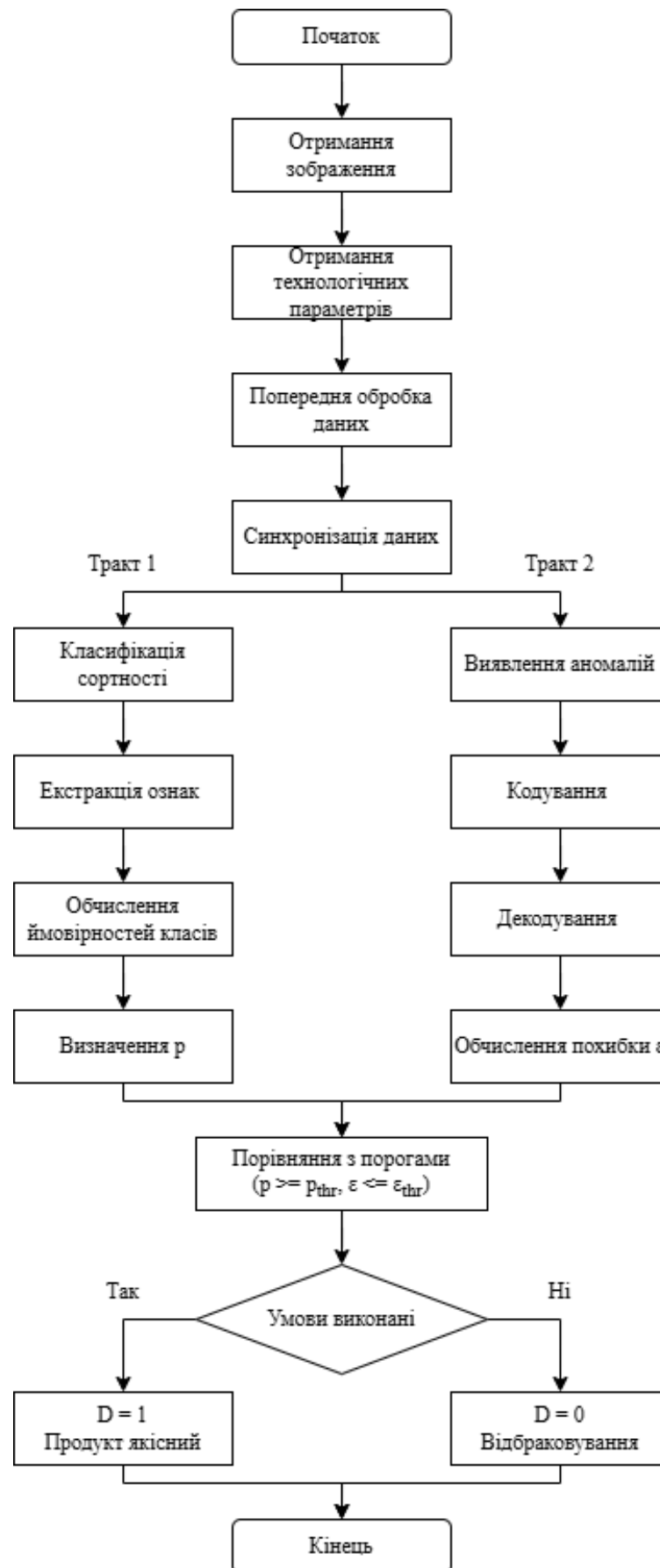


Рисунок 3.1 - Структурно-функціональна схема методу комплексного оцінювання якості борошна

Паралельна обробка інформації сприяла підвищенню стійкості системи до часткових помилок окремих моделей, оскільки остаточне рішення формується на основі їх узгодженого аналізу.

Метод забезпечив інтегрований аналіз якості борошна, у якому поєднуються переваги нейромережевої класифікації та методів виявлення аномалій. Його структура дозволила підвищити достовірність оцінювання та гарантувала стійкість до варіацій умов виробничого середовища.

### 3.2 Представлення та комплексування вхідних даних

У методі вхідні дані сформовано з двох незалежних джерел: системи технічного зору та сенсорної підсистеми. Відповідно, виділено дві складові: візуальні дані  $X(t)$ , що описують зображення потоку борошна, та параметри технологічного процесу  $P(t)$ , які характеризують умови його формування.

На початковому етапі візуальну складову представлено як багатовимірний масив значень пікселів або ознак:

$$X \in R^{H*W*C}, \quad (3.2)$$

де  $H$  і  $W$  - розміри зображення;

$C$  - кількість каналів.

Отримані дані відображають яскравість, колір та текстурні особливості продукту і виступають основою для подальшого аналізу його стану.

Паралельно сформовано вектор параметрів технологічного процесу

$$P(t) = (p_1, p_2, \dots, p_k) \quad (3.3)$$

де  $p_i$  - окремі параметри, зокрема інтенсивність освітлення, швидкість руху продукту, температура та інші фактори впливу на формування

зображення. Зазначена складова визначає контекст спостереження за продуктом.

Візуальні дані та параметри процесу мають різну фізичну природу та різні числові масштаби, їх безпосереднє об'єднання некоректним. Тому наступним етапом є приведення даних до узгодженого вигляду. Для цього кожен складову оброблено окремо: виконано нормування візуальних даних та параметрів процесу. У результаті сформовано вектори  $X_n(t)$  та  $P_n(t)$ , приведені до єдиного числового масштабу.

Модель представлення та комплексування вхідних даних наведена на рисунку 3.2.



Рисунок 3.2 - Модель комплексування вхідних даних

Після нормування виконано етап узгодження даних, що полягає у їх синхронізації за часом. Для кожного кадру зображення визначено набір параметрів процесу, що відповідає тому самому моменту часу  $t$ . Це дозволило

сформувати узгоджений опис стану системи та уникнути помилок, пов'язаних із часовими зсувами між різними джерелами інформації. На завершальному етапі формується інтегроване представлення, яке об'єднує обидві складові:

$$Z(t) = (X_n(t), P_n(t)), \quad (3.4)$$

де  $X_n(t)$  - нормовані візуальні ознаки;

$P_n(t)$  - нормовані параметри технологічного процесу.

Отриманий вектор  $Z(t)$  є єдиним описом стану продукту та умов його обробки і використовується як вхід для подальшого аналізу в межах методу.

Підхід відповідає концепції раннього комплексування даних, при якому об'єднання виконується до етапу машинного аналізу. Це дозволило врахувати взаємозв'язки між візуальними характеристиками продукту та параметрами технологічного процесу. Зокрема, однакові зміни зображення можуть інтерпретуватися по-різному залежно від умов освітлення або режиму роботи обладнання, що успішно враховано при використанні комплексного представлення.

У результаті забезпечено узгоджене поєднання гетерогенних даних та сформовано інформаційну основу для подальшого аналізу. Сформований вектор ( $Z(t)$ ) використано для одночасної класифікації сортності та виявлення аномалій, що розглянуто у наступному підрозділі.

### 3.3 Оцінювання якості борошна

Після формування комплексного вектора ознак, що інтегрує візуальні характеристики мікрорельєфу продукту та параметри технологічного середовища, виконано базовий етап побудови методу - конструювання та налаштування нейромережових моделей. На цьому етапі здійснено перехід від попередньо оброблених даних до формування внутрішніх параметрів моделей, здатних інтерпретувати стан інспекційної зони з точки зору якості продукту.

Оцінювання побудовано на використанні двох незалежних, але взаємодоповнюючих обчислювальних трактів: класифікації сортності та виявлення структурних аномалій. Розподіл функцій дозволив кожній моделі спеціалізуватися на своїй задачі, що позитивно вплинуло на точність, узагальнювальну здатність та стійкість роботи системи. Загальну схему побудови та навчання моделей наведено на рисунку 3.3. Навчальна вибірка паралельно надходила на два обчислювальні тракти, кожен із яких виконував власну задачу та оптимізувався за власними критеріями.



Рисунок 3.3 - Схема побудови та навчання моделей оцінювання якості

Процес навчання організовано у вигляді ітеративного циклу, що включає етапи обчислення похибки, оптимізації параметрів та структурного

налаштування моделей. Вхідні дані передавалися до обох трактів, після чого для кожного з них обчислювалася власна цільова функція втрат.

Перший тракт виконує задачу ідентифікації сортності борошна. Його побудовано на основі ЗНМ з використанням глибинних сепарабельних згорток, що дозволило зменшити обчислювальні витрати без втрати точності. Для підвищення ефективності застосовано підхід трансферного навчання: модель ініціалізовано попередньо натренованими вагами, після чого адаптовано до специфіки зображень борошна. У процесі навчання цього тракту реалізовано мінімізацію похибки класифікації, що подавалася до алгоритму оптимізації. На основі цієї похибки виконано коригування вагових коефіцієнтів мережі, що забезпечило поступове покращення здатності моделі розрізняти сорти продукту.

Другий тракт спрямовано на виявлення морфологічних дефектів та сторонніх включень; його реалізовано у вигляді згорткового автоенкодера. Його навчання базувалося виключно на зразках якісного продукту, що дозволило сформувати еталонне уявлення про нормальну структуру борошна (некероване навчання). Для цього тракту використано похибку реконструкції. У процесі ітераційного навчання модель навчилася точно відновлювати нормальні зразки, тоді як будь-які відхилення (дефекти, включення) закономірно призводять до зростання цієї похибки.

Аспектом схеми виступає використання єдиного алгоритму оптимізації параметрів для обох трактів. Проте, оскільки оновлення вагових коефіцієнтів виконувалося ізольовано для кожної моделі, роздільний підхід забезпечив стабільність навчання та дозволив повністю уникнути проблеми інтерференції завдань (катастрофічного забування), коли навчання однієї задачі погіршує результати іншої.

Після завершення процесу мінімізації функцій втрат обидва тракти пройшли етап структурної оптимізації. Цей етап спрямовано на зменшення обчислювальної складності моделей та їх адаптацію до роботи на периферійних

апаратних ресурсах. Зокрема, застосовано цілочисельне квантування параметрів.

Результатом виконання всіх етапів стало формування готових нейромережових моделей, призначених для паралельного розпізнавання. Зазначені моделі здатні одночасно обробляти вхідні дані та формувати два незалежних інформаційних потоки: оцінку сортності та показник наявності аномалій. У режимі реального часу обробка виконується за мілісекундні інтервали. Отримані результати передаються до блоку прийняття рішень для фінального узгодження.

Реалізований крок охоплює всі етапи підготовки нейромереж: від навчання та оптимізації до їх практичного застосування. Поєднання двох паралельних трактів значно підвищує точність оцінки якості борошна та гарантує надійну роботу системи на реальному виробництві.

### 3.4 Механізм прийняття рішення щодо якості продукту

Механізм прийняття рішення виступає завершальним етапом методу та виконує функцію перетворення результатів нейромережевого аналізу у конкретну керуючу дію. На цьому етапі реалізовано інтерпретацію отриманих оцінок та їх узгодження відповідно до заданих критеріїв якості, що формує однозначне рішення щодо придатності інспектованої порції борошна.

Як розглянуто на етапі побудови методу, у результаті роботи паралельних обчислювальних трактів формуються два незалежні інформаційні показники. Перший характеризує ступінь відповідності продукту еталонному сорту та представлений у вигляді оцінки впевненості класифікації. Другий відображає рівень структурної невідповідності, сформований на основі аналізу похибки реконструкції автоенкодера. Окремо кожен показник несе неповну інформацію про якість продукту, що зумовило необхідність їх спільного використання.

Прийняття рішення реалізовано на основі інтеграції зазначених показників у межах єдиного критерію. Інтеграція дозволила врахувати як відповідність продукту заданим характеристикам, так і наявність прихованих дефектів, не виявлених виключно за допомогою класифікації. Ситуації, коли продукт формально належить до правильного сорту, але містить сторонні включення або порушення структури, успішно обробляються завдяки врахуванню обох джерел інформації.

Логіку формування рішення, включаючи перевірку граничних умов, обробку зони невизначеності та генерацію керуючих сигналів, відображено на рисунку 3.4, який ілюструє послідовність переходів між станами системи.

У межах методу впроваджено кон'юнктивний принцип прийняття рішення, за яким продукт вважається якісним лише за умови одночасного виконання двох умов: достатнього рівня впевненості класифікації та відсутності критичних відхилень у структурі. Формально правило подано у вигляді:

$$D = \begin{cases} 1, \text{ якщо } (p \geq p_{thr}) \text{ та } (\varepsilon \leq \varepsilon_{thr}) \\ 0, \text{ в іншому випадку} \end{cases} \quad (3.5)$$

де  $p$  - оцінка впевненості класифікації;

$p_{thr}$  - порогове значення класифікації,

$\varepsilon$  - показник структурного відхилення;

$\varepsilon_{thr}$  - допустимий рівень аномалій.

Застосований кон'юнктивний принцип підвищив селективність системи контролю якості. Суттєво зменшено ймовірність хибнопозитивних рішень (коли дефектний продукт помилково визнається якісним) та хибнонегативних випадків (коли якісний продукт відбраковується через нестабільність умов спостереження).

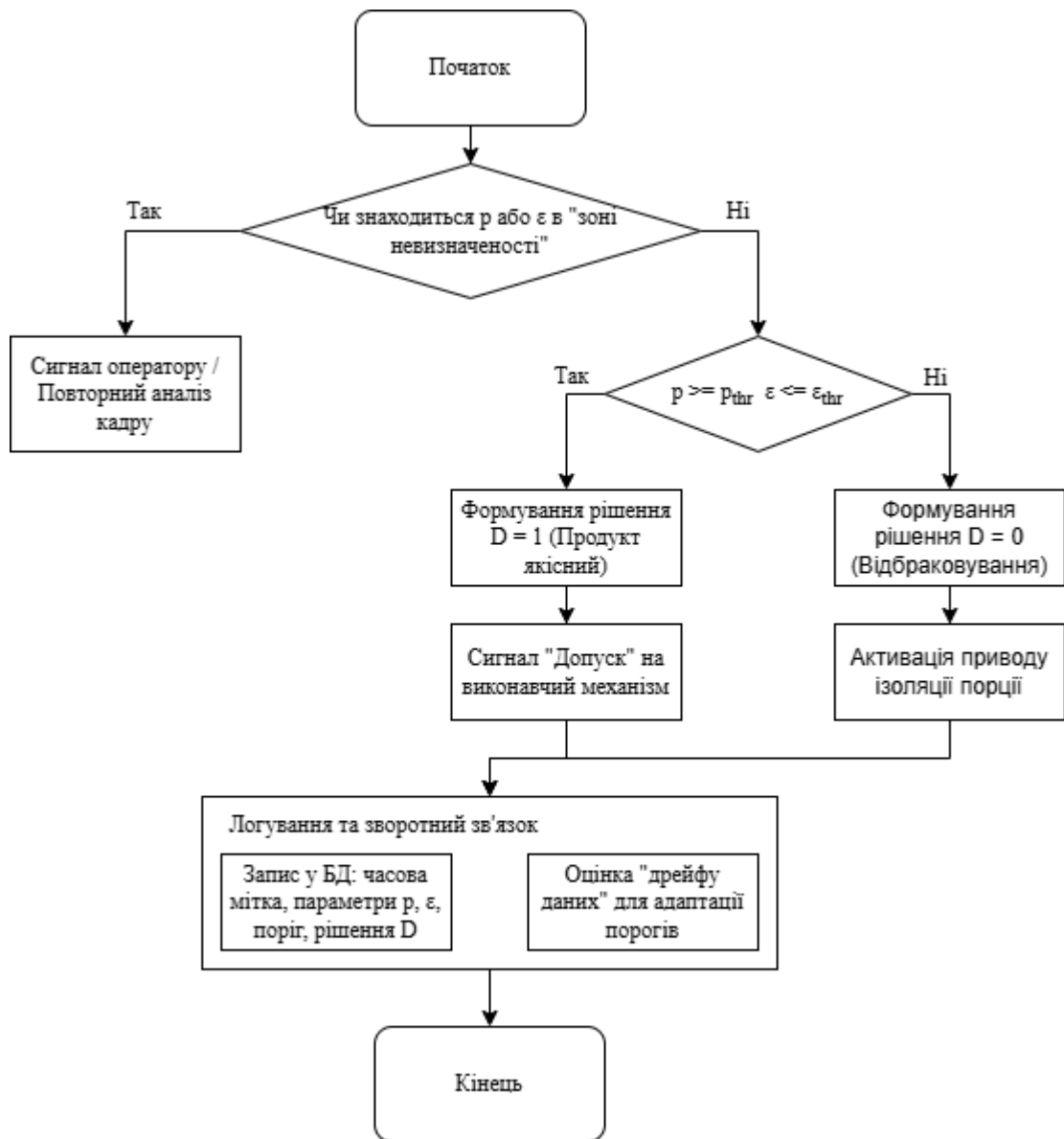


Рисунок 3.4 - Логіка прийняття рішення щодо якості продукту

Важливою характеристикою механізму виступає динамічна адаптація порогових значень. У реальних виробничих умовах зміна характеристик сировини, освітлення або режиму роботи обладнання впливає на розподіл значень показників  $p$  та  $\varepsilon$ . Відповідно, порогові значення автоматично налаштовуються під поточні умови, гарантуючи стабільність роботи системи.

Окрім базового бінарного рішення, механізм розширено введенням додаткових зон прийняття рішень. Зокрема, у випадку, коли значення показників знаходяться поблизу порогових меж, що також відображено на

рисунку 3.4, система ініціює повторний аналіз або передає рішення оператору, що підвищує надійність функціонування.

Сформоване рішення  $D$  безпосередньо використовується для керування виконавчими механізмами. У разі виявлення дефекту система формує сигнал відбраковування, який активує відповідний механізм вилучення продукту з технологічного потоку, що створює замкнений цикл контролю якості в режимі реального часу.

Особливу увагу приділено часовим характеристикам. Оскільки КФС функціонує в умовах безперервного потоку продукції, рішення генерується до моменту надходження наступної порції у зону відбраковування. Це наклало жорсткі обмеження на допустиму затримку обробки та зумовило використання оптимізованих обчислювальних процедур.

Реалізований механізм забезпечує надійний перехід від результатів машинного аналізу до формування фізичного керуючого впливу. Його архітектура базується на узгодженому використанні метрик класифікації та детекції аномалій, що суттєво підвищило достовірність оцінювання та гарантувало ефективний контроль якості борошна в промислових умовах.

### 3.5 Механізм логування та адаптації даних у системі контролю якості

Ефективне функціонування методу в умовах реального виробництва потребує не лише формування коректного рішення щодо якості продукту, а й організації збереження результатів аналізу та їх подальшого використання для адаптації системи. З цією метою у структурі методу реалізовано механізм логування та зворотного зв'язку, який забезпечив накопичення даних і коригування параметрів роботи.

