

Міністерство освіти і науки України  
Харківський національний університет імені В. Н. Каразіна  
Навчально-науковий інститут комп'ютерних наук та штучного інтелекту  
Кафедра комп'ютерних систем та робототехніки

«Затверджую»  
в.о. завідуючого кафедри  
комп'ютерних систем та робототехніки  
\_\_\_\_\_ к. ф.-м. н., доцент Максим ХРУСЛОВ  
«\_\_\_» червня 2025 р.



## Пояснювальна записка


до кваліфікаційної роботи  
бакалавра

на тему: «МОДЕЛЬ ВИЯВЛЕННЯ АНОМАЛІЙ НА ВИРОБНИЧИХ  
ЛІНІЯХ ЗА ДОПОМОГОЮ СИСТЕМ КОМП'ЮТЕРНОГО ЗОРУ»

Спеціальність 123 – Комп'ютерна інженерія  
Галузь знань 12 – Інформаційні технології  
Освітня програма «Комп'ютерна інженерія»

Захищено на засіданні  
Екзаменаційної комісії № 44  
протокол № \_\_ від \_\_.06.2025 р.  
Оцінка \_\_\_\_\_ / \_\_\_\_\_  
Голова Екзаменаційної комісії  
\_\_\_\_\_ ЧУГАЙ А. М.

Виконав:  
Студент групи КІ– 41  
БУШТЕЦЬ Кирило Віталійович   
Керівник: к.т.н., доцент кафедри  
комп'ютерних систем та робототехніки  
СТРІЛЕЦЬ Вікторія Євгенівна 

Рецензент: д.т.н., професор, професор  
кафедри математичного моделювання та  
аналізу даних  
КІРІЧЕНКО Людмила Олегівна 

Харків – 2025

## АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи містить в собі вступ, 3 розділи основної частини, висновок, список використаних джерел та додаток. Загальний обсяг роботи складає 75 сторінок з них основного тексту 50 сторінок, 22 рисунка, 1 таблиця, 16 найменувань використаних джерел, переліку умовних позначень та 4 додатки.

**Мета кваліфікаційної роботи** полягає у підвищенні ефективності виявлення дефектів продукції на виробничих лініях за допомогою систем комп'ютерного зору.

**Об'єктом дослідження** є процес виявлення аномалій і дефектів продукції на виробничих лініях.

**Предмет дослідження** – моделі і методи обробки зображень, алгоритми розпізнавання дефектів продукції на виробничих лініях.

Проблема, яку вирішує кваліфікаційна робота пов'язана з підвищенням вимог до сучасних виробничих ліній, що полягає у виявленні дефектів на виробництві з високою точністю та зменшення впливу людського фактору при перевірці готової продукції. Кваліфікаційна робота, присвячена розробці моделі комп'ютерного зору для виявлення аномалій на виробничих лініях, спрямована на вирішення цих проблем. У роботі розглядається можливість інтеграції в виробничі процеси системи комп'ютерного зору для моніторингу аномалій та виявлення дефектів для автоматизації контролю якості.

Область застосування – промислове виробництво, харчова промисловість, фармацевтика, електроніка, автомобілебудування. Розроблена модель може бути використана для підвищення якості продукції, зниження кількості дефектів та оптимізації виробництва.

*Ключові слова:* НЕЙРОННІ МЕРЕЖІ, МАШИННЕ НАВЧАННЯ, КОМП'ЮТЕРНИЙ ЗІР, ВИЯВЛЕННЯ АНОМАЛІЙ, ОБРОБКА ЗОБРАЖЕНЬ, АВТОМАТИЗАЦІЯ, ВИРОБНИЧА ЛІНІЯ.

## ABSTRACT

The explanatory note to the qualification work contains an introduction, 3 chapters of the main part, conclusion, list of references, and an appendix. The total volume of the work is 75 pages, of which the main text contains 50 pages, 22 figures, 1 tables, 16 reference sources, a list of conventional designations, and 4 appendices.

**The purpose of the qualification work** is to improve the efficiency of product defect detection on production lines using computer vision systems.

**The object of research** is the process of detecting anomalies and product defects on production lines.

**The subject of research** is models and methods of image processing, algorithms for recognizing product defects on production lines.

The problem that the qualification work solves is related to the increasing requirements for modern production lines, which consists in detecting defects in production with high accuracy and reducing the influence of the human factor when checking finished products. The qualification work, dedicated to the development of a computer vision model for detecting anomalies on production lines, aims to solve these problems. The paper examines the possibility of integrating computer vision systems into production processes for monitoring anomalies and detecting defects to automate quality control.

Area of application – industrial manufacturing, food industry, pharmaceuticals, electronics, automotive. The developed model can be used to improve product quality, reduce the number of defects, and optimize production.

*Keywords:* NEURAL NETWORKS, MACHINE LEARNING, COMPUTER VISION, ANOMALY DETECTION, IMAGE PROCESSING, AUTOMATION, PRODUCTION LINE.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП .....	7
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ІСНУЮЧИХ СИСТЕМ КОМП'ЮТЕРНОГО ЗОРУ .....	9
1.1 Поняття та принципи роботи систем комп'ютерного зору .....	9
1.2 Огляд сучасних систем комп'ютерного зору для виявлення аномалій .....	12
1.3 Алгоритми та моделі для виявлення дефектів на виробничих лініях.....	15
1.4 Аналіз ефективності існуючих рішень та їх обмеження .....	18
Висновки за розділом 1 .....	23
РОЗДІЛ 2. РОЗРОБКА МОДЕЛІ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ НА ЗОБРАЖЕННЯХ .....	24
2.1 Аналіз та вибір методів для розробки моделі .....	24
2.2 Архітектура системи комп'ютерного зору .....	26
2.3 Вибір програмних методів реалізації.....	28
2.4 Вибір методу навчання моделі виявлення аномалій .....	30
2.5 Проектування моделі .....	32
2.6 Навчальний набір даних .....	35
2.7 Алгоритми підготовки та обробки зображень .....	35
2.8 Стратегія валідації моделі.....	36
Висновки за розділом 2 .....	37
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА АНАЛІЗ ЕКСПЕРИМЕНТАЛЬНИХ РЕЗУЛЬТАТІВ .....	38
3.1 Програмна реалізація моделі виявлення аномалій .....	38
3.2 Навчання та валідація моделі .....	39
3.3 Інтерфейс користувача системи виявлення аномалій .....	44
3.4 Проведення та аналіз експериментальних досліджень .....	47

3.4.1 Експериментальні дослідження на датасеті «carpet» .....	48
3.4.2 Експериментальні дослідження на датасеті «screw» .....	50
Висновки за розділом 3 .....	52
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
ДОДАТКИ.....	58

**ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ**

API	– Application Programming Interface;
CNN	– Convolutional Neural Network;
CPU	– Central Processing Unit;
GPU	– Graphics Processing Unit;
HSV	– Hue, Saturation, Value;
MSE	– Mean Squared Error;
MVTec AD	– MVTec Anomaly Detection;
ReLU	– Rectified Linear Unit;
SVM	– Support Vector Machine;
UI	– User Interface;
VPU	– Vision Processing Unit.

## ВСТУП

**Актуальність роботи.** У сучасному світі інтенсивність виробництва постійно зростає і при цьому надзвичайно важливим є контроль якості продукції, що і стає ключовим фактором для підприємств. Підходи виявлення дефектів за допомогою візуального огляду працівниками є обмеженими та схильні до залежності від зовнішніх і суб'єктивних чинників, що впливає на ефективність оцінки якості продукції на виробничих лініях. Ці фактори особливо помітні при масовому виробництві, де створюється великий обсяг продукції в короткі терміни. Процес контролю якості може бути значно поліпшений через впровадження систем комп'ютерного зору для автоматизації процесу виявлення аномалій на виробничих лініях. Такі системи здатні працювати безперервно, забезпечуючи стабільну точність виявлення дефектів незалежно від зовнішніх факторів. Що робить їх інтеграцію перспективним напрямком і може значно підвищити ефективність виробництва. Актуальність посилюється розвитком технологій штучного інтелекту та машинного навчання, які дозволяють вдосконалювати алгоритми розпізнавання аномалій та дефектів.