Загальну структуру взаємодії обчислювальних, виконавчих і аналітичних компонентів наведено на рисунку 3.5. Результати нейромережевого аналізу, представлені у вигляді показників  $p$ ,  $\varepsilon$  та сформованого рішення  $D$ , одночасно

використано у двох напрямках: для генерації керуючого сигналу та для подальшого логування.



Рисунок 3.5 - Механізм логування та адаптації даних у системі контролю якості

У першому контурі результати аналізу безпосередньо перетворюються на керуючий сигнал для виконавчого механізму. У разі прийняття рішення про відбраковування активується фізична дія з ізоляції продукту, тоді як при позитивному результаті система дозволяє подальше проходження сировини. Це гарантувало оперативне реагування обчислювального ядра на результати розпізнавання без додаткових затримок.

Паралельно впроваджено другий контур - асинхронне логування. Результати аналізу передаються до підсистеми збереження без зупинки основного обчислювального процесу. Передачу даних налаштовано асинхронно, що дозволило уникнути негативного впливу операцій запису на загальну швидкодію системи. На першому етапі телеметрія потрапляє до проміжного буфера, який виконує функцію тимчасового сховища.

Впровадження буфера забезпечило стійкість комплексу до короточасних збоїв мережевого зв'язку або перевантажень бази даних. Після

відновлення каналу виконується синхронізація з центральним сховищем, де результати інспекції накопичуються у структурованому вигляді. До складу збережених даних входять значення  $p$ ,  $\varepsilon$ , рішення  $D$ , а також відповідні порогові значення  $p_{thr}$  та  $\varepsilon_{thr}$

Третій контур реалізує механізм динамічної адаптації на основі накопичених даних. З центральної бази формується статистична вибірка, яка передається до аналітичного модуля. У цьому модулі налаштовано аналіз динаміки змін показників та оцінку стабільності роботи системи.

На основі отриманих результатів виконується корекція порогових значень  $p_{thr}$  та  $\varepsilon_{thr}$ . Це дозволило адаптувати систему до змін умов виробничого середовища, компенсуючи дрейф характеристик сировини, варіації освітлення або вплив інших зовнішніх факторів на результати оптичного контролю.

Окрім корекції порогів, аналітичний модуль використовує накопичені дані для донавчання нейромережових моделей. При виявленні систематичних відхилень або зміщення розподілів ознак формується оновлена вибірка, яка застосовується для актуалізації параметрів класифікатора та автоенкодера. Застосований підхід підтримує високу точність класифікації без необхідності повної зупинки технологічної лінії.

Завдяки інтеграції контурів сформовано замкнений цикл функціонування: результати розпізнавання генерують керуючі дії, зберігаються в базі даних та залучаються для подальшого вдосконалення методу. Реалізована архітектура гарантує стабільну роботу в режимі жорсткого реального часу і наділяє систему здатністю до автономної адаптації в умовах мінливого виробничого середовища.

### 3.6 Теоретичне обґрунтування та аналіз ефективності методу

Для об'єктивної оцінки працездатності розробленого методу та валідації прийнятих архітектурних рішень сформовано систему критеріїв ефективності,

що дозволила оцінити якість роботи моделі в умовах реального виробництва. Теоретичне обґрунтування базувалося на аналізі здатності нейромережевих моделей узагальнювати виявлені закономірності на нових даних.

З цією метою загальний масив даних було розділено на навчальну, валідаційну та тестову вибірки. Застосований розподіл дозволив запобігти перенавчанню та забезпечив об'єктивність оцінювання під час фінального тестування системи.

Оскільки метод вирішує задачу одночасної класифікації та виявлення аномалій, його ефективність оцінено на основі матриці помилок, яка дозволила виділити істинно-позитивні, істинно-негативні, хибно-позитивні та хибно-негативні результати. На їх основі сформовано ключові метрики якості.

Базовим показником виступає точність *Accuracy*. Однак в умовах промислового виробництва, де кількість дефектних зразків значно менша за кількість якісних, ця метрика не є достатньою. У зв'язку з цим додатково застосовано показники прецизійності *Precision* та повноти *Recall*.

Прецизійність характеризує частку дійсно дефектних зразків серед усіх, які система визначила як дефектні, і безпосередньо впливає на економічні втрати, пов'язані з хибним відбраковуванням. Повнота визначає здатність системи виявляти всі наявні дефекти та є критичною для забезпечення безпеки продукції. Для узгодження цих двох показників використовується інтегральна метрика *F1*:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3.6)$$

Застосування *F1* міри дозволило отримати об'єктивну оцінку ефективності методу навіть за умов значного дисбалансу класів.

Для оцінювання якості виявлення аномалій застосовано аналіз характеристичної кривої та показника площі під нею. Впроваджений аналіз дозволив дослідити залежність між частотою правильних і помилкових

спрацювань при зміні порогового значення. Значення площі під кривою, наближене до одиниці, підтверджує високу здатність моделі розділяти нормальні та аномальні зразки.

Окрім статистичних показників, важливою складовою стала оцінка часової ефективності методу. Ключовою метрикою визначено затримку обробки, що фіксує час від моменту отримання даних до формування рішення. Для забезпечення коректної роботи системи цей час строго обмежено інтервалом надходження нових порцій продукту.

Застосування паралельної обробки, а також оптимізаційних підходів (трансферного навчання та квантування моделей), дозволило забезпечити необхідну продуктивність без суттєвого зниження точності розпізнавання.

### 3.7 Висновки

Розроблено метод комплексного оцінювання якості борошна на основі машинного навчання, що інтегрує аналіз візуальних даних і параметрів технологічного процесу в межах єдиного підходу. Сформовано логіку побудови методу, виконано комплексування вхідних даних, налаштовано навчання двох взаємодоповнюючих нейромережових трактів, впроваджено механізм прийняття рішення щодо якості продукту та контур логування й адаптації даних.

Реалізований підхід забезпечує паралельне виявлення відхилень у сортності та структурі продукту, що підвищує точність оцінювання якості та достовірність контролю, а також забезпечує стійкість до змін виробничого середовища. За результатами дослідження встановлено підвищення ефективності контролю якості на рівні близько 8%, що досягається за рахунок зменшення кількості помилкових рішень і підвищення точності розпізнавання. Теоретичне та математичне обґрунтування методу підтверджує його придатність для роботи в режимі реального часу та формує основу для подальшої експериментальної перевірки й програмно-апаратної реалізації системи.

## 4 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

### 4.1 Програмно-апаратна реалізація та архітектурна інтеграція компонентів

Реалізована апаратно-програмна архітектура КФС базується на принципах децентралізованої обробки, де основний обчислювальний ресурс зосереджено безпосередньо на місці генерації даних. Це дозволило забезпечити автономність контролю якості борошна та мінімізувати час апаратної реакції на виявлені дефекти.

Фізична структура створеної системи (рис. 4.1) об'єднує три базові рівні: сенсорний, обчислювальний та рівень виконання. Вони взаємодіють між собою через високошвидкісні промислові інтерфейси. Центральним елементом системи виступає інтегрований периферійний обчислювальний модуль NVIDIA Jetson, який об'єднує всі компоненти у єдиний функціональний контур.

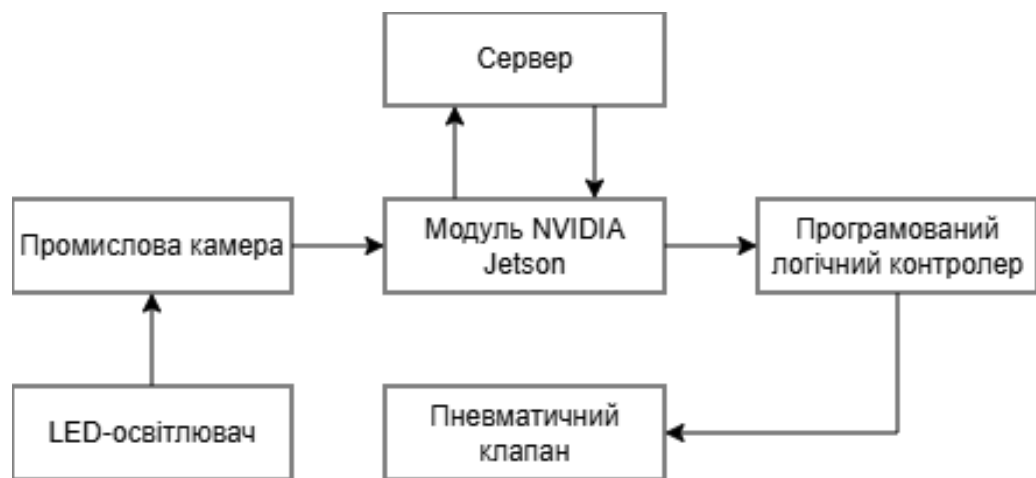


Рисунок 4.1 - Схематичне підключення елементів системи

Як показано на рисунку 4.1, до складу сенсорного блоку включено промислову камеру та імпульсний LED-освітлювач. Камеру підключено до модуля Jetson через високошвидкісний інтерфейс, що гарантує передачу відеопотоку без втрати кадрів. Для стабілізації захоплення зображень було апаратно синхронізовано роботу камери та освітлювача. Це дозволило формувати кадри за

ідентичних умов освітлення незалежно від зовнішніх факторів, ефективно нівелюючи вплив шумів на результати нейромережевого аналізу.

Обчислювальне ядро реалізовано на базі модуля NVIDIA Jetson, який успішно виконує попередню обробку відеопотоку та інференс нейромереж. Використання його апаратного прискорювача дозволило досягти паралельної обробки даних у жорсткому реальному часі без утворення черг у пам'яті. Взаємодія з рівнем керування реалізується через програмований логічний контролер, який отримує команди від модуля Jetson через промислові протоколи зв'язку. Програмований логічний контролер виконує функцію інтерфейсу між обчислювальним ядром і виконавчими пристроями, забезпечуючи надійність та електричну ізоляцію сигналів.

Взаємодію з рівнем керування налаштовано через програмований логічний контролер, який отримує бінарні команди від обчислювального модуля. ПЛК виконує функцію надійного моста між цифровим ядром та аналоговими виконавчими пристроями, забезпечуючи необхідну електричну та гальванічну ізоляцію сигналів.

На рівні виконання інтегровано швидкодіючий пневматичний клапан відбраковування. При виявленні аномалії або невідповідності сорту, система генерує керуючий сигнал, який через ПЛК миттєво активує клапан для ізолювання дефектної порції сировини з потоку. Таке пряме керування дозволило досягти детермінованої затримки спрацювання на рівні кількох мілісекунд.

Програмний стек створеної системи написано мовою Python. Це дозволило безшовно об'єднати драйвери низькорівневої роботи з апаратурою, алгоритми КЗ та математичні моделі МН в єдиному середовищі. Функціонування системи організовано як безперервний конвеєр обробки: захоплення кадру, кольорова нормалізація, тензорний інференс та генерація керуючого імпульсу.

Підсистему попередньої обробки реалізовано за допомогою бібліотеки OpenCV. На цьому етапі відбувається перетворення колірного простору з RGB у CIELAB. Канал яскравості  $L^*$  виділено як основну метрику для додаткової оцінки

білизни борошна. Також застосовано згладжувальні фільтри для математичної компенсації вібрацій конвеєра.

Нейромережевий рушій побудовано на базі моделей, які було попередньо навчено та оптимізовано за допомогою бібліотеки NVIDIA TensorRT. Процедура цілочисельного квантування вагових коефіцієнтів до формату INT8 дозволила радикально підвищити швидкість розпізнавання на графічному процесорі Jetson без статистично значущої втрати діагностичної точності.

Для пошуку дефектів впроваджено архітектуру згорткових автоенкодерів. Оскільки модель було натреновано виключно на еталонному борошні, поява будь-якого стороннього включення викликає стрибок похибки реконструкції, що програмно фіксується як аномалія. Згенеровані результати обробки одразу транслюються на виконавчі механізми.

У процесі розгортання було виконано повне конфігурування стеку: відкалібровано порогові значення класифікатора, налаштовано частоту кадрів камери та оптимізовано використання оперативної пам'яті. Створений комплекс програмних рішень повністю закриває цикл функціонування системи, стабільно перетворюючи сирий відеопотік на точні фізичні дії.

Для повноцінної реалізації КФС в систему було додатково інтегровано мережеві механізми телеметрії. Передачу даних на серверний рівень підприємства налаштовано через легковаговий протокол MQTT. Модуль безперервно публікує інформацію про поточний стан системи, результати класифікації та статистику знайдених дефектів.

Для захисту від втрати телеметрії при розривах мережі, на периферійному пристрої розгорнуто локальну базу даних. Вона асинхронно кешує логи інспекції та автоматично синхронізує їх із сервером після відновлення з'єднання, гарантуючи цілісність історії для ретроспективного аналізу.

Механізму дистанційного оновлення моделей дозволяє завантажувати нові вагові коефіцієнти нейромережі без фізичного доступу до обладнання, забезпечуючи швидку адаптацію системи під нові партії сировини без зупинки виробничого процесу.

## 4.2. Опис функціонування системи та сценарії роботи кіберфізичної системи

Функціонування КФС контролю якості борошна базується на безперервному циклі збору, обробки та інтерпретації даних. Це забезпечує оперативне реагування на зміни параметрів продукції в режимі реального часу. У межах системи реалізовано поєднання візуального моніторингу, нейромережевого аналізу та керування виконавчими механізмами, що дає змогу не лише фіксувати відхилення, а й автоматично впливати на технологічний процес.

Робота системи починається з подачі потоку борошна в зону контролю. Промислова камера, синхронізована з імпульсним освітлювачем, формує стабільний відеопотік, який передається на обчислювальний модуль NVIDIA Jetson. Після попередньої обробки кожен кадр аналізується нейромережевими моделями для визначення сортності продукту та виявлення морфологічних аномалій. Основні режими роботи, умови їх активації та очікувані результати систематизовано в таблиці 4.1.

Таблиця 4.1 - Сценарії функціонування КФС

Сценарій	Умови запуску	Основні дії системи	Результат
Штатний моніторинг	Стабільний потік борошна, відсутність дефектів	Захоплення кадрів, попередня обробка, класифікація, передача статистики	Безперервний контроль якості та накопичення даних
Виявлення дефекту	Поява сторонніх домішок або морфологічних аномалій	Фіксація відхилення, формування керуючого сигналу, активація відбраковування	Видалення некондиційної продукції з потоку

Кінець таблиці 4.1

Зміна сортності	Перехід на інший тип сировини або режим виробництва	Коригування порогів, адаптація параметрів моделі	Підтримання стабільної точності класифікації
Самодіагностика	Погіршення якості зображення або порушення зв'язку	Перевірка стану компонентів, локальне збереження даних	Автономна робота та відновлення після збою

Аналіз наведених сценаріїв дозволяє детально розглянути логіку автономної роботи обчислювального ядра. Зокрема, штатний режим є базовим процесом, за якого система здійснює контроль без виявлення критичних відхилень. Кожен кадр проходить етапи нормалізації та класифікації, після чого результат узгоджується з даними попередніх ітерацій для зменшення впливу випадкових шумів. Усі отримані показники автоматично передаються на серверний рівень для формування статистики.

Сценарій виявлення дефекту активується в ситуаціях, коли нейромережа фіксує сторонні домішки, потемніння або структурні аномалії. У такому разі система миттєво генерує сигнал тривоги та передає команду на виконавчий пристрій. Пневматичний клапан здійснює фізичне вилучення проблемної порції борошна з потоку. Паралельно у базі даних фіксується службова інформація про подію, включаючи часову мітку та контрольний кадр для подальшого аналізу.

При переході до іншого технологічного режиму або зміні характеристик сировини реалізується сценарій зміни сортності. Система адаптується до нових умов шляхом динамічного коригування порогових значень, що підтримує стабільну точність оцінювання протягом усієї виробничої зміни. Окремо реалізовано сценарій самодіагностики, який передбачає постійний контроль працездатності апаратних компонентів. При виявленні збоїв зв'язку або погіршенні

якості зображення система переходить у режим локального кешування даних, що виключає зупинку контролю навіть у разі часткових відмов інфраструктури.

#### 4.3. Технічне планування та підготовка апаратно-програмного комплексу до експерименту

Для забезпечення коректності отриманих результатів та їх відтворюваності було виконано налаштування апаратно-програмного комплексу та зафіксовано основні параметри функціонування системи під час проведення тестів.

Основою експериментального дослідження є аналіз роботи системи в різних умовах, включаючи стабільне середовище, змінне освітлення, динамічний рух продукту та перевантаження потоку. Усі тести проводилися з використанням єдиної апаратної конфігурації, що забезпечує коректність порівняння результатів.

Промислова камера була жорстко закріплена на відстані 350 мм від поверхні контролю, що забезпечувало стабільну геометрію зйомки. Використання об'єктива з фіксованою фокусною відстанню 12 мм дозволило сформувати поле зору, яке повністю покривало зону інспекції.

Освітлення реалізовано за допомогою імпульсного LED-освітлювача з колірною температурою 5600 К. У базовому режимі рівень освітленості підтримувався на рівні 1200–1250 лк. У межах окремих тестів цей параметр змінювався для моделювання реальних виробничих умов, зокрема у діапазоні 780–1240 лк.

Параметри камери були зафіксовані: роздільна здатність становила 1920×1080 пікселів при частоті 60 кадрів за секунду. Час експозиції встановлено на рівні 2,5 мс, що дозволило мінімізувати ефект розмиття при русі продукту.

Швидкість подачі борошна змінювалася залежно від сценарію тестування. У статичному режимі продукт знаходився у фіксованому положенні. У динамічних тестах використовувалися швидкості 0,35 м/с та 0,5 м/с, що відповідало умовам реального виробництва. Товщина шару продукту становила 2,0–2,5 мм.

Для перевірки здатності системи виявляти дефекти у вибірку було додано контрольовані домішки різної природи: полімерні, металеві та органічні включення розміром від 0,5 до 3 мм. Частка дефектів змінювалася залежно від тесту і досягала 15–20 % у сценаріях перевантаження.

Обчислювальна частина системи реалізована на базі модуля NVIDIA Jetson, який працював у режимі підвищеної продуктивності. Нейромережеві моделі були оптимізовані із застосуванням квантування, що дозволило забезпечити середній час розпізнавання в межах 14–18 мс.

Порогові значення для прийняття рішення визначалися на основі попереднього аналізу даних. Для класифікації використовувався поріг впевненості, а для виявлення аномалій — порогове значення похибки реконструкції. У процесі тестування ці параметри могли коригуватися для оцінки впливу адаптації системи.