**Метою дослідження** є підвищення ефективності виявлення дефектів продукції на виробничих лініях за допомогою систем комп'ютерного зору, що забезпечить точну ідентифікацію аномалій у виробничому процесі.

**Об'єктом дослідження** є процес виявлення аномалії і дефектів продукції на виробничих лініях.

**Методи дослідження:** аналіз літератури та наукових джерел, порівняльний аналіз, моделювання, навчання нейронної мережі, тестування моделі.

**Предмет дослідження** – моделі і методи обробки зображень, алгоритми розпізнавання дефектів продукції на виробничих лініях.

Для досягнення сформульованої мети були виділені такі **завдання дослідження**:

- виконати аналіз існуючих методів та моделей систем комп'ютерного зору, які використовуються для аналізу дефектів виробів на сьогодні;
- розробити модель виявлення аномалій на основі нейронних мереж;
- створити програмну реалізацію запропонованої моделі та провести експериментальне дослідження;
- розробити опис рекомендацій щодо застосування програмної реалізації моделі комп'ютерного зору для виявлення дефектів і аномалій виробничої продукції.

# РОЗДІЛ 1

## ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ІСНУЮЧИХ СИСТЕМ КОМП'ЮТЕРНОГО ЗОРУ

### 1.1 Поняття та принципи роботи систем комп'ютерного зору

Комп'ютерний зір – це технологія штучного інтелекту, яка використовуючи нейронні мережі та машинне навчання прагне імітувати людську зорову систему. Завдання комп'ютерного зору охоплює методи отримання, обробки, аналізу зображень для отримання потрібної інформації.

На відміну від традиційної обробки зображень, яка спрямована на зміну візуальних даних, таких як яскравість, контрастність, насиченість, різкість та інші, комп'ютерний зір спрямований на інтелектуальне розуміння вмісту зображень – зокрема, на ідентифікацію, класифікацію або сегментацію об'єктів, відстеження руху, реконструкцію тощо. Це робить технологію комп'ютерного зору невід'ємною частиною у багатьох сучасних галузях. Вона використовується в промисловій автоматизації, медичній діагностиці, системах відеоспостереження та системах контролю якості на виробництві [16].

Архітектури сучасних систем комп'ютерного зору містять такі основні складові як:

- пристрої захоплення та введення зображень – камери видимого спектру, інфрачервоні камери, спеціалізовані датчики та тепловізійні сенсори;
- системи освітлення – світлодіоди та лампи;
- апаратні засоби – центральні процесори (CPU), графічні процесори (GPU) з підтримкою CUDA [8] та ROCm [1] для GPU AMD, спеціалізовані процесори комп'ютерного зору (VPU), периферійні обчислювальні пристрої;

- програмні засоби – бібліотеки комп'ютерного зору, фреймворки машинного навчання, алгоритми обробки зображень та даних;
- датасети – підготовлені набори даних, які за потреби містять дані разом з відповідною розміткою (анотаціями).

Функціонування систем комп'ютерного зору дозволяють комп'ютерам «бачити» за принципом імітації зорового сприйняття людини та інтерпретувати інформацію зображень або відео. Вони базуються на послідовній обробці візуальної інформації за кількома ключовими етапами.

1. Отримання даних – початковий етап, на якому система захоплює дані за допомогою відповідних пристроїв. Якість даних залежить від зовнішніх умов (освітлення, погодні умови та інше), формату представлення даних, характеристик та роздільної здатності пристроїв.

2. Попередня обробка. Цей етап спрямований на покращення якості даних та їх підготовка для аналізу. Основні методи включають корекцію освітлення та кольору, фільтрацію шумів, трансформація та бінаризація зображень.

3. Сегментація. На цьому етапі проходить процес розділення зображення на значущі області та об'єкти для спрощення подальшого аналізу. Алгоритми сегментації поділяються на порогові методи, методи кластеризації, методи виявлення країв або контурів, сегментація на основі глибоких нейронних мереж, та виявлення ключових точок чи патернів.

4. Виділення ознак. Етап ідентифікації та виділення характеристик, які будуть використовуватися для класифікації або розпізнавання, включаючи геометричні, текстурні, колірні та інші ознаки.

5. Визначення та класифікація. На етапі розпізнавання система приймає рішення щодо ідентифікації класу об'єкту на зображенні на основі виділених ознак. У системах виявлення аномалій цей процес може базуватися як на класифікації, так і на реконструкції нормального стану з виявлення відхилень.

б. Постобробка та прийняття рішень. Фінальний етап роботи системи комп'ютерного зору на якому передбачається інтерпретація результатів та відображення результату. Це може бути визначення об'єкту та оцінку його характеристик, класифікація об'єкту, локалізацію об'єктів чи генерація сигналів для подальших дії.

Для комп'ютерного зору математичні основи є фундаментом для розробки та роботи систем. Ці основи охоплюють математичні дисципліни, такі як лінійна алгебра, статистика, оптимізація та геометрія, які забезпечують теоретичну базу, підходи і методи машинного навчання, які формують інструменти для обробки та аналізу даних та подальшу їх інтерпретацію.

Цифрове зображення у машинному навчанні представляє собою набір числових даних у вигляді матриць. Кожен елемент матриці – це піксель, що містить числову інформацію про свої характеристики у точці зображення. Кольорові зображення зазвичай представлені у вигляді набору з трьох масивів чисел, у різних колірних просторах ці масиви можуть відповідати різним параметрам (рис. 1.1).

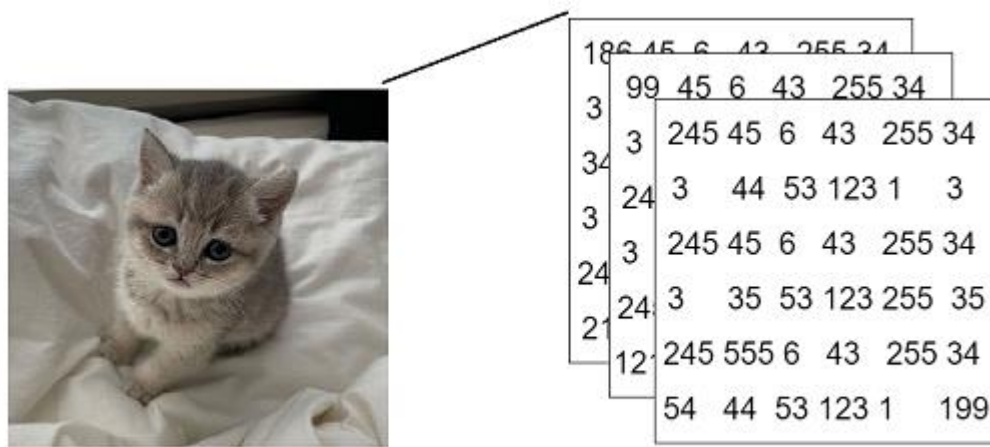


Рисунок 1.1 – Представлення зображення у машинному навчанні

Для RGB кожна матриця відповідає інтенсивності червоного, зеленого та синього кольору. Для HSV – відтінок, насиченість та яскравість. Для LAB – світлість, зелено-червона градація та синьо-жовта градація. Для YCbCr – яскравість, різниця блакитного кольору та різниця червоного кольору. Тому зображення можна розглядати як тривимірну матрицю розміром  $H \times W \times 3$ , де кожен елемент є числовим значення інтенсивності відповідного каналу. Дана дискретна модель дає змогу виконувати математичні операції над зображенням, зокрема сегментацію, трансформацію, бінаризацію та інші методи обробки. На відміну від кольорових, чорно-біле зображення є одноканальним зображенням, у якому кожен елемент містить інформацію про градацію сірого.