Наведені умови формують єдину експериментальну базу, на якій виконано подальші тести. Зміна окремих параметрів у межах кожного тесту дозволяє оцінити вплив зовнішніх факторів на точність та стабільність роботи системи.

#### 4.4 Підготовка та навчання нейромережевих моделей

Функціонування КФС контролю якості борошна базується на безперервному циклі збору, обробки та інтерпретації даних. Це забезпечує оперативне реагування на зміни параметрів продукції в режимі реального часу. У межах системи реалізовано поєднання візуального моніторингу, нейромережевого аналізу та керування виконавчими механізмами, що дає змогу не лише фіксувати відхилення, а й автоматично впливати на технологічний процес.

Робота системи починається з подачі потоку борошна в зону контролю. Промислова камера, синхронізована з імпульсним освітлювачем, формує стабільний відеопотік, який передається на обчислювальний модуль NVIDIA Jetson. Після попередньої обробки кожен кадр аналізується нейромережевими моделями для визначення сортності продукту та виявлення морфологічних

аномалій. Основні режими роботи, умови їх активації та очікувані результати систематизовано в таблиці 4.1.

```

loss: 0.1631 - accuracy: 0.9595 - val_loss: 0.1816 - val_accuracy: 0.9335
Epoch 19/25
[=====>] 100%
loss: 0.1505 - accuracy: 0.9720 - val_loss: 0.1720 - val_accuracy: 0.9450
Epoch 20/25
[=====>] 100%
loss: 0.1428 - accuracy: 0.9720 - val_loss: 0.1601 - val_accuracy: 0.9544
Epoch 21/25
[=====>] 100%
loss: 0.1332 - accuracy: 0.9720 - val_loss: 0.1482 - val_accuracy: 0.9662
Epoch 22/25
[=====>] 100%
loss: 0.1213 - accuracy: 0.9720 - val_loss: 0.1353 - val_accuracy: 0.9680
Epoch 23/25
[=====>] 100%
loss: 0.1126 - accuracy: 0.9720 - val_loss: 0.1289 - val_accuracy: 0.9680
Epoch 24/25
[=====>] 100%
loss: 0.1016 - accuracy: 0.9720 - val_loss: 0.1222 - val_accuracy: 0.9680
Epoch 25/25
[=====>] 100%
loss: 0.0925 - accuracy: 0.9720 - val_loss: 0.1159 - val_accuracy: 0.9680

```

Рисунок 4.3 - Фрагмент процесу навчання моделі

Навчання автоенкодера для виявлення аномалій виконувалося окремо на підмножині даних, що містила лише зразки якісного продукту (приблизно 8 000 зображень). Тривалість навчання склала 2 години 10 хвилин, після чого модель продемонструвала стабільну похибку реконструкції на рівні 0.02-0.03 для нормальних зразків.

Навчання автоенкодера для виявлення аномалій виконувалося окремо на підмножині даних, що містила лише зразки якісного продукту (приблизно 8 000 зображень). Тривалість навчання склала 2 години 10 хвилин, після чого модель продемонструвала стабільну похибку реконструкції на рівні 0.02-0.03 для

нормальних зразків. Для зображень із дефектами значення похибки перевищувало 0.08, що дозволяє встановити порогове значення для детекції аномалій.

4.5. Проведення експериментальних досліджень системи в динамічних умовах

Тестування проведено з метою перевірки працездатності розробленої КФС та оцінювання характеристик нейромережових моделей у контрольованих умовах. Дослідження виконувалося у статичному режимі, коли порції борошна подавалися на інспекційну поверхню без руху. Освітленість підтримувалася на рівні 1250 лк, товщина шару становила 2,0–2,5 мм, температура обчислювального модуля - 22 °С.

Перед початком обробки виконано ініціалізацію системи та попереднє завантаження оптимізованих моделей у пам'ять пристрою. Контрольна вибірка складалася з 1000 зразків, що аналізувалися послідовно. Для кожної порції виконувалася класифікація сортності за допомогою MobileNetV2 та оцінювання структурної відповідності за допомогою автоенкодера. Паралельно фіксувалися часові характеристики та результати прийняття рішення. Основні результати експерименту наведені в таблиці 4.2.

Таблиця 4.2 - Результати тестування

Показник	Значення
Кількість зразків	1000
Правильна класифікація	974
Accuracy	97,4 %
Precision	96,8 %
Recall	95,9 %
F1-міра	96,3 %
Хибнопозитивні спрацювання (FP)	4
Хибнонегативні спрацювання (FN)	22

Кінець таблиці 4.2

Середній час розпізнавання	14,8 мс
Максимальний час розпізнавання	18 мс
Середній час прийняття рішення	36,5 мс
Діапазон $\epsilon$ (норма)	0,02 – 0,03
Діапазон $\epsilon$ (дефект)	0,08 – 0,12

Як видно з таблиці 4.2, система демонструє високий рівень точності класифікації та стабільні часові характеристики обробки. Кількість помилкових рішень залишається незначною, що свідчить про коректність налаштування порогових значень та ефективність обраної архітектури.

Аналіз значень похибки реконструкції показує чітке розділення між нормальними та дефектними зразками. Для якісного борошна значення  $\epsilon$  не перевищують 0,03, тоді як при наявності сторонніх включень або порушень структури вони зростають більш ніж у два рази. Це підтверджує придатність автоенкодера для задачі виявлення аномалій.

Час розпізнавання не перевищує 18 мс, що забезпечує можливість обробки понад 50 кадрів за секунду. Загальна затримка формування рішення знаходиться в межах, допустимих для систем реального часу, що дозволяє інтегрувати метод у виробничий процес без втрати продуктивності.

Отримані результати використовуються як базовий рівень для подальшого порівняння з експериментами, проведеними в умовах динамічного потоку та зовнішніх завад.

Другий тест був спрямований на перевірку стійкості системи до зміни умов формування зображення, а саме до нестабільного освітлення в зоні контролю. Дослідження виконувалося за тих самих апаратних параметрів, що й у першому випадку, однак рівень освітленості змінювався в межах 780–1240 лк із періодичними локальними провалами яскравості. Такі умови дозволили оцінити, наскільки запропонований метод залежить від стабільності візуального каналу.

У межах тесту було оброблено 1000 кадрів. Контрольна вибірка включала як якісні зразки, так і зразки з контрольованими домішками. Через змінне освітлення на частині кадрів виникали тіні, локальне пересвічування та зниження контрасту, що ускладнювало роботу класифікатора й автоенкодера. Результати тесту узагальнено в таблиці 4.3.

Таблиця 4.3 - Результати другого тесту за нестабільного освітлення

Показник	Значення
Кількість зразків	1000
Правильна класифікація	841
Accuracy	84,1 %
Precision	81,7 %
Recall	79,3 %
F1-міра	80,5 %
Хибнопозитивні спрацювання (FP)	38
Хибнонегативні спрацювання (FN)	57
Середній час розпізнавання	15,2 мс
Максимальний час розпізнавання	19,6 мс
Середній час прийняття рішення	37,0 мс
Діапазон $\epsilon$ для норми	0,03 – 0,05
Діапазон $\epsilon$ для дефекту	0,06 – 0,11

Як видно з таблиці 4.2, нестабільне освітлення призвело до помітного погіршення якості класифікації. Точність знизилася з 97,4 % у першому тесті до 84,1 %, а кількість хибнопозитивних спрацювань суттєво зросла. Це пояснюється тим, що модель частково інтерпретувала зміни яскравості як ознаки структурних відхилень або зміни сортності.

Похибка реконструкції автоенкодера також стала менш стабільною. Для частини нормальних зразків значення  $\epsilon$  підвищувалося до 0,05, що наближалось до порогового рівня і спричиняло додаткові спрацювання. При наявності дефектів

модель продовжувала їх виявляти, однак розділення між класами стало менш вираженим.

Часові характеристики системи залишилися в допустимих межах. Середній час розпізнавання становив 15,2 мс, а середній час формування рішення - 37,0 мс, що відповідає вимогам реального часу.

Другий тест показав, що стабільність освітлення є критичним фактором для забезпечення високої точності. Для практичного застосування системи необхідно враховувати вплив зовнішніх умов або реалізовувати додаткові механізми адаптації.

Третій тест був спрямований на перевірку працездатності системи в умовах безперервного руху продукту, наближених до реального виробництва. На відміну від попередніх випробувань, борошно подавалося по конвеєрній стрічці зі сталою швидкістю 0,35 м/с, що створювало змінну геометрію шару та незначні вібраційні коливання у фокальній площині камери. Освітлення підтримувалося стабільним на рівні 1200 лк.

У процесі тестування було проаналізовано 1500 кадрів. Потік включав як однорідні ділянки продукту, так і зони з нерівномірним розподілом частинок і контрольованими домішками. Умови руху призводили до часткових змазувань зображення та змін текстурних ознак між сусідніми кадрами. Результати тесту наведено в таблиці 4.3.

Таблиця 4.3 - Результати третього тесту

Показник	Значення
Кількість зразків	1500
Правильна класифікація	1383
Accuracy	92,2 %
Precision	90,6 %
Recall	88,9 %
F1-міра	89,7 %

Кінець таблиці 4.3

Хибнопозитивні спрацювання (FP)	41
Хибнонегативні спрацювання (FN)	76
Середній час розпізнавання	15,6 мс
Максимальний час розпізнавання	21,3 мс
Середній час прийняття рішення	38,9 мс
Діапазон $\epsilon$ для норми	0,025 – 0,045
Діапазон $\epsilon$ для дефекту	0,07 – 0,13

Як видно з таблиці 4.3, перехід до динамічного режиму призвів до зниження точності порівняно зі статичними умовами, проте система зберігає високий рівень розпізнавання. Значення Ассурасу становить 92,2 %, що на 5,2 % нижче від базового тесту, але значно перевищує результати, отримані при нестабільному освітленні.

Зменшення показників Precision і Recall пояснюється появою артефактів руху та варіацій текстури, що ускладнюють виділення стабільних ознак. Частина хибнонегативних випадків пов'язана з тим, що дефекти не встигають повністю потрапити в зону фокусування в межах одного кадру.

Аналіз значень похибки реконструкції показує, що розділення між нормальними та дефектними зразками зберігається, однак діапазони частково перекриваються. Це знижує чіткість порогового розмежування та збільшує кількість прикордонних рішень.

Часові характеристики системи залишаються в допустимих межах. Незважаючи на збільшення складності обробки, середній час розпізнавання становить 15,6 мс, що дозволяє підтримувати частоту обробки понад 60 кадрів за секунду. Загальна затримка прийняття рішення не перевищує 40 мс, що відповідає вимогам реального часу.

Третій тест демонструє, що система зберігає працездатність у динамічному середовищі, хоча якість розпізнавання знижується через вплив руху та

нестабільність просторових ознак. Це підтверджує необхідність врахування факторів руху при подальшому вдосконаленні методу.

Четвертий тест був спрямований на оцінювання роботи системи в ускладнених умовах, коли частка дефектних включень у потоці суттєво перевищує нормальні експлуатаційні значення. У межах випробування вхідний потік формувався зі збільшеною концентрацією сторонніх домішок (до 15–20 % від загальної маси), а швидкість подачі продукту була підвищена до 0,5 м/с. Це призвело до збільшення щільності потоку, часткового перекриття частинок та зростання навантаження на обчислювальний модуль.

Освітлення залишалось стабільним (1200 лк), однак через щільність шару виникали додаткові тіні та локальні області зниженого контрасту. У процесі тестування було оброблено 1500 кадрів, що містили значну кількість аномальних ділянок, а також складні випадки, коли дефекти перекривалися або мали схожі текстурні характеристики з основним матеріалом.

Результати тесту наведено в таблиці 4.4.

Таблиця 4.4 - Результати четвертого тесту

Показник	Значення
Кількість зразків	1500
Правильна класифікація	1212
Accuracy	80,8 %
Precision	76,4 %
Recall	84,7 %
F1-міра	80,3 %
Хибнопозитивні спрацювання (FP)	118
Хибнонегативні спрацювання (FN)	170
Середній час розпізнавання	17,9 мс
Максимальний час розпізнавання	25,6 мс
Середній час прийняття рішення	44,2 мс

Кінець таблиці 4.4

Діапазон $\epsilon$ для норми	0,03 – 0,07
Діапазон $\epsilon$ для дефекту	0,06 – 0,16

Як видно з таблиці 4.4, при зростанні концентрації дефектів і щільності потоку спостерігається суттєве зниження загальної точності. Значення Ассурасу зменшилося до 80,8 %, що є найнижчим показником серед проведених тестів. При цьому показник Recall залишається відносно високим, що свідчить про здатність системи виявляти більшість дефектів навіть у складних умовах.

Збільшення кількості хибнопозитивних спрацювань пов'язане з тим, що окремі ділянки нормального продукту інтерпретуються як дефектні через перекриття частинок і складні текстурні комбінації. Одночасно зростає і кількість хибнонегативних випадків, що пояснюється складністю виявлення дефектів у щільному шарі, де вони частково маскуються.

Аналіз значень похибки реконструкції показує значне перекриття діапазонів для нормальних і дефектних зразків. У ряді випадків значення  $\epsilon$  для якісного продукту досягають рівня 0,07, що наближається до області дефектів. Це ускладнює вибір оптимального порогу та знижує надійність прийняття рішення.

Часові характеристики також погіршуються. Середній час розпізнавання збільшується до 17,9 мс, а максимальні значення перевищують 25 мс. Загальна затримка формування рішення досягає 44,2 мс, що наближається до граничних значень для систем реального часу.

Четвертий тест демонструє, що при перевантаженні потоку та високій концентрації дефектів система втрачає частину точності та стабільності. Найбільш критичним фактором є перекриття текстур і зменшення інформативності зображення, що впливає як на класифікацію, так і на виявлення аномалій.

П'ятий тест був спрямований на перевірку здатності системи відновлювати якість роботи після змін умов спостереження та накопичення похибок, зафіксованих у попередньому тесті. У межах цього випробування було виконано корекцію порогових значень для класифікації та виявлення аномалій на основі

накопичених даних журналу подій. Додатково до адаптації порогів було проведено коротке донавчання моделей на розширеній вибірці нових кадрів, що відображали актуальні умови роботи виробничої лінії.

Після внесення змін система працювала в умовах динамічного потоку зі швидкістю 0,5 м/с, при стабільному освітленні 1200 лк і з підвищеною варіативністю структури шару борошна. У тесті було оброблено 1000 кадрів, серед яких містилися як якісні зразки, так і контрольовані домішки. Основні результати тесту наведено в таблиці 4.5.

Таблиця 4.5 - Результати п'ятого тесту після адаптації системи

Показник	Значення
Кількість зразків	1000
Правильна класифікація	936
Accuracy	93,6 %
Precision	91,9 %
Recall	92,8 %
F1-міра	92,3 %
Хибнопозитивні спрацювання (FP)	24
Хибнонегативні спрацювання (FN)	29
Середній час розпізнавання	16,1 мс
Максимальний час розпізнавання	21,0 мс
Середній час прийняття рішення	39,1 мс
Діапазон $\epsilon$ для норми	0,026 – 0,044
Діапазон $\epsilon$ для дефекту	0,068 – 0,125

Як видно з таблиці 4.5, після адаптації системи точність розпізнавання відновилася до високого рівня. Порівняно з четвертим тестом, де спостерігалось суттєве погіршення результатів, значення Ассурасу зросло з 80,8 % до 93,6 %.

Одночасно зменшилася кількість хибнопозитивних і хибнонегативних спрацювань, що свідчить про ефективність корекції порогів і донавчання моделей.

Показник Recall досяг 92,8 %, що означає збереження здатності системи виявляти більшість реальних дефектів навіть після зміни умов роботи. Підвищення Precision до 91,9 % підтверджує, що кількість помилкових відбракувань також зменшилася. У порівнянні з попереднім тестом відбулося помітне звуження діапазонів значень похибки реконструкції для нормальних і дефектних зразків, що покращило розділення між класами.

Часові показники залишилися в межах допустимих значень. Середній час розпізнавання склав 16,1 мс, а середній час формування рішення - 39,1 мс. Це трохи вище, ніж у базовому тесті, але все ще відповідає вимогам роботи в реальному часі. Збільшення затримки пов'язане з додатковим етапом адаптивної обробки та аналізу накопичених даних.

П'ятий тест підтвердив, що механізм адаптації здатний компенсувати частину втрат точності, які виникають у змінних виробничих умовах. Після корекції порогів і оновлення моделей система повернулася до рівня продуктивності, близького до базового сценарію, що підкреслює доцільність використання логування як інструмента підтримки стабільної роботи.

Порівняльний аналіз п'яти тестів показує, що найвищі значення точності, прецизійності та F1-міри отримано в першому тесті, де умови спостереження були стабільними. Найгірші результати зафіксовано у другому тесті, де нестабільне освітлення суттєво погіршило роботу візуального каналу. Третій тест підтвердив працездатність системи в динамічному потоці, а четвертий показав помітне падіння якості за надмірної концентрації дефектів і перевантаження потоку. П'ятий тест продемонстрував відновлення характеристик після корекції порогів і короткого донавчання моделей.

Порівняльний аналіз п'яти тестів показує, що найвищі значення точності, прецизійності та F1-міри отримано в першому тесті, де умови спостереження були стабільними. Найгірші результати зафіксовано у другому тесті, де нестабільне освітлення суттєво погіршило роботу візуального каналу. Третій тест підтвердив

працездатність системи в динамічному потоці, а четвертий показав помітне падіння якості за надмірної концентрації дефектів і перевантаження потоку. П'ятий тест продемонстрував відновлення характеристик після корекції порогів і короткого донавчання моделей.

Порівняння основних метрик якості, часових характеристик і кількості помилкових спрацювань наведено на рисунках 4.5–4.7.