Одним із компонентів комп'ютерного зору є згорткові нейронні мережі, які є галуззю машинного навчання і також використовують математичні обчислення як основу для своїх операцій згортки, активаційних функцій, функцій помилки та інших.

## **1.2 Огляд сучасних систем комп'ютерного зору для виявлення аномалій**

Методи глибокого навчання та згорткові нейронні мережі (CNN) революційно вплинули на галузь комп'ютерного зору [4, 7, 13]. Завдяки їх високій здатності до автоматичного вилучення характеристик і навчання на великих обсягах даних, ці підходи значно підвищили надійність аналізу зображень, зокрема у системах комп'ютерного зору для виявлення аномалій.

У традиційних системах використовувалися базові методи обробки зображень – фільтрація, порогова сегментація та інші евристичні підходи. Ці системи вимагали ретельного ручного налаштування для коректної роботи та були дуже чутливі до змін характеристик даних (освітлення, фону, форми чи положення об'єкту на зображенні). Сучасні системи побудовані на основі глибоких нейронних мереж, завдяки чому вони мають здатність

самостійно вивчати представлення ознак на заданих датасетах, що значно знижує залежність їх ефективності від людського втручання та зовнішніх факторів та покращує якість розпізнавання.

Сучасні архітектури моделей:

- класифікаційні мережі – VGG, ResNet, EfficientNet;
- рекурентні мережі – CNN-RNN;
- трансформери комп'ютерного зору – ViT, Swin Transformer, DETR;
- сегментаційні мережі – U-Net, Mask R-CNN, DeepLabv3+;
- мережі для виявлення об'єктів - Faster R-CNN, YOLO, SSD;
- генеративні моделі – AE, VAE, GAN, One-Class SVM;
- гібридні архітектури – EfficientNet, ConvNeXt.

Системи комп'ютерного зору для виявлення аномалій розвиваються дуже активно через широке застосування в промислових, медичних, безпекових та транспортних галузях. Основна мета систем – це виявлення відхилень від норм на зображеннях чи відеопотоці [1]. Ці дані можна передавати як в реальному часі, так і у форматі архівних даних.

*Автоенкодер* (Autoencoder) стали одним із найпопулярніших підходів до розробки системи виявлення аномалій за допомогою комп'ютерного зору [2]. Ці нейромережі навчаються відтворювати зображення без дефектів, стискаючи їх у латентний простір з подальшою реконструкцією. Серед автоенкодерів існують кілька версій.

Класичний автоенкодер (Vanilla Autoencoder) – базова модель, що складається з енкодера, який стискає дані у латентний простір, та декодера, що відтворює дані з латентного простору.

Варіаційні автоенкодер (Variational Autoencoder) – це вдосконалена різновидність автоенкодерів, до яких додано кодування вхідних даних не у одну точку в латентному просторі, а у ймовірнісний нормальний розподіл в латентному просторі.

Згорткові автоенкодери (Convolutional Autoencoder) – це різновид автоенкодера, який використовує згорткові та деконволюційні шари для обробки даних.

Контрактивний автоенкодер (Contractive Autoencoder) – автоенкодер який додає штраф через чутливість енкодеру до малих змін у вхідних даних, це дозволяє латентному простору бути більш стійким до змін.

Стекований автоенкодер (Stacked Autoencoder) – складається з кількох шарів автоенкодерів, де кожен під'єднаний до наступного.

**Генеративні змагальні мережі** (Generative adversarial networks) – це клас моделей глибокого навчання, які спроможні до генерації нових даних виходячи з навчального набору, шляхом створення змагального процесу між двома нейронними мережами: генератором і дискримінатором. Ці мережі активно використовуються в задачах виявлення аномалій, серед них також є модифікації.

AnoGAN (Anomaly Detection with Generative Adversarial Networks) – підхід, що одним із перших використовувався в генеративно змагальних мережах. Використовує глибокі згорткові генеративні змагальні мережі для навчання на даних без дефектів без вчителя.

GANomaly – напівконтрольований підхід для виявлення аномалій шляхом поєднання автоенкодерів та генеративно змагальних мереж. Використовує два енкодери та декодер у генераторі.

На основі цих моделей побудовано багато систем для контролю якості продукції. Приклади сучасних систем комп'ютерного зору для виявлення аномалій ілюструють практичне застосування методів виявлення дефектів:

- Amazon Lookout for Vision. Хмарна платформа, основною задачею якої автоматизація виявлення візуальних дефектів на виробничих лініях. Використовує моделі глибокого навчання для аналізу зображень. Не потребує глибокого розуміння машинного навчання від користувача.

- Microsoft Azure Anomaly Detector. Програмний інтерфейс програми (API) для виявлення аномалій, підтримує аналіз зображень на

попередньо навчених моделях глибокого навчання. Легко інтегрується в додатки та використовується в медичних закладах та промислових IoT-системах.

– Landing Visual AI. Хмарна платформа для розробки та впровадження моделей комп'ютерного зору у виробництво. Особливістю є доступність для підприємств різних галузей та не потребує знань у програмуванні чи машинному навчанні.

– MVtec HALCON. Комерційна програмна платформа для комп'ютерного зору. Використовується в промисловій автоматизації, контролі якості, робототехніці та медичних галузях.

Ці системи демонструють різноманітність підходів у вирішенні задач виявлення відхилень.

### **1.3 Алгоритми та моделі для виявлення дефектів на виробничих лініях**

У сучасних системах комп'ютерного зору для виявлення аномалій та дефектів використовуються багато різноманітних алгоритмів, які можна класифікувати за підходом до аналізу зображень та алгоритмами виявлення відхилень від норми. Ці методи включають традиційні техніки обробки зображень, класичні алгоритми машинного навчання, а також сучасні моделі глибокого навчання, які забезпечують найвищу точність і адаптивність.

*Традиційні методи* обробки зображень базуються на аналізі візуальних характеристик зображень і не потребують навчання. Вони ефективні для простих сценаріїв із чіткими дефектами та однорідними фонами, але менш стійкі до зовнішніх умов.

– порогові методи та сегментація – один з найпростіших ефективних способів виявлення дефектів на однорідних поверхнях. Принцип роботи алгоритму полягає у встановленні граничного значення інтенсивності пікселів, на основі якого зображення поділяється на об'єкт та фон.

Сегментація зображень дозволяє розділити зображення на області, що спрощують подальший аналіз. Методи сегментації засновані на кластеризації. Використовуються для виділення потенційних дефектів.

– методи на основі контурів та морфологічної обробки забезпечують виявлення контурів за допомогою операторів Собеля, Кенні, Прюїтт і визначають різкі зміни на зображенні, що часто відповідають краям дефектів. Після знаходження контурів застосовуються методи морфологічної обробки для усунення шумів та покращення даних.

– методи статистичного аналізу та фільтрації. Статистичні методи, такі як аналіз гістограм, обчислення моментів зображення та статистик текстур, використовуються для характеристикації нормальних зразків та виявлення відхилень. Фільтри Гаусса, медіанні фільтри та фільтри Вінера застосовуються для попередньої обробки зображень та зниження рівня шуму.

– методи на основі шаблонів порівнюють зображення з еталонним шаблоном для виявлення відхилень. Використовуються нормалізовані методи кореляції для стійкості до змін освітлення.