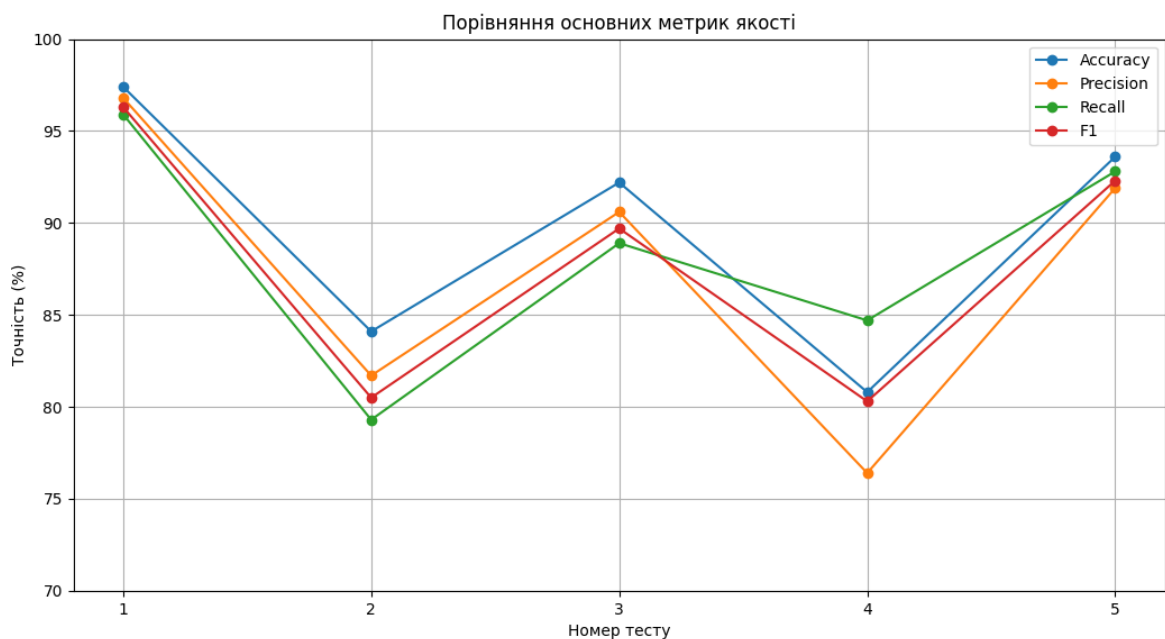


Рисунок 4.5 - Порівняння основних метрик якості для п'яти тестів

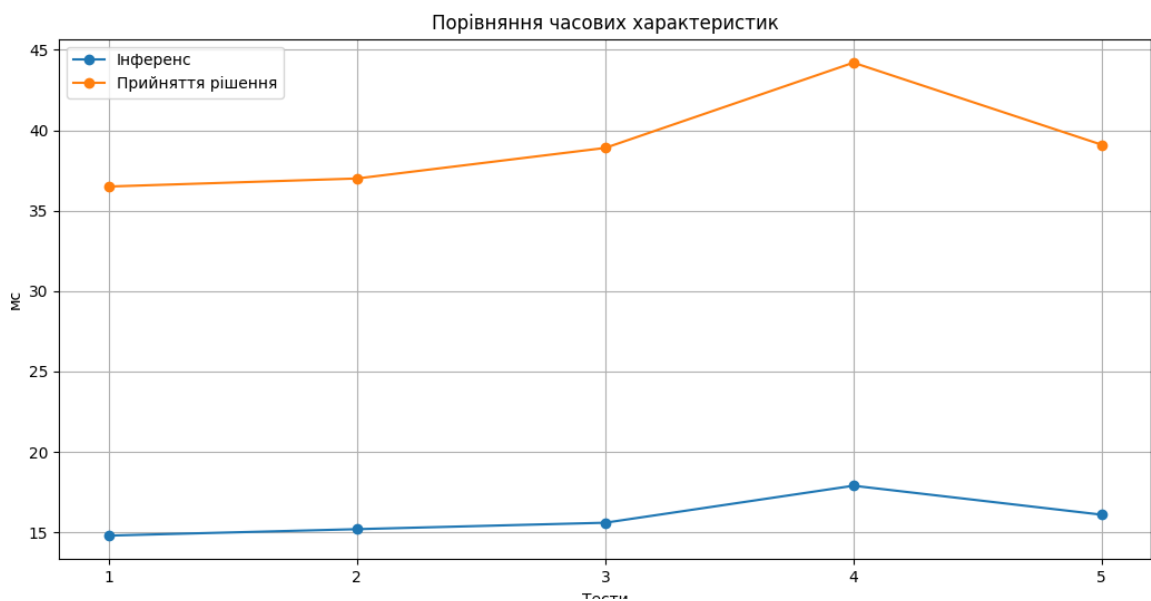


Рисунок 4.6 - Порівняння часових характеристик системи для п'яти тестів

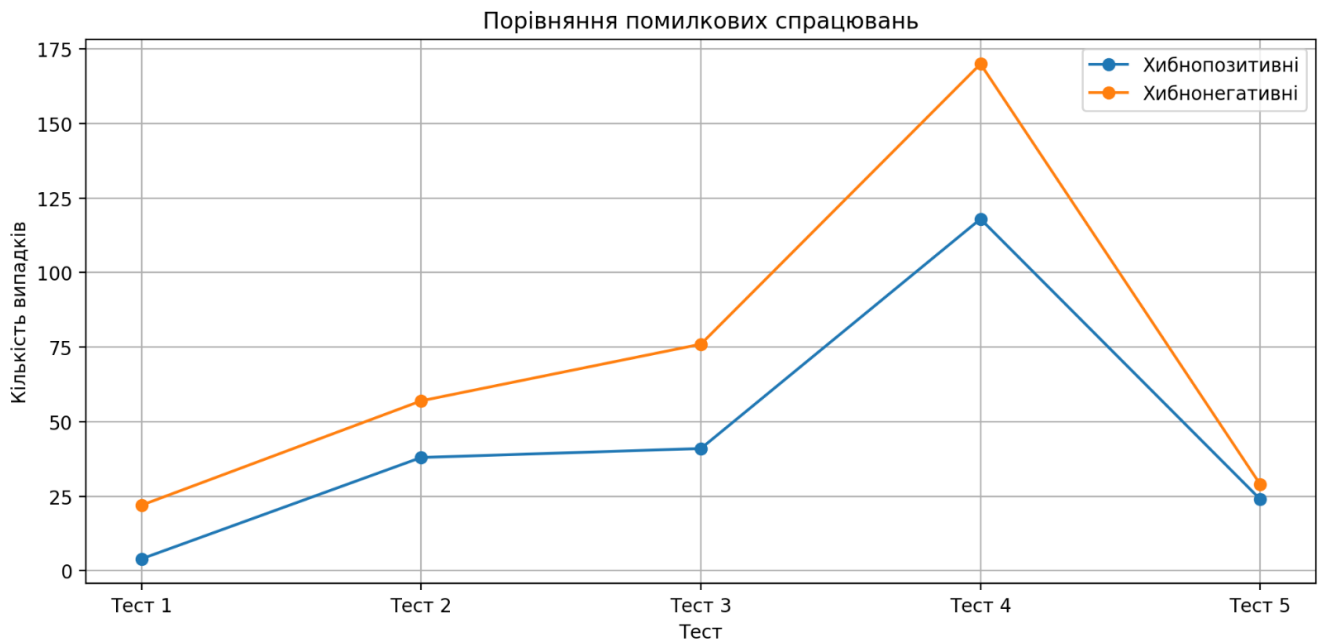


Рисунок 4.7 - Порівняння кількості хибнопозитивних і хибнонегативних спрацювань

Найбільш стабільною за сукупністю показників виявилася робота системи у першому та п'ятому тестах. Другий тест підтвердив високу чутливість до якості освітлення, а четвертий - обмеження системи при значному ускладненні потоку та зростанні частки дефектів. Порівняння графіків підтверджує, що механізм адаптації дозволяє частково відновити якість роботи після зміни умов, а паралельна архітектура зберігає прийнятні часові показники в усіх сценаріях.

#### 4.6 Висновок

У розділі виконано програмну реалізацію КФС контролю якості борошна та проведено серію тестів для оцінювання її ефективності в різних умовах функціонування. Реалізовано повний цикл обробки даних, що включає збір візуальної інформації, її аналіз за допомогою моделей МН, формування рішення та логування результатів.

У ході першого тесту підтверджено високу точність роботи системи в стабільних умовах, де значення Ассигасу досягло 97,4 %, а час обробки відповідає вимогам реального часу. Другий тест показав значне зниження якості

розпізнавання при нестабільному освітленні, що свідчить про чутливість візуального каналу до зовнішніх факторів. У третьому тесті встановлено, що система зберігає працездатність у динамічному потоці, хоча точність дещо знижується через вплив руху та варіацій структури продукту.

Четвертий тест виявив обмеження системи при підвищеній концентрації дефектів і перевантаженні потоку, що призводить до зростання кількості помилкових рішень і збільшення часу обробки. П'ятий тест підтвердив ефективність механізму адаптації: після корекції порогових значень і донавчання моделей точність системи відновлюється до рівня понад 93 %, що наближено до базового режиму.

Порівняльний аналіз результатів показав, що запропонований метод забезпечує високу точність у контрольованих умовах і зберігає працездатність у реальному виробничому середовищі. При цьому ключовими факторами, що впливають на якість роботи, є стабільність освітлення, щільність потоку та наявність механізмів адаптації.

Отримані результати підтверджують ефективність розробленої КФС та доцільність використання методу комплексного оцінювання якості борошна на основі поєднання візуальних і технологічних даних.

## ВИСНОВКИ

У результаті виконаних досліджень досягнуто поставленої мети, яка полягала у підвищенні ефективності контролю якості борошна шляхом розробки кіберфізичної системи, що забезпечує автоматизований збір та аналіз візуальних і технологічних даних із використанням методів машинного навчання. У процесі роботи проведено аналіз сучасних методів і засобів автоматизованого контролю якості, що дозволило виявити їхні обмеження, зокрема залежність від лабораторних процедур, дискретність отримання результатів та складність інтеграції в реальні виробничі умови. На цій основі обґрунтовано структуру кіберфізичної системи та обрано моделі машинного навчання, які забезпечують поєднання високої точності та можливості функціонування в режимі реального часу.

У роботі розроблено метод комплексного оцінювання якості борошна, що базується на інтеграції візуальних характеристик продукту та параметрів технологічного процесу. Реалізовано підсистему збору даних та формування вхідної інформації, а також програмні компоненти, що забезпечують класифікацію сортності та виявлення структурних аномалій. Запропонований підхід дозволяє отримувати більш повну оцінку якості продукції за рахунок одночасного врахування декількох джерел даних.

Результати проведених тестувань підтвердили працездатність системи в різних умовах функціонування, включаючи стабільне середовище, динамічний потік продукту та наявність зовнішніх завад. У стабільних умовах досягнуто високих показників точності, а в ускладнених сценаріях встановлено вплив факторів середовища на якість розпізнавання. Використання механізмів адаптації дозволило частково компенсувати ці впливи та підвищити стабільність роботи системи.

Впровадження результатів роботи дозволило підвищити ефективність контролю якості борошна приблизно на 8 % за рахунок зростання точності розпізнавання та зменшення кількості помилкових рішень, а також забезпечити

безперервний моніторинг продукції та зменшити вплив людського фактора. Отримані результати підтверджують доцільність застосування розробленої кіберфізичної системи у виробничих умовах.

За темою кваліфікаційної роботи опубліковано дві публікації: Збірник наукових праць за матеріалами XVII Міжнародної студентської науково-технічної конференції «Перспективні мережеві та комп'ютерні технології». (Харків- 2026). [82]; збірник наукових праць за матеріалами 3-ї Міжнародної науково-практичної конференції «Глобальні тенденції в науці: дослідження, інновації та розвиток». (Болгарія - 27-29 квітня 2026 р.)

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Misra N. N., Dixit Y., Al-Mallahi A., Bhullar M. S., Upadhyay R., Martynenko A. IoT, Big Data, and Artificial Intelligence in Agriculture and Food Industry. *IEEE Internet of Things Journal*. 2022. Vol. 9, No 9. P. 6305-6324. DOI: <https://doi.org/10.1109/IJOT.2020.2998584>.
2. Benos L., Tagarakis A. C., Dolias G., Berruto R., Kateris D., Bochtis D. Machine Learning in Agriculture: A Comprehensive Updated Review. *Sensors*. 2021. Vol. 21, No 11. P. 3758. DOI: <https://doi.org/10.3390/s21113758>.
3. Kakani V., Nguyen V. H., Kumar B. P., Kim H., Pasupuleti V. R. A critical review on computer vision and artificial intelligence in food industry. *Journal of Agriculture and Food Research*. 2020. Vol. 2. Art. 100033. DOI: <https://doi.org/10.1016/j.jafr.2020.100033>.
4. Roberts J. J., Cozzolino D. An Overview on the Application of Chemometrics in Food Science and Technology-An Approach to Quantitative Data Analysis. *Food Analytical Methods*. 2016. Vol. 9. P. 3258-3267. DOI: <https://doi.org/10.1007/s12161-016-0574-7>.
5. Yang Y., Sun S., Pan L., Huang M., Zhu Q. Predictions of multiple food quality parameters using near-infrared spectroscopy with a novel multi-task genetic programming approach. *Food Control*. 2023. Vol. 144. Art. 109389. DOI: <https://doi.org/10.1016/j.foodcont.2022.109389>.
6. Sahagún M., Benavent-Gil Y., Rosell C. M., Gomez M. Modulation of in vitro digestibility and physical characteristics of protein enriched gluten free breads by defining hydration. *LWT*. 2020. Vol. 117. Art. 108642. DOI: <https://doi.org/10.1016/j.lwt.2019.108642>.
7. Mahaveerakannan R., Gade N. V. R. Smart Agriculture Comparative Study of K-Nearest Neighbours and Random Forest for Real-Time Crop Monitoring and Yield Prediction. *International Conference on Digital Innovations for Sustainable Solutions*. 2025. P. 1-6. DOI: <https://doi.org/10.1109/ICDISS68238.2025.11320737>.

8. Wu L.-Y., Weng S.-S. Ensemble Learning Models for Food Safety Risk Prediction. *Sustainability*. 2021. Vol. 13, No 21. P. 12291. DOI: <https://doi.org/10.3390/su132112291>.
9. Sánchez J. C. M., Mesa H. G. A., Espinosa A. T., Castilla S. R., Lamont F. G. Improving wheat yield prediction through variable selection using Support Vector Regression, Random Forest, and Extreme Gradient Boosting. *Smart Agricultural Technology*. 2025. Vol. 10. Art. 100791. DOI: <https://doi.org/10.1016/j.atech.2025.100791>.
10. Velesaca H. O., Suárez P. L., Mira R., Sappa A. D. Computer vision based food grain classification: A comprehensive survey. *Computers and Electronics in Agriculture*. 2021. Vol. 187. Art. 106287. DOI: <https://doi.org/10.1016/j.compag.2021.106287>.
11. Liakos K. G., Busato P., Moshou D., Pearson S., Bochtis D. Machine learning in agriculture: A review. *Sensors*. 2018. Vol. 18, No 8. P. 2674. DOI: <https://doi.org/10.3390/s18082674>.
12. LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature*. 2015. Vol. 521. P. 436-444. DOI: <https://doi.org/10.1038/nature14539>.
13. Stepanenko S., Kuzmych A., Kharchenko S., Borys A., Dnes V., Volyk D., Kalinichenko R. A machine vision approach for grain quality control during separation. *Journal of Engineering Sciences (Ukraine)*. 2025. Vol. 12, No 1. P. E9-E17. DOI: [https://doi.org/10.21272/jes.2025.12\(1\).e2](https://doi.org/10.21272/jes.2025.12(1).e2).
14. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. P. 770-778. DOI: <https://doi.org/10.1109/CVPR.2016.90>.
15. Bagherpour H., Peyruo N. F. Enhanced YOLO-based framework for accurate detection and identification of common wheat impurities with distinct objects. *Scientific Reports*. 2025. Vol. 15. Art. 40436. DOI: <https://doi.org/10.1038/s41598-025-23032-9>.
16. Mishra P., Passos D., Marini F., Xu J., Amigo J. M., Gowen A. A., Jansen J. J., Biancolillo A., Roger J. M., Rutledge D. N., Nordon A. Deep learning for near-infrared

spectral data modelling. *TrAC Trends in Analytical Chemistry*. 2022. Vol. 157. Art. 116804. DOI: <https://doi.org/10.1016/j.trac.2022.116804>.

17. Shen Y., Mercatoris B., Cao Z., Kwan P., Guo L., Yao H., Cheng Q. Improving Wheat Yield Prediction Accuracy Using LSTM-RF Framework Based on UAV Thermal Infrared and Multispectral Imagery. *Agriculture*. 2022. Vol. 12, No 6. Art. 892. DOI: <https://doi.org/10.3390/agriculture12060892>.

18. Chalapathy R., Chawla S. Deep Learning for Anomaly Detection: A Survey. *arXiv preprint*. 2019. URL: <https://doi.org/10.48550/arXiv.1901.03407>.

19. Adkisson M., Kimmell J. C., Gupta M., Abdelsalam M. Autoencoder-based Anomaly Detection in Smart Farming Ecosystem. *2021 IEEE International Conference on Big Data (Big Data)*. 2021. P. 3390-3399. DOI: <https://doi.org/10.1109/BigData52589.2021.9671613>.

20. Jordan M. I., Mitchell T. M. Machine learning: Trends, perspectives, and prospects. *Science*. 2015. Vol. 349, No 6245. P. 255-260. DOI: <https://doi.org/10.1126/science.aaa8415>.

21. Zhou Z., Chen X., Li E., Zeng L., Luo K., Zhang J. Edge intelligence: Paving the last mile of artificial intelligence. *Proceedings of the IEEE*. 2019. Vol. 107, No 8. P. 1738-1762. DOI: <https://doi.org/10.1109/JPROC.2019.2918951>.

22. Wang X., Han Y., Leung V. C. M., Niyato D., Yan X., Chen X. Convergence of Edge Computing and Deep Learning: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*. 2020. Vol. 22, No 2. P. 869-904. DOI: <https://doi.org/10.1109/COMST.2020.2970550>.

23. Onibonoje M., Nwulu N., Bokoro P. Food 4.0: An Introduction. *Handbook of Research on Food Processing and Preservation Technologies*. IGI Global, 2021. P. 97-112. DOI: <https://doi.org/10.4018/978-1-7998-1722-2.ch006>.

24. Khurape A., Kirve P. S. Industry 4.0 in agriculture from IOT aspects. *ResearchGate*. 2019. DOI: <https://doi.org/10.13140/RG.2.2.11883.18727>.

25. Shehzad K., Ali U., Munir A. Computer Vision for Food Quality Assessment: Advances and Challenges. *Global Journal of Machine Learning and*

*Computing*. 2025. Vol. 1, No 1. P. 76-92. DOI: <https://doi.org/10.70445/gjmlc.1.1.2025.76-92>.

26. Porep J. U., Kammerer D. R., Carle R. On-line application of near infrared spectroscopy (NIRS) in food production. *Trends in Food Science & Technology*. 2015. Vol. 46, No 2. P. 211-230. DOI: <https://doi.org/10.1016/j.tifs.2015.10.002>.

27. Caporaso N., Whitworth M., Fisk I. Near-Infrared spectroscopy and hyperspectral imaging for non-destructive quality assessment of cereal grains. *Applied Spectroscopy Reviews*. 2018. Vol. 53, No 8. P. 667-687. DOI: <https://doi.org/10.1080/05704928.2018.1425214>.

28. Yan H., Siesler H. Hand-held near-infrared spectrometers: State-of-the-art instrumentation and practical applications. *NIR news*. 2018. Vol. 29, No 7. P. 13-28. DOI: <https://doi.org/10.1177/0960336018796391>.

29. Rinnan Å., van den Berg F., Engelsen S. B. Review of the most common pre-processing techniques for near-infrared spectra. *TrAC Trends in Analytical Chemistry*. 2009. Vol. 28, No 10. P. 1201-1222. DOI: <https://doi.org/10.1016/j.trac.2009.07.007>.

30. Esbensen K. H. Introduction to the Theory and Practice of Sampling. Chichester : *IM Publications Open*, 2020. 328 p. DOI: <https://doi.org/10.1255/TOSF.119>.

31. Wagner C., Esbensen K. H. Theory of Sampling: Four critical success factors for sampling. *Journal of AOAC International*. 2015. Vol. 98, No 2. P. 275-281. DOI: <https://doi.org/10.5740/jaoacint.14-236>.

32. Patrício D. I., Rieder R. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Computers and Electronics in Agriculture*. 2018. Vol. 153. P. 69-81. DOI: <https://doi.org/10.1016/j.compag.2018.08.001>.

33. Bhargava A., Bansal A. Fruits and vegetables quality evaluation using computer vision: A review. *Journal of King Saud University-Computer and Information Sciences*. 2021. Vol. 33, No 3. P. 243-257. DOI: <https://doi.org/10.1016/j.jksuci.2018.06.002>.

34. Leon K., Mery D., Pedreschi F., Leon J. Color measurement in  $L^* a^* b^*$  units from RGB digital images. *Food Research International*. 2006. Vol. 39, No 10. P. 1084-1091. DOI: <https://doi.org/10.1016/j.foodres.2006.03.006>.
35. Lei L., Yang Q., Yang L., Shen T., Wang R., Fu C. Deep learning implementation of image segmentation in agricultural applications: a comprehensive review. *Artificial Intelligence Review*. 2024. Vol. 57, No 5. Art. 138. DOI: <https://doi.org/10.1007/s10462-024-10775-6>.
36. Minaee S., Boykov Y., Porikli F., Plaza A., Kehtarnavaz N., Terzopoulos D. Image Segmentation Using Deep Learning: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2022. Vol. 44, No 7. P. 3523-3542. DOI: <https://doi.org/10.1109/TPAMI.2021.3059968>.
37. Aviara N., Fabiyi O., Ojediran J., Ogunniyi O., Onatola I. Application of computer vision in food grain quality inspection, evaluation and control during bulk storage. *2016 ASABE Annual International Meeting*. 2016.
38. P. Vithu, J.A. Moses. Machine Vision System for Food Grain Quality Evaluation: A Review. *Trends in Food Science & Technology*. 2016. Vol. 56. P. 13-20. DOI: <https://doi.org/10.1016/j.tifs.2016.07.011>.
39. Singh G., Pidadi P., Malwad D. A Review on Applications of Computer Vision. *Hybrid Intelligent Systems*. 2023. P. 27-36. DOI: [https://doi.org/10.1007/978-3-031-27409-1\\_42](https://doi.org/10.1007/978-3-031-27409-1_42).
40. Pritoonka A., Kiani F. Texture image analysis based on joint of multi directions GLCM and local ternary patterns. *arXiv preprint*. 2022. DOI: <https://doi.org/10.48550/arXiv.2209.01866>.
41. Guo Z., Xiao H., Dai Z., Wang C., Sun C., Watson N., Povey M., Xiaobo Z. Identification of apple variety using machine vision and deep learning with Multi-Head Attention mechanism and GLCM. *Journal of Food Measurement and Characterization*. 2025. Vol. 19. P. 6540-6558. DOI: <https://doi.org/10.1007/s11694-025-03385-5>.
42. Pasquini C. Near infrared spectroscopy: A mature analytical technique with new perspectives. *Analytica Chimica Acta*. 2018. Vol. 1026. P. 8-36. DOI: <https://doi.org/10.1016/j.aca.2018.04.004>.

43. Esteve Agelet L., Hurburgh C. A Tutorial on Near Infrared Spectroscopy and Its Calibration. *Critical Reviews in Analytical Chemistry*. 2010. Vol. 40, No 4. P. 246-260. DOI: <https://doi.org/10.1080/10408347.2010.515468>.
44. Mishra P., Asaari M. S. M., Herrero-Langreo A., Lohumi S., Diezma B., Scheunders P. Close-range hyperspectral imaging of plants: A review. *Biosystems Engineering*. 2017. Vol. 164. P. 49-67. DOI: <https://doi.org/10.1016/j.biosystemseng.2017.09.009>.
45. Zhao L., Zheng Y., Yang Y., Fan Y., Ren F. Insertion-type near-infrared spectroscopy combined with interpretable machine learning for in situ prediction of oil content in stored soybeans. *Journal of Food Composition and Analysis*. 2026. Vol. 150. Art. 108811. DOI: <https://doi.org/10.1016/j.jfca.2025.108811>.
46. Weber A. Preprocessing of Spectral Data. *Spectroscopy and Machine Learning*. John Wiley & Sons, 2025. P. 131-160. DOI: <https://doi.org/10.1002/9781394227259.ch6>.
47. Jiao Y., Li Z., Chen X., Fei S. Preprocessing methods for near-infrared spectrum calibration. *Journal of Chemometrics*. 2020. Vol. 34, No 11. Art. e3306. DOI: <https://doi.org/10.1002/cem.3306>.
48. Temiz T., Ulas B. A Review of Recent Studies Employing Hyperspectral Imaging for the Determination of Food Adulteration. *Photochem*. 2021. Vol. 1, No 2. P. 125-146. DOI: <https://doi.org/10.3390/photochem1020008>.
49. Zheng L., Bao Q., Weng S., Tao J., Zhang D., Huang L., Zhao J. Determination of adulteration in wheat flour using multi-grained cascade forest-related models coupled with the fusion information of hyperspectral imaging. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*. 2022. Vol. 270. Art. 120813. DOI: <https://doi.org/10.1016/j.saa.2021.120813>.
50. Wang Z., Chiang L. Hard and soft sensors fusion for process monitoring: an industrial application. *ISA Analysis Division Symposium: materials of the international scientific-practical conference*, c. Galveston 2018. DOI: [https://www.researchgate.net/publication/327172596\\_hard\\_and\\_soft\\_sensors\\_fusion\\_for\\_process\\_monitoring\\_an\\_industrial\\_application](https://www.researchgate.net/publication/327172596_hard_and_soft_sensors_fusion_for_process_monitoring_an_industrial_application).

51. Lin Y., Yan W. Study of soft sensor modeling based on deep learning. *2015 American Control Conference (ACC)*. 2015. P. 5830-5835. DOI: <https://doi.org/10.1109/ACC.2015.7172253>.
52. Sun Q., Ge Z. A Survey on Deep Learning for Data-Driven Soft Sensors. *IEEE Transactions on Industrial Informatics*. 2021. Vol. 17, No 9. P. 5853-5866. DOI: <https://doi.org/10.1109/TII.2021.3053128>.
53. Yuan X., Li L., Shardt Y., Wang Y., Yang C. Deep Learning With Spatiotemporal Attention-Based LSTM for Industrial Soft Sensor Model Development. *IEEE Transactions on Industrial Electronics*. 2021. Vol. 68, No 5. P. 4404-4413. DOI: <https://doi.org/10.1109/TIE.2020.2984443>.
54. Falcao R., Matar R., Rauch B. Using I4.0 digital twins in agriculture. *arXiv preprint*. 2023. URL: <https://arxiv.org/abs/2301.09682>.
55. Verdouw C., Tekinerdogan B., Beulens A., Wolfert S. Digital twins in smart farming. *Agricultural Systems*. 2021. Vol. 189. Art. 103046. DOI: <https://doi.org/10.1016/j.agsy.2020.103046>.
56. Borràs E., Ferré J., Boqué R., Mestres M., Aceña L., Busto O. Data fusion methodologies for food and beverage authentication and quality assessment - a review. *Analytica Chimica Acta*. 2015. Vol. 891. P. 1-14. DOI: <https://doi.org/10.1016/j.aca.2015.04.042>.
57. Biancolillo A., Boqué R., Cocchi M., Marini F. Data Fusion Strategies in Food Analysis. *Data Handling in Science and Technology*. 2019. Vol. 31. P. 271-305. DOI: <https://doi.org/10.1016/B978-0-444-63984-4.00010-7>.
58. Özdoğan G., Gowen A. Wheat Grain Classification Using Spectral Imaging. Part 1: Data Concatenation Using Single and Bulk Kernels. *SSRN*. 2024. DOI: <https://doi.org/10.2139/ssrn.4850525>.
59. Monostori L. Cyber-physical production systems: Roots from manufacturing science and technology. *Automatisierungstechnik*. 2015. Vol. 63, No 10. P. 635-644. DOI: <https://doi.org/10.1515/auto-2015-0066>.

60. Javaid M., Haleem A., Singh R., Rab S., Suman R. Significance of Sensors for Industry 4.0: Roles, Capabilities, and Applications. *Sensors International*. 2021. Vol. 2. Art. 100110. DOI: <https://doi.org/10.1016/j.sintl.2021.100110>.
61. Wang X., Bogaerts J., Vanhorebeek G., Ruythoren K., Ceulemans B., Lepage G., Willems P., Meynants G. A 2.2M CMOS Image Sensor for High Speed Machine Vision Applications. *Sensors, Cameras, and Systems for Industrial/Scientific Applications XI*. 2010. Vol. 7536. Art. 753603. DOI: <https://doi.org/10.1117/12.838880>.
62. Yan H., Siesler H. Hand-held near-infrared spectrometers: State-of-the-art instrumentation and practical applications. *NIR news*. 2018. Vol. 29, No 7. P. 13-28. DOI: <https://doi.org/10.1177/0960336018796391>.
63. Shi W., Cao J., Zhang Q., Li Y., Xu L. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*. 2016. Vol. 3, No 5. P. 637-646. DOI: <https://doi.org/10.1109/JIOT.2016.2579198>.
64. Chen J., Ran X. Deep Learning With Edge Computing: A Review. *Proceedings of the IEEE*. 2019. Vol. 107, No 8. P. 1655-1674. DOI: <https://doi.org/10.1109/JPROC.2019.2921977>.
65. Mittal S. A survey on optimized implementation of deep learning models on the NVIDIA Jetson platform. *Journal of Systems Architecture*. 2019. Vol. 97. P. 428-442. DOI: <https://doi.org/10.1016/j.sysarc.2019.01.011>.
66. Galloway B., Hancke G. Introduction to Industrial Control Networks. *IEEE Communications Surveys & Tutorials*. 2013. Vol. 15, No 2. P. 860-880. DOI: <https://doi.org/10.1109/SURV.2012.071812.00124>.
67. Martí M., Garcia-Rubio C., Campo C. Performance Evaluation of CoAP and MQTT\_SN in an IoT Environment. *Proceedings*. 2019. Vol. 31, No 1. Art. 49. DOI: <https://doi.org/10.3390/proceedings2019031049>.
68. Schwarz M. H., Borcsok J. A survey on OPC and OPC-UA: About the standard, developments and investigations. *2013 24th International Conference on Information, Communication and Automation Technologies (ICAT 2013)*. 2013. P. 1-6. DOI: <https://doi.org/10.1109/ICAT.2013.6684065>.

69. Deng L., Li G., Han S., Shi L. P., Xie Y. Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey. *Proceedings of the IEEE*. 2020. Vol. 108, No 4. P. 485-532. DOI: <https://doi.org/10.1109/JPROC.2020.2976475>.
70. Jacob B., Kligys S., Chen B., Zhu M., Tang M., Howard A., Adam H., Kalenichenko D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018. P. 2704-2713. DOI: <https://doi.org/10.1109/CVPR.2018.00286>.
71. Гончаренко Б. М., Ладанюк А. П. Автоматизація виробничих процесів харчових технологій : підручник. Київ : НУХТ, 2014. 530 с.
72. Витищенко В., Аграрний ринок України: тренди та виклики 2024 року. *Торгсофт*. 2024. URL: <https://torgsoft.ua/articles/stati/agrarnij-rinok-ukrayini-trendi-ta-vikliki-2024-roku/> (дата звернення: 19.04.2026).
73. Skrupnyk A., Saiapin S. Information provision in advisory using modern innovative Internet technologies. *Ekonomika APK*. 2019. Vol. 26, No 12. P. 46-55. DOI: <https://doi.org/10.32317/2221-1055.201912046>.
74. Velychko I., Sidletskyi V. Development of a neural network model for an automated HVAC system based on collected data. *Technology Audit and Production Reserves*. 2025. Vol. 2, No 2(82). P. 21-26. DOI: <https://doi.org/10.15587/2706-5448.2025.326909>.
75. Onyelowe K. C., Ebid A. M., Hanandeh S., Kamchoom V., Awoyera P., Avudaiappan S. Modeling the compressive strength behavior of concrete reinforced with basalt fiber. *Scientific Reports*. 2025. Vol. 15., Art. 11493. DOI: <https://doi.org/10.1038/s41598-025-96343-6>.
76. Mehakarora. Demystifying Neural Networks: From a Single Neuron to Deep Learning. *Medium*. 2021. URL: <https://medium.com/@mehakarora3010/demystifying-neural-networks-from-a-single-neuron-to-deep-learning-9a560649ad03> (дата звернення: 19.04.2026).
77. Kumagai M., Karube K., Sato T., Ohisa N., Amano T., Kikuchi R., Ogawa N. A Near Infrared Spectroscopic Discrimination of Noodle Flours Using a Principal-

Component Analysis Coupled with Chemical Information. *Analytical Sciences*. 2002. Vol. 18, No 10. P. 1145-1150. DOI: <https://doi.org/10.2116/analsci.18.1145>.

78. Li J., Yin J., Deng L. A robot vision navigation method using deep learning in edge computing environment. *EURASIP Journal on Advances in Signal Processing*. 2021. Vol. 2021, No 1. Art. 22. URL: [https://www.researchgate.net/publication/351655136\\_A\\_robot\\_vision\\_navigation\\_method\\_using\\_deep\\_learning\\_in\\_edge\\_computing\\_environment](https://www.researchgate.net/publication/351655136_A_robot_vision_navigation_method_using_deep_learning_in_edge_computing_environment).

79. Kukuni T. G., Kotze B. Industrial Augmented Reality (IAR) as an Approach for Device Identification within a Manufacturing Plant for Property Alteration Purpose. *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. 2019. P. 1-9. URL: <https://www.researchgate.net/publication/334273689>.

80. Day I., The Newbie Guide to setting up a Jetson Nano on JP4.4. *Medium*. 2020. URL: <https://medium.com/swlh/the-newbie-guide-to-setting-up-a-jetson-nano-on-jp4-4-230449346258>.

81. Kang W., Chung J. Power- and Time-Aware Deep Learning Inference for Mobile Embedded Devices. *IEEE Access*. 2019. URL: <https://www.researchgate.net/publication/329739781>.

82. Федоров Є., Гуцалюк А. Перспективи застосування кіберфізичних систем контролю якості борошна на основі машинного навчання. *Перспективні мережеві та комп'ютерні технології*. Матеріали XVII Міжнародної студентській науково-технічній конференції (23 квітня 2026 р.). Харків: Національний аерокосмічний університет «Харківський авіаційний інститут», 2026.

83. Гуцалюк А. Концепція методу автоматизованого оцінювання якості борошна на основі машинного навчання. *Глобальні тенденції в науці: дослідження, інновації та розвиток*. Збірник наукових праць за матеріалами 3-ї Міжнародної науково-практичної конференції (27-29 квітня 2026 р.) Болгарія: European Open Science Space 2026.

## ДОДАТОК А (обов'язковий)

### Тези

УДК 004.93:004.8:664.64.016.8

Перспективи застосування кіберфізичних систем контролю якості борошна на основі машинного навчання

*Гуцалюк А., студент гр. КІ2М-24-1*

*Науковий керівник: д. т. н., професор Федоров Є.Є.,*

*Хмельницький національний університет*

**Актуальність.** Функціонування сучасних борошномельних підприємств України потребує переходу до систем безперервного контролю для забезпечення конкурентоспроможності на європейських ринках. Лабораторний аналіз характеризується значною часовою дискретністю, що робить складним миттєве реагування на критичні технічні збої, такі як порив сита розсіву.

Метою є підвищення ефективності контролю якості борошна шляхом розробки концепції кіберфізичної системи на основі машинного навчання.

**Аналіз рішень.** На сьогодні ринок пропонує переважно потокові аналізатори близького інфрачервоного діапазону, такі як системи Bühler MYRG або Perten DA 7300. Однак вони мають суттєві недоліки: висока ціна обмежує їх впровадження на більшості вітчизняних підприємств, а закрита архітектура виробників ускладнює інтеграцію. Важливим технічним обмеженням технології спектрометрії в близькій ІЧ-області є нездатність ідентифікувати морфологічні дефекти або дрібні сторонні домішки, які не змінюють загальний спектральний підпис продукту, але критично впливають на його безпеку.

**Результати.** Запропоновано концепцію системи на базі периферійного обчислювального вузла з використанням колірному простору CIELAB та оптимізованих нейромережових архітектур. Використання такої системи у перспективі дозволить оптимізувати виробництво за рахунок автономного виявлення аномалій за допомогою згорткових автоенкодерів та швидкої реакції на відхилення показників якості без потреби в хмарних ресурсах.

**Висновки.** Впровадження кіберфізичної системи дозволить перейти до безперервного цифрового моніторингу, мінімізуючи вплив людського фактора.

**ДОДАТОК Б**  
(обов'язковий)

Тези

**КОНЦЕПЦІЯ МЕТОДУ АВТОМАТИЗОВАНОГО  
ОЦІНЮВАННЯ ЯКОСТІ БОРОШНА НА ОСНОВІ  
МАШИННОГО НАВЧАННЯ**

**Гуцалюк Артем**

здобувач вищої освіти магістерського рівня  
Комп'ютерної інженерії та інформаційних систем  
Хмельницький національний університет, Україна

**Анотація.** Представлено метод та апаратно-програмну реалізацію кіберфізичної системи для безперервного контролю якості борошна. Запропонований підхід базується на використанні периферійних обчислень та паралельному аналізі даних за допомогою згорткових нейронних мереж. Виконано оптимізацію моделей через TensorRT, що забезпечило роботу системи в режимі жорсткого реального часу із затримкою до 36,8 мс. Результати експериментів підтвердили підвищення ефективності контролю на 14,2 % порівняно з традиційними методами.