З розвитком технологій та зростанням обчислювальних потужностей, **методи машинного навчання** також стали одним із найпотужніших інструментів для виявлення дефектів на виробничих лініях. На відміну від традиційних методів, алгоритми машинного навчання здатні автоматично виявляти складні патерни та адаптуватися до різноманітних типів дефектів без необхідності розробки правил для кожного випадку. Машинне навчання дозволяє системам комп'ютерного зору навчатися на прикладах, постійно вдосконалюючи точність виявлення аномалій. Виділяють такі класи методів машинного навчання:

– методи навчання з вчителем (контрольоване) передбачають навчання на наборі даних, де кожен зразок має відповідну мітку, модель навчається передбачати ці мітки для нових даних, мінімізуючи помилку між передбаченими та справжніми мітками. Використовуються для

розпізнавання об'єктів, класифікації та діагностики зображень або системах розпізнавання облич;

– методи навчання без вчителя (неконтрольоване) передбачають навчання на наборі, коли дані не мають міток, тому модель повинна самостійно знаходити залежності, структури чи розподіли в даних. Ці методи часто застосовуються для зменшення розмірності, кластеризації, генерації даних або виявлення аномалій;

– методи навчання з підкріплення передбачають навчання шляхом проб та помилок, отримуючи винагороду або штраф за результат. Використовується для задач, де агент повинен навчитися приймати рішення;

– напівконтрольоване навчання поєднує невелику кількість міток із неанотованими даними. Наприклад, псевдомітки, згенеровані моделями, використовуються для подальшого навчання.

Разом з впровадженням *методів глибокого навчання* відбувся революційний прорив у галузі виявлення дефектів, вони принципово змінили підходи до вирішення задач комп'ютерного зору на виробництві [15, 14]. Глибокі нейронні мережі з їхньою здатністю автоматично вилучати ієрархічні ознаки з зображень дозволили досягти безпрецедентної точності у виявленні навіть найскладніших та найменш помітних дефектів. На відміну від традиційних алгоритмів та класичних методів машинного навчання, глибокі моделі не потребують ручного проектування ознак і можуть ефективно працювати з сирими даними, адаптуючись до специфіки конкретного виробничого процесу. Особливо цінною характеристикою є здатність цих систем виявляти нові типи дефектів та функціонувати в складних умовах промислового середовища. Моделі, що використовують методи глибокого навчання поділяються на:

– згорткові нейронні мережі, які стали найпопулярнішими архітектурами для аналізу зображень у промисловості через свою здатність ефективно обробляти зображення та вилучати ієрархічні ознаки. CNN

імітують роботу зорової системи людини, використовуючи шари для вилучення ознак із зображень. Серед них згорткові шари, пулінг-шари, повнозв'язні шари, активаційні функції, нормалізація;

- класифікаційні мережі (ResNet, EfficientNet, VGG) призначені для визначення категорії зображення, тобто вирішення задачі, чи є дефект на зображенні, і, якщо так, до якого класу він належить. Вони приймають зображення як вхід і видають ймовірності належності до певних класів;

- мережі для семантичної сегментації (U-Net, SegNet, DeepLabV3+) забезпечують піксельну класифікацію, тобто визначають клас кожного пікселя на зображенні (наприклад, "дефект" чи "фонове зображення"). Це дозволяє точно локалізувати дефекти, виділяючи їхні межі та форму;

- мережі для детекції об'єктів (Faster R-CNN, YOLO, SSD) визначають позиції об'єктів (у вигляді прямокутних рамок, bounding boxes) і їхні класи на зображенні. У контексті виявлення дефектів вони ідентифікують місця розташування дефектів і класифікують їх (наприклад, "тріщина", "подряпина");

- генеративно-змагальні мережі (GAN) навчаються генерувати зображення нормальних зразків. При появі дефекту генерована модель не здатна точно відтворити аномальну область, що використовується для виявлення дефектів шляхом виявлення відхилення від вхідних даних.

Гібридні методи поєднують традиційні техніки, машинне навчання та глибоке навчання для підвищення ефективності. Наприклад, порогові методи використовуються для попередньої сегментації, після чого CNN класифікує регіони. Або GAN генерує псевдомітки для напівконтрольованого навчання SVM.