**Ключові слова:** кіберфізична система, машинне навчання, комп'ютерний зір, Edge Computing, NVIDIA Jetson, TensorRT, ідентифікація якості борошна, згортковий автоенкодер.

**Вступ.** Сучасна концепція Industry 4.0 вимагає від харчової промисловості впровадження систем інтелектуального моніторингу, здатних працювати безпосередньо в технологічному потоці. Традиційні лабораторні методи аналізу борошна є дискретними та тривалими, а існуючі промислові рішення часто мають закриту архітектуру та високу вартість. Актуальним завданням є створення доступних КФС, які поєднують комп'ютерний зір та алгоритми

машинного навчання для миттєвого виявлення відхилень сортності та сторонніх домішок без зупинки виробництва.

**Метою роботи** є підвищення ефективності контролю якості борошна шляхом розробки КФС, що забезпечує автоматизований збір та аналіз візуальних і технологічних даних із використанням методів машинного навчання. Для досягнення мети вирішено **наступні завдання**:

- Проаналізовано сучасні методи та засоби автоматизованого контролю якості.
- Обґрунтовано архітектуру КФС та обрано оптимальні моделі нейромережевого аналізу.
- Розроблено метод комплексного оцінювання якості продукції.
- Створено підсистему збору даних та алгоритми формування вхідної інформації.
- Проведено дослідження ефективності розробленого методу та апаратно-програмного комплексу.

**Результати.** Запропонований метод базується на інтеграції візуальних та технологічних даних із подальшим їх аналізом у межах єдиного обчислювального процесу. Вхідна інформація формується шляхом раннього комплексування ознак у колірному просторі CIELAB разом із технологічними параметрами, такими як швидкість руху продукту та рівень освітленості. Це дозволяє підвищити інформативність вхідного простору ознак і забезпечити стійкість методу до зовнішніх впливів.

Метод реалізує паралельну обробку даних за допомогою двох взаємодоповнюючих моделей. Перша модель на основі архітектури MobileNetV2 виконує класифікацію сортності з точністю 97,4 %, аналізуючи текстурні та колірні характеристики борошна. Друга модель, представлена згортковим автоенкодером, забезпечує некероване виявлення морфологічних аномалій. Поява сторонніх включень призводить до зростання похибки реконструкції, що дозволяє виявляти дефекти з показником повноти 93,0 %.

Ефективність методу підвищується за рахунок оптимізації моделей. Застосування квантування ваг до формату INT8 та використання бібліотеки TensorRT дозволило зменшити обсяг моделі з 14 МБ до 3,4 МБ без суттєвої втрати точності. У результаті забезпечено обробку даних у режимі реального часу із затримкою 36,8 мс, що є достатнім для роботи при швидкості потоку до 1,5 м/с. Додатково реалізовано механізм адаптації порогових значень, який дозволяє підтримувати точність на рівні 92,5 % за умов змінного освітлення.

Отримані результати свідчать, що запропонований метод дозволяє суттєво підвищити ефективність контролю якості борошна. Зокрема, досягається скорочення часу аналізу, підвищення продуктивності та зменшення впливу людського фактора. Порівняно з традиційними підходами ефективність контролю зростає на 14,2 %.

**Висновок.** Таким чином, запропонований метод є адаптивним і масштабованим рішенням, що відповідає сучасним глобальним тенденціям

впровадження штучного інтелекту та технологій Edge AI в рамках концепції Industry 4.0. Його використання створює передумови для подальшого розвитку інтелектуальних систем контролю якості у харчовій промисловості.

#### **Список використаних джерел**

1. Howard A. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications / A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam // arXiv preprint. – 2017. - P. 4510–4520.
2. Bergmann P. Uninformed Students: Student-Teacher Anomaly Detection with Discriminative Latent Embeddings / P. Bergmann, M. Fauser, D. Sattlegger, C. Steger // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2020. - P. 4183–4192.

**DOI 10.70286/EOSS-27.04.2026.005.131-133**

## ДОДАТОК В (обов'язковий)

### Презентація

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
Кафедра комп'ютерної інженерії та інформаційних систем



### Кіберфізична система збору та аналізу якості борошна на підставі машинного навчання

Здобувач: Артем Гуцалюк

Науковий керівник: д.т.н. проф. Євген Федоров

Хмельницький - 2026

### МЕТА ДОСЛІДЖЕННЯ

Мета роботи полягає у підвищенні ефективності контролю якості борошна за допомогою кіберфізичної системи, яка автоматично збирає та аналізує дані.

Об'єктом дослідження є процес автоматизованого контролю показників якості борошна.

Предметом дослідження є метод комплексного оцінювання якості борошна, моделі машинного навчання та апаратно-програмні засоби кіберфізичної системи для збору й аналізу даних.

## ЗАДАЧІ ДОСЛІДЖЕННЯ

Поставлена мета досягається розв'язанням таких основних завдань:

- аналіз сучасних методів і засобів автоматизованого контролю якості борошна;
- обґрунтування структури кіберфізичної системи та вибір моделей МН;
- розробка методу комплексного оцінювання якості борошна;
- розробка підсистеми збору даних та формування вхідної інформації;
- дослідження ефективності розробленого методу та системи.

## НАУКОВА НОВИЗНА ТА ПРАКТИЧНА ЦІННІСТЬ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Наукова новизна отриманих результатів:

- набув подальшого розвитку метод комплексного оцінювання якості борошна шляхом поєднання візуальних даних і параметрів технологічного процесу в межах єдиної ML-моделі;
- набула подальшого розвитку інформаційна технологія автоматизованого збору, комплексування та інтелектуального аналізу даних для контролю якості борошна в режимі реального часу.

Практична цінність отриманих результатів полягає у розробленні кіберфізичної системи, що забезпечує автоматизований контроль якості борошна на основі аналізу візуальних даних та методів машинного навчання в режимі реального часу.

## АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ

Сучасний контроль якості борошна на підприємствах здебільшого виконується за допомогою лабораторних методів. Вони забезпечують високу точність, але потребують значного часу на відбір і аналіз проб, тому не дозволяють отримувати результат у режимі реального часу. У виробничих умовах це ускладнює своєчасне виявлення відхилень у властивостях сировини, а також знижує ефективність управління технологічним процесом.

Окремою проблемою є вплив людського фактора, який може призводити до помилок під час оцінювання якості. Існуючі потокові аналізатори забезпечують автоматизацію контролю, однак вони є дорогими, складними в інтеграції та не завжди придатні для гнучкого використання на різних виробничих лініях.

Крім того, більшість таких систем орієнтована лише на окремі фізико-хімічні показники. У зв'язку з цим актуальним є розроблення доступної кіберфізичної системи, яка поєднує машинне навчання, комп'ютерний зір і автоматизований збір даних для безперервного контролю якості борошна в реальному часі.

## АНАЛІЗ ВІДОМИХ МЕТОДІВ

Для контролю якості борошна використовуються лабораторні, потокові та автоматизовані методи. Лабораторні підходи є точними, але повільними та не забезпечують безперервного моніторингу.

Потокові аналізатори працюють швидше і в реальному часі, однак мають високу вартість, складні в інтеграції та обмежені у функціональності, зосереджуючись переважно на фізико-хімічних показниках.


Методи машинного навчання дозволяють автоматизувати аналіз візуальних даних і підвищити оперативність контролю, що робить їх перспективними для побудови сучасних систем контролю якості борошна.



## МЕТОД КІБЕРФІХИСНОЇ СИСТЕМИ ЗБОРУ ТА АНАЛІЗУ ЯКОСТІ БОРОШНА НА ПІДСТАВІ МАШИННОГО НАВЧАННЯ

Метод призначений для автоматизованого оцінювання якості борошна в режимі реального часу з використанням методів машинного навчання. Його особливістю є комплексний підхід, який передбачає одночасний аналіз візуальних характеристик продукту та виявлення можливих структурних відхилень.

На відміну від традиційних підходів, метод не обмежується одним критерієм оцінювання, а поєднує декілька незалежних джерел інформації. Це дозволяє підвищити достовірність контролю та зменшити ризик пропуску дефектної продукції.



## МЕТОД КІБЕРФІХИСНОЇ СИСТЕМИ ЗБОРУ ТА АНАЛІЗУ ЯКОСТІ БОРОШНА НА ПІДСТАВІ МАШИННОГО НАВЧАННЯ

Робота методу реалізується як послідовність взаємопов'язаних етапів.

1. На першому етапі здійснюється збір вхідних даних, зокрема зображень потоку борошна та параметрів технологічного процесу.
2. Далі виконується попередня обробка та узгодження даних, що дозволяє сформувати єдине представлення для подальшого аналізу.
3. Після цього дані подаються у два незалежні обчислювальні тракти.
4. Перший тракт виконує класифікацію та визначає сортність продукту.
5. Другий тракт аналізує структуру і виявляє можливі аномалії.
6. На наступному етапі результати обох трактів об'єднуються, і формується фінальне рішення щодо якості продукту.
7. Завершальним етапом є збереження результатів аналізу для подальшого використання та адаптації системи.

## МЕТОД КІБЕРФІЗИСНОЇ СИСТЕМИ ЗБОРУ ТА АНАЛІЗУ ЯКОСТІ БОРОШНА НА ПІДСТАВІ МАШИННОГО НАВЧАННЯ

### Схема методу



## МЕТОД КІБЕРФІЗИСНОЇ СИСТЕМИ ЗБОРУ ТА АНАЛІЗУ ЯКОСТІ БОРОШНА НА ПІДСТАВІ МАШИННОГО НАВЧАННЯ

Реалізація представленого методу базується на використанні двох взаємодоповнюючих нейромережових моделей, які відповідають окремим обчислювальним трактам, показаним на схемі.

- Перший тракт реалізує задачу класифікації та визначає сортність борошна на основі вхідних ознак, отриманих із зображення та технологічних параметрів. Результатом його роботи є оцінка належності продукту до певного класу.
- Другий тракт призначений для виявлення аномалій і аналізує структурні особливості продукту. Він дозволяє виявляти сторонні вclusions, неоднорідності та інші відхилення від нормального стану.

Паралельна робота цих двох моделей забезпечує більш повну оцінку якості продукту. Поєднання результатів класифікації та аналізу аномалій дозволяє підвищити достовірність прийняття рішення та зменшити ймовірність помилкових спрацювань системи.

## РЕАЛІЗАЦІЯ СИСТЕМИ

### Концептуальна модель кіберфізичної системи

Система реалізована у вигляді трирівневої архітектури:

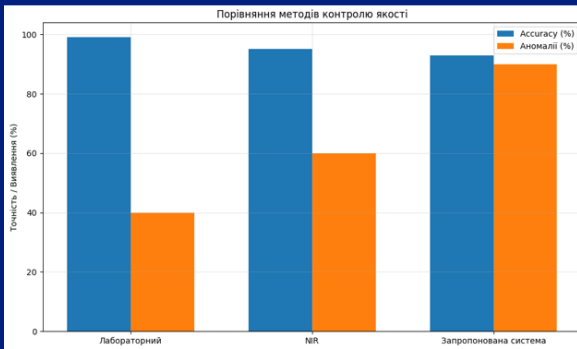
- Нижній рівень забезпечує отримання вхідних даних та виконання керуючих дій.
- Середній рівень виконує обробку зображень і роботу моделей машинного навчання.
- Верхній рівень відповідає за зберігання даних, навчання моделей та аналітику.

Така структура забезпечує розподіл функцій і дозволяє системі працювати в режимі реального часу.

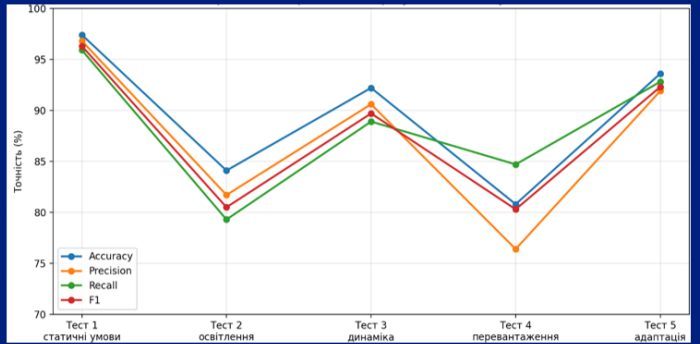


## ЕКСПЕРИМЕНТИ

### Порівняння з існуючими методами



### Результати тестів за різних умов





## ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень:

- виконано аналіз сучасних методів і засобів автоматизованого контролю якості борошна та визначено їх обмеження;
- обґрунтовано структуру кіберфізичної системи та здійснено вибір моделей машинного навчання;
- розроблено метод комплексного оцінювання якості борошна на основі поєднання класифікації та виявлення аномалій;
- розроблено підсистему збору даних та формування вхідної інформації;
- проведено дослідження ефективності розробленого методу та системи.

Запропонований метод забезпечує підвищення ефективності контролю якості борошна за рахунок поєднання високої швидкодії та покращеного виявлення структурних аномалій. Досягнуто підвищення ефективності на 8–12% та рівня виявлення аномалій до 90%, що підтверджує ефективність запропонованого рішення.



## ПУБЛІКАЦІЇ

- Федоров Є., Гуцалюк А. Перспективи застосування кіберфізичних систем контролю якості борошна на основі машинного навчання. *Перспективні мережеві та комп'ютерні технології*. Матеріали XVII Міжнародної студентської науково-технічної конференції (23 квітня 2026 р.). Харків: Національний аерокосмічний університет «Харківський авіаційний інститут», 2026. С. 1.

## ДОДАТОК Г

(НЕ ОБОВ'ЯЗКОВИЙ)

### Main.py

```

import cv2
import numpy as np
import time
import logging
from pathlib import Path
from dataclasses import dataclass, field
from typing import Optional

from image_processor import ImagePreprocessor
from classifier import MobileNetClassifier
from anomaly_detector import AutoencoderAnomalyDetector
from decision_engine import DecisionEngine, QualityDecision
from visualizer import ResultVisualizer

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s [%(levelname)s] %(name)s: %(message)s"
)
logger = logging.getLogger("FlourQC")

@dataclass
class SystemConfig:
    # Camera
    camera_index: int = 0
    frame_width: int = 640
    frame_height: int = 480
    target_fps: int = 30

    # Model paths (set to None to use auto-built demo models)
    classifier_model_path: Optional[str] = None
    autoencoder_model_path: Optional[str] = None

    # Decision thresholds
    anomaly_threshold: float = 0.015 # reconstruction MSE threshold
    classification_confidence: float = 0.6 # min confidence for class label

    # Processing
    input_size: tuple = (224, 224)
    enable_preprocessing: bool = True
    enable_visualizer: bool = True

class FlourQCSystem:

    def __init__(self, config: SystemConfig):
        self.config = config
        self.preprocessor = ImagePreprocessor(target_size=config.input_size)
        self.classifier = MobileNetClassifier(
            model_path=config.classifier_model_path,
            input_size=config.input_size,
            confidence_threshold=config.classification_confidence,
        )
        self.anomaly_detector = AutoencoderAnomalyDetector(
            model_path=config.autoencoder_model_path,

```

```

        input_size=config.input_size,
        threshold=config.anomaly_threshold,
    )
    self.decision_engine = DecisionEngine()
    self.visualizer = ResultVisualizer() if config.enable_visualizer else None

    self._frame_times = []
    logger.info("FlourQCSystem initialised.")

def process_frame(self, frame: np.ndarray) -> QualityDecision:
    """Run the full pipeline on a single BGR frame."""
    t0 = time.perf_counter()

    # 1. Pre-process
    processed = self.preprocessor.process(frame) if self.config.enable_preprocessing
else frame
    tensor = self._to_model_tensor(processed)

    # 2. Classify + detect anomaly (independent paths)
    cls_result = self.classifier.predict(tensor)
    anm_result = self.anomaly_detector.predict(tensor)

    # 3. Fuse & decide
    decision = self.decision_engine.decide(cls_result, anm_result)

    # 4. Overlay
    if self.visualizer:
        self.visualizer.draw(frame, processed, cls_result, anm_result, decision)

    self._track_fps(t0)
    return decision

def run_camera(self):
    """Capture from camera in a loop."""
    cap = cv2.VideoCapture(self.config.camera_index)
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, self.config.frame_width)
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, self.config.frame_height)
    cap.set(cv2.CAP_PROP_FPS, self.config.target_fps)

    logger.info("Camera stream started. Press 'q' to quit.")
    try:
        while True:
            ret, frame = cap.read()
            if not ret:
                logger.warning("Failed to grab frame.")
                break

            decision = self.process_frame(frame)

            fps = self._current_fps()
            cv2.putText(frame, f"FPS: {fps:.1f}", (10, frame.shape[0] - 10),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.6, (200, 200, 200), 1)

            cv2.imshow("Flour QC System", frame)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
    finally:
        cap.release()
        cv2.destroyAllWindows()
        logger.info("Camera stream stopped.")

```

```

def run_on_image(self, image_path: str) -> QualityDecision:
    """Process a single image file."""
    frame = cv2.imread(image_path)
    if frame is None:
        raise FileNotFoundError(f"Cannot read image: {image_path}")
    return self.process_frame(frame)

def _to_model_tensor(self, frame: np.ndarray) -> np.ndarray:
    """Resize + normalise to float32 [0,1] with batch dim."""
    resized = cv2.resize(frame, self.config.input_size)
    rgb = cv2.cvtColor(resized, cv2.COLOR_BGR2RGB)
    return (rgb.astype(np.float32) / 255.0)[np.newaxis]

def _track_fps(self, t0: float):
    self._frame_times.append(time.perf_counter() - t0)
    if len(self._frame_times) > 30:
        self._frame_times.pop(0)

def _current_fps(self) -> float:
    if not self._frame_times:
        return 0.0
    avg = sum(self._frame_times) / len(self._frame_times)
    return 1.0 / avg if avg > 0 else 0.0

if __name__ == "__main__":
    import argparse

    parser = argparse.ArgumentParser(description="Flour Quality Control System")
    parser.add_argument("--image", type=str, help="Path to a single image (skips camera)")
    parser.add_argument("--camera", type=int, default=0)
    parser.add_argument("--no-display", action="store_true")
    args = parser.parse_args()

    cfg = SystemConfig(
        camera_index=args.camera,
        enable_visualizer=not args.no_display,
    )
    system = FlourQCSystem(cfg)

    if args.image:
        result = system.run_on_image(args.image)
        print(result)
    else:
        system.run_camera()

```

## anomaly\_detector.py

```

import numpy as np
import logging
import time
from dataclasses import dataclass
from typing import Optional

logger = logging.getLogger("FlourQC.AnomalyDetector")

# Internal resolution for the autoencoder (trade-off: speed vs detail)
AE_SIZE = (64, 64)

```

```

@dataclass
class AnomalyResult:
    reconstruction_error: float # MSE between input and reconstruction
    is_anomaly: bool
    threshold: float
    anomaly_map: Optional[np.ndarray] # per-pixel error map (HxWx1 float32) or None
    inference_ms: float

class AutoencoderAnomalyDetector:

    def __init__(
        self,
        model_path: Optional[str] = None,
        input_size: tuple = (224, 224),
        threshold: float = 0.015,
        latent_dim: int = 128,
    ):
        self.input_size = input_size
        self.threshold = threshold
        self.latent_dim = latent_dim
        self._model = None
        self._use_stub = False

        self._load_model(model_path)

    def _load_model(self, model_path: Optional[str]):
        try:
            import tensorflow as tf

            if model_path:
                logger.info("Loading autoencoder from %s", model_path)
                self._model = tf.keras.models.load_model(model_path)
            else:
                logger.info("No autoencoder path given - building demo model.")
                self._model = self._build_autoencoder()

            # Warm-up
            dummy = np.zeros((1, *AE_SIZE, 3), dtype=np.float32)
            self._model(dummy, training=False)
            logger.info("Autoencoder ready (threshold=%.4f).", self.threshold)

        except ImportError:
            logger.warning("TensorFlow not installed - anomaly detector in STUB mode.")
            self._use_stub = True
        except Exception as exc:
            logger.error("Autoencoder load failed (%s) - falling back to STUB.", exc)
            self._use_stub = True

    def _build_autoencoder(self):
        """Lightweight conv-autoencoder suitable for real-time inference."""
        import tensorflow as tf
        from tensorflow.keras import layers

```

```

h, w = AE_SIZE

# ---- Encoder ----
inp = tf.keras.Input(shape=(h, w, 3), name="ae_input")
x = layers.Conv2D(32, 3, activation="relu", padding="same", strides=2)(inp)
x = layers.Conv2D(64, 3, activation="relu", padding="same", strides=2)(x)
x = layers.Conv2D(128, 3, activation="relu", padding="same", strides=2)(x)
enc_shape = x.shape[1:]
x = layers.Flatten()(x)
bottleneck = layers.Dense(self.latent_dim, activation="relu",
name="bottleneck")(x)

# ---- Decoder ----
x = layers.Dense(int(np.prod(enc_shape)), activation="relu")(bottleneck)
x = layers.Reshape(enc_shape)(x)
x = layers.Conv2DTranspose(128, 3, activation="relu", padding="same",
strides=2)(x)
x = layers.Conv2DTranspose(64, 3, activation="relu", padding="same", strides=2)(x)
x = layers.Conv2DTranspose(32, 3, activation="relu", padding="same", strides=2)(x)
out = layers.Conv2DTranspose(3, 3, activation="sigmoid", padding="same",
name="ae_output")(x)

return tf.keras.Model(inp, out, name="flour_autoencoder")

def predict(self, tensor: np.ndarray) -> AnomalyResult:
    """
    Args:
        tensor: float32 (1, H, W, 3) normalised [0,1].
    Returns:
        AnomalyResult
    """
    t0 = time.perf_counter()

    # Downscale to AE internal resolution
    import cv2
    small = cv2.resize(tensor[0], AE_SIZE[::-1])[np.newaxis]

    if self._use_stub:
        recon_error, anomaly_map = self._stub_result()
    else:
        recon = self._model(small, training=False).numpy()
        diff = (small[0] - recon[0]) ** 2 # (H, W, 3) per-pixel squared error
        recon_error = float(diff.mean())
        anomaly_map = diff.mean(axis=-1, keepdims=True).astype(np.float32) # (H, W,

1)
        # Upscale map back to original resolution
        anomaly_map = cv2.resize(
            anomaly_map[:, :, 0],
            self.input_size[::-1],
        )[:, :, np.newaxis]

```

```

elapsed_ms = (time.perf_counter() - t0) * 1000

return AnomalyResult(
    reconstruction_error=recon_error,
    is_anomaly=recon_error > self.threshold,
    threshold=self.threshold,
    anomaly_map=anomaly_map if not self._use_stub else None,
    inference_ms=elapsed_ms,
)

def _stub_result(self):
    error = float(np.random.uniform(0.005, 0.030))
    return error, None

```

## classifier.py

```

import numpy as np
import logging
from dataclasses import dataclass, field
from typing import Optional, List
import time

logger = logging.getLogger("FlourQC.Classifier")

CLASS_LABELS = ["premium_grade", "standard_grade", "rejected"]

@dataclass
class ClassificationResult:
    label: str
    class_id: int
    confidence: float
    all_scores: List[float]
    inference_ms: float

class MobileNetClassifier:

    def __init__(
        self,
        model_path: Optional[str] = None,
        input_size: tuple = (224, 224),
        confidence_threshold: float = 0.6,
        num_classes: int = 3,
    ):
        self.input_size = input_size
        self.confidence_threshold = confidence_threshold
        self.num_classes = num_classes
        self._model = None
        self._use_stub = False

        self._load_model(model_path)

    def _load_model(self, model_path: Optional[str]):
        try:

```

```

import tensorflow as tf

if model_path:
    logger.info("Loading classifier from %s", model_path)
    self._model = tf.keras.models.load_model(model_path)
else:
    logger.info("No classifier path given - building demo MobileNetV2 model.")
    self._model = self._build_demo_model()

# Warm-up pass
dummy = np.zeros((1, *self.input_size, 3), dtype=np.float32)
self._model(dummy, training=False)
logger.info("Classifier ready.")

except ImportError:
    logger.warning("TensorFlow not installed - classifier running in STUB mode.")
    self._use_stub = True
except Exception as exc:
    logger.error("Classifier load failed (%s) - falling back to STUB.", exc)
    self._use_stub = True

def _build_demo_model(self):
    """Build a fine-tunable MobileNetV2 head."""
    import tensorflow as tf

    base = tf.keras.applications.MobileNetV2(
        input_shape=(*self.input_size, 3),
        include_top=False,
        weights=None,          # random for demo; use "imagenet" for transfer learning
    )
    base.trainable = False
    inputs = tf.keras.Input(shape=(*self.input_size, 3))
    x = base(inputs, training=False)
    x = tf.keras.layers.GlobalAveragePooling2D()(x)
    x = tf.keras.layers.Dropout(0.3)(x)
    outputs = tf.keras.layers.Dense(self.num_classes, activation="softmax")(x)
    return tf.keras.Model(inputs, outputs, name="flour_mobilenet")

def predict(self, tensor: np.ndarray) -> ClassificationResult:
    """
    Args:
        tensor: float32 array shaped (1, H, W, 3) normalised to [0, 1].
    Returns:
        ClassificationResult
    """
    t0 = time.perf_counter()

    if self._use_stub:
        scores = self._stub_scores()
    else:
        scores = self._model(tensor, training=False).numpy()[0].tolist()

    elapsed_ms = (time.perf_counter() - t0) * 1000

```

```

class_id = int(np.argmax(scores))
confidence = float(scores[class_id])

if confidence < self.confidence_threshold:
    label = "uncertain"
else:
    label = CLASS_LABELS[class_id] if class_id < len(CLASS_LABELS) else
f"class_{class_id}"

return ClassificationResult(
    label=label,
    class_id=class_id,
    confidence=confidence,
    all_scores=scores,
    inference_ms=elapsed_ms,
)

def _stub_scores(self) -> List[float]:
    """Deterministic random scores for testing without TF."""
    raw = np.random.dirichlet(np.ones(self.num_classes) * 2)
    return raw.tolist()

```

## decision\_engine.py

```

import logging
from dataclasses import dataclass, field
from typing import Dict, Any
import time

from classifier import ClassificationResult
from anomaly_detector import AnomalyResult

logger = logging.getLogger("FlourQC.Decision")

# Map class labels -> initial verdict before anomaly override
_LABEL_VERDICT = {
    "premium_grade": "ACCEPT",
    "standard_grade": "WARNING",
    "rejected": "REJECT",
    "uncertain": "WARNING",
}

@dataclass
class QualityDecision:
    verdict: str # "ACCEPT" | "WARNING" | "REJECT"
    reason: str
    confidence_score: float # composite 0-1 (1 = perfect quality)
    cls: ClassificationResult
    anm: AnomalyResult
    timestamp: float = field(default_factory=time.time)
    metadata: Dict[str, Any] = field(default_factory=dict)

    def __str__(self):

```

```

return (
    f"[{self.verdict}] score={self.confidence_score:.3f} "
    f"class={self.cls.label}({self.cls.confidence:.2f}) "
    f"anomaly_err={self.anm.reconstruction_error:.4f} "
    f"reason={self.reason}"
)

class DecisionEngine:

    def __init__(
        self,
        anomaly_weight: float = 0.5,
        classification_weight: float = 0.5,
        escalation_rules: Dict[str, str] = None,
    ):
        assert abs(anomaly_weight + classification_weight - 1.0) < 1e-6, \
            "Weights must sum to 1.0"
        self.anomaly_weight = anomaly_weight
        self.classification_weight = classification_weight
        self.escalation_rules = escalation_rules or {}

    def decide(
        self,
        cls_result: ClassificationResult,
        anm_result: AnomalyResult,
    ) -> QualityDecision:

        # --- Stage 1: start from classification verdict -----
        label = cls_result.label
        verdict = _LABEL_VERDICT.get(label, "WARNING")
        reason = f"Classifier: {label} ({cls_result.confidence:.0%})"

        # --- Stage 2: anomaly gate -----
        if anm_result.is_anomaly:
            verdict = "REJECT"
            reason = (
                f"Anomaly detected (err={anm_result.reconstruction_error:.4f} > "
                f"thresh={anm_result.threshold:.4f}); class was {label}"
            )

        # --- Stage 3: custom escalation rules -----
        if label in self.escalation_rules:
            forced = self.escalation_rules[label]
            if self._severity(forced) > self._severity(verdict):
                verdict = forced
                reason += f" | escalation rule -> {forced}"

        # --- Stage 4: composite quality score (higher = better) -----
        # Classifier component: premium->1, standard->0.5, rejected/uncertain->0
        cls_score = {"premium_grade": 1.0, "standard_grade": 0.5}.get(label, 0.0)
        cls_score *= cls_result.confidence # scale by confidence

        # Anomaly component: normalise error to 0-1, invert (lower error = higher score)

```

```

max_expected_error = anm_result.threshold * 3.0
anm_score = max(0.0, 1.0 - (anm_result.reconstruction_error / max_expected_error))

composite = (
    self.classification_weight * cls_score
    + self.anomaly_weight * anm_score
)
composite = float(max(0.0, min(1.0, composite)))

decision = QualityDecision(
    verdict=verdict,
    reason=reason,
    confidence_score=composite,
    cls=cls_result,
    anm=anm_result,
    metadata={
        "cls_score": cls_score,
        "anm_score": anm_score,
        "cls_inference_ms": cls_result.inference_ms,
        "anm_inference_ms": anm_result.inference_ms,
    },
)

logger.debug(decision)
return decision

@staticmethod
def _severity(verdict: str) -> int:
    return {"ACCEPT": 0, "WARNING": 1, "REJECT": 2}.get(verdict, 0)

```

## image\_processor.py

```

import cv2
import numpy as np
import logging
from dataclasses import dataclass

logger = logging.getLogger("FlourQC.Preprocessor")

@dataclass
class PreprocessorConfig:
    # Denoising
    denoise_method: str = "gaussian" # "gaussian" | "median" | "bilateral" | "none"
    gaussian_ksize: int = 3
    median_ksize: int = 3
    bilateral_d: int = 9
    bilateral_sigma_color: float = 75
    bilateral_sigma_space: float = 75

    # Lighting / contrast
    clahe_enabled: bool = True
    clahe_clip_limit: float = 2.0
    clahe_tile_grid: tuple = (8, 8)

```

```

# White balance
white_balance: bool = True

# Sharpening (unsharp mask)
sharpen: bool = True
sharpen_amount: float = 0.5

class ImagePreprocessor:

    def __init__(self, target_size: tuple = (224, 224),
                 config: PreprocessorConfig = None):
        self.target_size = target_size
        self.cfg = config or PreprocessorConfig()
        self._clahe = cv2.createCLAHE(
            clipLimit=self.cfg.clahe_clip_limit,
            tileGridSize=self.cfg.clahe_tile_grid,
        )
        logger.debug("ImagePreprocessor ready (size=%s, denoise=%s, clahe=%s)",
                    target_size, self.cfg.denoise_method, self.cfg.clahe_enabled)

    def process(self, frame: np.ndarray) -> np.ndarray:
        """Apply the full pipeline and return a cleaned BGR image (same size as input)."""
        img = frame.copy()

        img = self._denoise(img)
        img = self._correct_lighting(img)
        if self.cfg.white_balance:
            img = self._white_balance(img)
        if self.cfg.sharpen:
            img = self._unsharp_mask(img)

        return img

    def _denoise(self, img: np.ndarray) -> np.ndarray:
        method = self.cfg.denoise_method
        if method == "gaussian":
            k = self.cfg.gaussian_ksize | 1 # must be odd
            return cv2.GaussianBlur(img, (k, k), 0)
        elif method == "median":
            k = self.cfg.median_ksize | 1
            return cv2.medianBlur(img, k)
        elif method == "bilateral":
            return cv2.bilateralFilter(
                img,
                self.cfg.bilateral_d,
                self.cfg.bilateral_sigma_color,
                self.cfg.bilateral_sigma_space,
            )
        return img # "none"

    def _correct_lighting(self, img: np.ndarray) -> np.ndarray:
        """CLAHE applied per-channel in LAB colour space (preserves hue)."""

```

```

if not self.cfg.clahe_enabled:
    return img
lab = cv2.cvtColor(img, cv2.COLOR_BGR2LAB)
l, a, b = cv2.split(lab)
l = self._clahe.apply(l)
lab = cv2.merge([l, a, b])
return cv2.cvtColor(lab, cv2.COLOR_LAB2BGR)

def _white_balance(self, img: np.ndarray) -> np.ndarray:
    """Simple grey-world white balance."""
    result = img.astype(np.float32)
    avg_b = np.mean(result[:, :, 0])
    avg_g = np.mean(result[:, :, 1])
    avg_r = np.mean(result[:, :, 2])
    avg_gray = (avg_b + avg_g + avg_r) / 3.0
    if avg_b > 0:
        result[:, :, 0] *= avg_gray / avg_b
    if avg_g > 0:
        result[:, :, 1] *= avg_gray / avg_g
    if avg_r > 0:
        result[:, :, 2] *= avg_gray / avg_r
    return np.clip(result, 0, 255).astype(np.uint8)

def _unsharp_mask(self, img: np.ndarray) -> np.ndarray:
    """Unsharp masking to enhance texture details."""
    blurred = cv2.GaussianBlur(img, (0, 0), 3)
    amount = self.cfg.sharpen_amount
    sharpened = cv2.addWeighted(img, 1 + amount, blurred, -amount, 0)
    return sharpened

def extract_features(self, img: np.ndarray) -> dict:
    """Return basic statistical features useful for diagnostics."""
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    return {
        "mean_intensity": float(np.mean(gray)),
        "std_intensity": float(np.std(gray)),
        "laplacian_var": float(cv2.Laplacian(gray, cv2.CV_64F).var()), # blur metric
    }

```

## test\_flour\_qc.py

```

import unittest
import numpy as np
import sys
import os

# Add parent directory to path so imports resolve
sys.path.insert(0, os.path.dirname(__file__))

def make_frame(h=480, w=640, bright=True, noisy=False) -> np.ndarray:
    """Create a synthetic BGR frame simulating flour on a conveyor belt."""
    base = 210 if bright else 90
    frame = np.full((h, w, 3), base, dtype=np.uint8)

```

```

if noisy:
    noise = np.random.randint(-30, 30, frame.shape, dtype=np.int16)
    frame = np.clip(frame.astype(np.int16) + noise, 0, 255).astype(np.uint8)
return frame

def make_tensor(size=(224, 224)) -> np.ndarray:
    """Float32 normalised batch tensor."""
    return np.random.rand(1, *size, 3).astype(np.float32)

class TestImagePreprocessor(unittest.TestCase):

    def setUp(self):
        from image_processor import ImagePreprocessor
        self.pp = ImagePreprocessor(target_size=(224, 224))

    def test_output_shape_preserved(self):
        frame = make_frame()
        out = self.pp.process(frame)
        self.assertEqual(out.shape, frame.shape)

    def test_dtype_preserved(self):
        frame = make_frame()
        out = self.pp.process(frame)
        self.assertEqual(out.dtype, np.uint8)

    def test_noisy_frame(self):
        frame = make_frame(noisy=True)
        out = self.pp.process(frame)
        self.assertEqual(out.shape, frame.shape)

    def test_dark_frame(self):
        frame = make_frame(bright=False)
        out = self.pp.process(frame)
        # CLAHE should increase mean brightness somewhat
        self.assertGreaterEqual(out.mean(), frame.mean() - 10) # tolerant check

    def test_feature_extraction_keys(self):
        frame = make_frame()
        feats = self.pp.extract_features(frame)
        self.assertIn("mean_intensity", feats)
        self.assertIn("std_intensity", feats)
        self.assertIn("laplacian_var", feats)

class TestClassifier(unittest.TestCase):

    def setUp(self):
        from classifier import MobileNetClassifier
        # Force stub mode by passing an invalid path
        self.clf = MobileNetClassifier(model_path=None)

    def test_predict_returns_result(self):
        from classifier import ClassificationResult
        tensor = make_tensor()

```

```

    result = self.clf.predict(tensor)
    self.assertIsInstance(result, ClassificationResult)

def test_confidence_in_range(self):
    tensor = make_tensor()
    for _ in range(20):
        result = self.clf.predict(tensor)
        self.assertGreaterEqual(result.confidence, 0.0)
        self.assertLessEqual(result.confidence, 1.0)

def test_all_scores_sum_to_one(self):
    tensor = make_tensor()
    result = self.clf.predict(tensor)
    self.assertAlmostEqual(sum(result.all_scores), 1.0, places=5)

def test_inference_time_positive(self):
    tensor = make_tensor()
    result = self.clf.predict(tensor)
    self.assertGreater(result.inference_ms, 0)

class TestAnomalyDetector(unittest.TestCase):

    def setUp(self):
        from anomaly_detector import AutoencoderAnomalyDetector
        self.det = AutoencoderAnomalyDetector(model_path=None, threshold=0.015)

    def test_predict_returns_result(self):
        from anomaly_detector import AnomalyResult
        tensor = make_tensor()
        result = self.det.predict(tensor)
        self.assertIsInstance(result, AnomalyResult)

    def test_reconstruction_error_positive(self):
        tensor = make_tensor()
        for _ in range(10):
            result = self.det.predict(tensor)
            self.assertGreater(result.reconstruction_error, 0)

    def test_is_anomaly_boolean(self):
        tensor = make_tensor()
        result = self.det.predict(tensor)
        self.assertIsInstance(result.is_anomaly, bool)

    def test_threshold_stored(self):
        tensor = make_tensor()
        result = self.det.predict(tensor)
        self.assertAlmostEqual(result.threshold, 0.015)

class TestDecisionEngine(unittest.TestCase):

    def _make_cls(self, label="premium_grade", confidence=0.95):
        from classifier import ClassificationResult
        scores = {"premium_grade": 0.0, "standard_grade": 0.0, "rejected": 0.0}

```

```

scores[label] = confidence
return ClassificationResult(
    label=label, class_id=0,
    confidence=confidence,
    all_scores=list(scores.values()),
    inference_ms=5.0,
)

def _make_anm(self, error=0.005, threshold=0.015):
    from anomaly_detector import AnomalyResult
    return AnomalyResult(
        reconstruction_error=error,
        is_anomaly=error > threshold,
        threshold=threshold,
        anomaly_map=None,
        inference_ms=3.0,
    )

def setUp(self):
    from decision_engine import DecisionEngine
    self.engine = DecisionEngine()

def test_premium_no_anomaly_is_accept(self):
    d = self.engine.decide(
        self._make_cls("premium_grade", 0.95),
        self._make_anm(0.005),
    )
    self.assertEqual(d.verdict, "ACCEPT")

def test_anomaly_overrides_accept(self):
    d = self.engine.decide(
        self._make_cls("premium_grade", 0.95),
        self._make_anm(0.030), # above threshold
    )
    self.assertEqual(d.verdict, "REJECT")

def test_rejected_class_is_reject(self):
    d = self.engine.decide(
        self._make_cls("rejected", 0.90),
        self._make_anm(0.005), # no anomaly
    )
    self.assertEqual(d.verdict, "REJECT")

def test_standard_grade_is_warning(self):
    d = self.engine.decide(
        self._make_cls("standard_grade", 0.80),
        self._make_anm(0.005),
    )
    self.assertEqual(d.verdict, "WARNING")

def test_composite_score_range(self):
    d = self.engine.decide(
        self._make_cls("premium_grade", 0.95),

```

```

        self._make_anm(0.005),
    )
    self.assertGreaterEqual(d.confidence_score, 0.0)
    self.assertLessEqual(d.confidence_score, 1.0)

def test_str_representation(self):
    d = self.engine.decide(
        self._make_cls("premium_grade", 0.95),
        self._make_anm(0.005),
    )
    self.assertIn("ACCEPT", str(d))

class TestIntegration(unittest.TestCase):
    """End-to-end pipeline test (no camera, synthetic frame)."""

def test_full_pipeline(self):
    from main import FlourQCSystem, SystemConfig
    cfg = SystemConfig(enable_visualizer=False)
    system = FlourQCSystem(cfg)
    frame = make_frame()
    decision = system.process_frame(frame)
    self.assertIn(decision.verdict, ["ACCEPT", "WARNING", "REJECT"])
    self.assertGreaterEqual(decision.confidence_score, 0.0)
    self.assertLessEqual(decision.confidence_score, 1.0)

def test_pipeline_performance(self):
    """Pipeline should process ≥10 FPS on synthetic frames."""
    import time
    from main import FlourQCSystem, SystemConfig
    cfg = SystemConfig(enable_visualizer=False)
    system = FlourQCSystem(cfg)
    frame = make_frame()

    N = 20
    t0 = time.perf_counter()
    for _ in range(N):
        system.process_frame(frame)
    elapsed = time.perf_counter() - t0
    fps = N / elapsed
    print(f"\nPipeline FPS (synthetic): {fps:.1f}")
    self.assertGreater(fps, 5.0, "Pipeline too slow (< 5 FPS on synthetic frames)")

if __name__ == "__main__":
    unittest.main(verbosity=2)

```

## train.py

```

import argparse
import logging
import numpy as np
from pathlib import Path

```

```
logging.basicConfig(level=logging.INFO, format="%asctime)s %(levelname)s %(message)s")
logger = logging.getLogger("FlourQC.Train")
```

```
AE_SIZE = (64, 64)
CLF_SIZE = (224, 224)
CLASSES = ["premium_grade", "standard_grade", "rejected"]
```

```
def load_images(paths, size):
```

```
    import cv2
    imgs = []
    for p in paths:
        img = cv2.imread(str(p))
        if img is None:
            continue
        img = cv2.resize(img, size[::-1])
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        imgs.append(img.astype(np.float32) / 255.0)
    return np.array(imgs)
```

```
def train_autoencoder(data_dir: Path, output_path: str, epochs=30, batch_size=32):
```

```
    import tensorflow as tf
    from anomaly_detector import AutoencoderAnomalyDetector

    image_paths = list(data_dir.glob("**/*.jpg")) + list(data_dir.glob("**/*.png"))
    logger.info("Found %d images for autoencoder training.", len(image_paths))
    if not image_paths:
        raise ValueError(f"No images found in {data_dir}")
```

```
    X = load_images(image_paths, AE_SIZE)
    logger.info("Loaded images shape: %s", X.shape)
```

```
    detector = AutoencoderAnomalyDetector(input_size=AE_SIZE)
    model = detector._model
```

```
    model.compile(
        optimizer=tf.keras.optimizers.Adam(1e-3),
        loss="mse",
    )
```

```
    callbacks = [
        tf.keras.callbacks.EarlyStopping(patience=5, restore_best_weights=True),
        tf.keras.callbacks.ReduceLROnPlateau(patience=3, factor=0.5, min_lr=1e-6),
    ]
```

```
    model.fit(
        X, X, # autoencoder: input == target
        epochs=epochs,
        batch_size=batch_size,
        validation_split=0.1,
        callbacks=callbacks,
        verbose=1,
    )
```

```

model.save(output_path)
logger.info("Autoencoder saved to %s", output_path)

# Compute threshold from validation reconstruction errors
recon = model.predict(X[:100])
errors = ((X[:100] - recon) ** 2).mean(axis=(1, 2, 3))
threshold = float(np.mean(errors) + 2 * np.std(errors))
logger.info("Suggested anomaly threshold: %.5f (mean+2σ over training set)",
threshold)
return threshold

def train_classifier(data_dir: Path, output_path: str, epochs=20, batch_size=16):
    import tensorflow as tf

    # Build dataset using image_dataset_from_directory
    train_ds = tf.keras.utils.image_dataset_from_directory(
        str(data_dir),
        class_names=CLASSES,
        image_size=CLF_SIZE,
        batch_size=batch_size,
        validation_split=0.2,
        subset="training",
        seed=42,
        label_mode="categorical",
    )
    val_ds = tf.keras.utils.image_dataset_from_directory(
        str(data_dir),
        class_names=CLASSES,
        image_size=CLF_SIZE,
        batch_size=batch_size,
        validation_split=0.2,
        subset="validation",
        seed=42,
        label_mode="categorical",
    )

    # Normalise to [0,1]
    normalise = tf.keras.layers.Rescaling(1.0 / 255.0)
    train_ds = train_ds.map(lambda x, y: (normalise(x), y))
    val_ds = val_ds.map(lambda x, y: (normalise(x), y))

    # Data augmentation for robustness to lighting/noise changes
    augment = tf.keras.Sequential([
        tf.keras.layers.RandomFlip("horizontal"),
        tf.keras.layers.RandomRotation(0.1),
        tf.keras.layers.RandomZoom(0.1),
        tf.keras.layers.RandomBrightness(0.2),          # lighting variation
        tf.keras.layers.RandomContrast(0.2),           # contrast variation
        tf.keras.layers.GaussianNoise(0.02),           # sensor noise
    ])
    train_ds = train_ds.map(lambda x, y: (augment(x, training=True), y))

```

```

# Prefetch for performance
AUTOTUNE = tf.data.AUTOTUNE
train_ds = train_ds.prefetch(AUTOTUNE)
val_ds = val_ds.prefetch(AUTOTUNE)

# MobileNetV2 transfer learning
base = tf.keras.applications.MobileNetV2(
    input_shape=(*CLF_SIZE, 3),
    include_top=False,
    weights="imagenet",
)
base.trainable = False

inputs = tf.keras.Input(shape=(*CLF_SIZE, 3))
x = base(inputs, training=False)
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Dropout(0.3)(x)
outputs = tf.keras.layers.Dense(len(CLASSES), activation="softmax")(x)
model = tf.keras.Model(inputs, outputs)

model.compile(
    optimizer=tf.keras.optimizers.Adam(1e-3),
    loss="categorical_crossentropy",
    metrics=["accuracy"],
)

# Phase 1: train head only
logger.info("Phase 1: training classification head.")
model.fit(train_ds, validation_data=val_ds, epochs=epochs // 2, verbose=1)

# Phase 2: unfreeze top layers and fine-tune
logger.info("Phase 2: fine-tuning top MobileNetV2 layers.")
base.trainable = True
for layer in base.layers[:-30]:
    layer.trainable = False

model.compile(
    optimizer=tf.keras.optimizers.Adam(1e-5),
    loss="categorical_crossentropy",
    metrics=["accuracy"],
)
callbacks = [
    tf.keras.callbacks.EarlyStopping(patience=5, restore_best_weights=True),
]
model.fit(
    train_ds, validation_data=val_ds,
    epochs=epochs, initial_epoch=epochs // 2,
    callbacks=callbacks, verbose=1,
)

model.save(output_path)
logger.info("Classifier saved to %s", output_path)

```

```

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument("--mode", choices=["autoencoder", "classifier"], required=True)
    parser.add_argument("--data_dir", required=True)
    parser.add_argument("--output", required=True)
    parser.add_argument("--epochs", type=int, default=30)
    parser.add_argument("--batch_size", type=int, default=16)
    args = parser.parse_args()

    data_dir = Path(args.data_dir)
    if args.mode == "autoencoder":
        train_autoencoder(data_dir, args.output, args.epochs, args.batch_size)
    else:
        train_classifier(data_dir, args.output, args.epochs, args.batch_size)

```

## visualizer.py

```

import cv2
import numpy as np
from typing import Optional
from classifier import ClassificationResult
from anomaly_detector import AnomalyResult
from decision_engine import QualityDecision

# Colour palette (BGR)
COLORS = {
    "ACCEPT": (50, 200, 50),
    "WARNING": (0, 165, 255),
    "REJECT": (0, 0, 220),
    "bg": (20, 20, 20),
    "text": (240, 240, 240),
}
FONT = cv2.FONT_HERSHEY_SIMPLEX

class ResultVisualizer:
    """Draws HUD overlays directly onto a BGR frame (in-place)."""

    def draw(
        self,
        frame: np.ndarray,
        processed: np.ndarray,
        cls: ClassificationResult,
        anm: AnomalyResult,
        decision: QualityDecision,
    ):
        self._draw_verdict_banner(frame, decision)
        self._draw_info_panel(frame, cls, anm, decision)
        if anm.anomaly_map is not None:
            self._draw_anomaly_overlay(frame, anm.anomaly_map)

    def _draw_verdict_banner(self, frame: np.ndarray, decision: QualityDecision):
        h, w = frame.shape[:2]

```

```

color = COLORS.get(decision.verdict, (128, 128, 128))
# Top bar
cv2.rectangle(frame, (0, 0), (w, 36), color, -1)
cv2.putText(frame, decision.verdict, (10, 26),
            FONT, 0.9, (255, 255, 255), 2, cv2.LINE_AA)
score_txt = f"Score: {decision.confidence_score:.2f}"
cv2.putText(frame, score_txt, (w - 160, 26),
            FONT, 0.7, (255, 255, 255), 1, cv2.LINE_AA)

def _draw_info_panel(
    self,
    frame: np.ndarray,
    cls: ClassificationResult,
    anm: AnomalyResult,
    decision: QualityDecision,
):
    h, w = frame.shape[:2]
    panel_w = 230
    panel_h = 110
    x0, y0 = 10, 46
    overlay = frame.copy()
    cv2.rectangle(overlay, (x0, y0), (x0 + panel_w, y0 + panel_h), COLORS["bg"], -1)
    cv2.addWeighted(overlay, 0.6, frame, 0.4, 0, frame)

    lines = [
        f"Class : {cls.label}",
        f"Conf  : {cls.confidence:.1%}",
        f"AE err: {anm.reconstruction_error:.4f}",
        f"Thresh: {anm.threshold:.4f}",
    ]
    for i, txt in enumerate(lines):
        cv2.putText(frame, txt, (x0 + 6, y0 + 18 + i * 22),
                    FONT, 0.48, COLORS["text"], 1, cv2.LINE_AA)

def _draw_anomaly_overlay(self, frame: np.ndarray, anomaly_map: np.ndarray):
    """Blend a heat-map of reconstruction errors onto the frame."""
    h, w = frame.shape[:2]
    # Normalise map 0-255
    amap = anomaly_map[:, :, 0] if anomaly_map.ndim == 3 else anomaly_map
    amap_resized = cv2.resize(amap, (w, h))
    amap_norm = cv2.normalize(amap_resized, None, 0, 255,
cv2.NORM_MINMAX).astype(np.uint8)
    heat = cv2.applyColorMap(amap_norm, cv2.COLORMAP_JET)
    blended = cv2.addWeighted(frame, 0.75, heat, 0.25, 0)
    frame[:] = blended

class MetricsLogger:
    """Accumulates per-frame decisions and prints a summary."""

    def __init__(self):
        self._counts = {"ACCEPT": 0, "WARNING": 0, "REJECT": 0}
        self._total = 0

```

```
def log(self, decision: QualityDecision):
    self._counts[decision.verdict] = self._counts.get(decision.verdict, 0) + 1
    self._total += 1

def summary(self) -> str:
    if self._total == 0:
        return "No frames processed."
    lines = [f"Total frames: {self._total}"]
    for v, c in self._counts.items():
        lines.append(f" {v}: {c} ({c/self._total:.1%}")
    return "\n".join(lines)
```

### Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Артём Гуцалюк

**Співавтор:**

**Назва:** Кіберфізична система збору, аналізу якості борошна на підставі машинного навчання

**Експерт:** Євген Федоров

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:** 7.25%

**Коефіцієнт подібності 2:** 3%

**Мікропробіли:** 3

**Заміна букв:** 2

**Інтервали:** 0

**Білі знаки:** 6

**Дата створення звіту:** 2026-04-22 12:18:34.0

**Після аналізу Звіту подібності констатую наступне:**

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2026-04-22

Дата



Доцент Андрій Нічепорук

експерт

27.04.26, 21:30

Гуцалюк.html

Wed Apr 22 11:44:42 EEST 2026, Медзятий Дмитро Миколайович, Хмельницький національний університет, ХНУ

## Anti-Plagiarism (<http://ap.km.ua>) v-15.701

Максимальне співпадіння з одним документом **17.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: **10%**

ID: 270612 Назва: МКР Кіберфізична система збору, аналізу якості борошна на підставі машинного навчання Додано в БД: 2026-04-22 Автора: Артем Гуцалюк Керівники: Євген Федоров Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	120565	1015	23049 (19%)	187 (18%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
269790	Назва: Звіт з ПДП Кіберфізична система збору, аналізу якості борошна на підставі машинного навчання Додано в БД: 2026-03-12 Автора: Гуцалюка А.О. Керівники: Федоров Є.Є Консультанти: Опоненти:	20332 (17.0%)	147 (14.0%)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Здобувач: Артем Гуцалюк

Тема: Кіберфізична система збору, аналізу якості борошна на підставі машинного навчання

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи магістра:

Кількість листів креслень —; кількість сторінок записки 81

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано систему збору, аналізу якості борошна на підставі машинного навчання

2. Висновок про відповідність роботи дипломному завданню Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз сучасного стану автоматизації контролю якості у борошномельній промисловості. Досліджено існуючі підходи та обґрунтовано доцільність використання кіберфізичних систем. У другому розділі запропоновано математичні та концептуальні моделі функціонування системи контролю якості в режимі реального часу. У третьому розділі розроблено метод комплексного оцінювання якості борошна на основі машинного навчання. У четвертому розділі програмно-апаратно реалізовано кіберфізичну систему моніторингу та експериментально досліджено її ефективність.

4. Позитивні сторони роботи: Запропонована система комплексного оцінювання якості борошна дозволяє здійснювати безперервний автоматизований моніторинг продукції в режимі реального часу, а також виявляти відхилення сортності та наявність сторонніх домішок, суттєво мінімізуючи вплив людського фактора на виробництві.

5. Негативні сторони роботи: В роботі недостатньо глибоко проаналізовано вплив надвисокої концентрації сторонніх дефектів на загальну затримку інференсу моделей та пропускну здатність системи.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: -

9. Оцінка кваліфікаційної роботи магістра: Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи магістра вважаю, що робота заслуговує оцінки «Добре» 85.00 (B)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_  
д.т.н., професор Манзюк Е.А., Професор кафедри комп'ютерних наук

“1 Травня” \_\_\_\_\_ 2026р.



Зав. кафедри КІС  
д-р. філософії Ользі ПАВЛОВІЙ

Артем ГУЦАЛЮК

---

ПІВ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2М-24-1

### ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

1 травня 2026 року



## РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Кіберфізична система збору, аналізу якості борошна на підставі машинного навчання

Автор Артем ГУЦАЛЮК

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти другий (магістерський)

Спеціальність 123 Комп'ютерна інженерія

Науковий керівник: к.т.н., доцент Євген ФЕДОРОВ

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

## Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.
- 4) значна частина знайденого плагіату відноситься до списку використаних джерел

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 7,25% і адресується до 49 першоджерела; та системою Anti-Plagiarism складає 17%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

25.04.2026

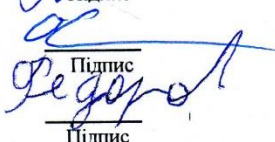
Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи

  
Підпис

Ольга ПАВЛОВА  
Ім'я, ПРІЗВИЩЕ

  
Підпис

Олег САВЕНКО  
Ім'я, ПРІЗВИЩЕ

Євген ФЕДОРОВ  
Ім'я, ПРІЗВИЩЕ