#### **1.4 Аналіз ефективності існуючих рішень та їх обмеження**

Сучасні системи комп'ютерного зору для виявлення аномалій і дефектів на виробничих лініях, пропонують потужні інструменти для автоматизації контролю якості. Вони використовують комбінацію

традиційних методів обробки зображень, класичного машинного навчання та глибокого навчання, забезпечуючи високу точність і адаптивність. Однак кожна система має свої сильні сторони, обмеження та специфічні сценарії застосування. Аналіз ефективності систем проводився за ключовими критеріями та розглядалися їхні обмеження, які впливають на впровадження в промислових умовах.

Критерії оцінки ефективності:

- точність – здатність системи правильно ідентифікувати дефекти та мінімізувати хибнопозитивні/хибнонегативні результати;
- швидкість – час обробки зображень, що критично для реального часу на конвеєрних лініях;
- масштабованість – можливість обробки великих обсягів даних і адаптації до різних виробничих ліній;
- доступність – простота використання для користувачів без глибоких знань у машинному навчанні;
- адаптивність – здатність виявляти нові типи дефектів і працювати в змінних умовах (освітлення, текстури, типи поверхонь).

Були розглянуті існуючі рішення, такі як Amazon Lookout for Vision, Microsoft Azure Anomaly Detector, Landing Visual AI та MVTec HALCON (таблиця 1.1).

Таблиця 1.1

## Порівняння рішень комп'ютерного зору для виявлення дефектів за критеріями ефективності

	Amazon Lookout for Vision	Microsoft Azure Anomaly Detector	Landing Visual AI	MVTec HALCON
Точність	Досягає високої точності за умови якісного набору даних для навчання. Ефективна для виявлення подряпин, вм'ятин, дефектів текстури.	Забезпечує хорошу точність завдяки попередньо навченим моделям, але результати залежать від якості даних і попередньої обробки	Досягає конкурентної точності за умови правильного налаштування моделей. Ефективна для класифікації та сегментації дефектів	Забезпечує високу точність завдяки гібридному підходу, який поєднує традиційні методи і глибоке навчання
Швидкість	Швидка обробка в хмарі, але затримки можуть виникати через мережеву передачу даних, що обмежує використання в реальному часі без локального розгортання	Висока швидкість обробки в хмарі, але затримки через API-запити можуть обмежувати реальний час	Обробка залежить від хмарної інфраструктури, що може викликати затримки при високих навантаженнях	Оптимізована для реального часу, особливо при локальному розгортанні, що робить її ідеальною для конвеєрних ліній
Масштабованість	Хмарна інфраструктура Amazon Web Services забезпечує високу масштабованість, дозволяючи обробляти тисячі зображень на годину	Хмарна платформа Azure дозволяє масштабувати обробку для великих виробничих систем	Хмарна платформа забезпечує масштабованість, але залежить від інтеграції з виробничими системами	Підтримує інтеграцію з промисловими системами, але масштабованість залежить від апаратного забезпечення

Продовження табл. 1.1

	Amazon Lookout for Vision	Microsoft Azure Anomaly Detector	Landing Visual AI	MVTec HALCON
Доступність	Інтуїтивний інтерфейс і готові інструменти для навчання роблять платформу доступною для інженерів без спеціалізації в машинному навчанні	API простий у використанні, а інтеграція через SDK робить його доступним для розробників без глибоких знань у комп'ютерному зорі	Графічний інтерфейс і автоматизовані інструменти роблять платформу доступною для широкого кола користувачів, включаючи малий бізнес	Вимагає певних знань у комп'ютерному зорі для налаштування, що може бути викликом для новачків
Адаптивність	Добре адаптується до нових типів дефектів за умови перенавчання, але потребує достатньої кількості анотованих даних	Попередньо навчені моделі добре працюють для типових аномалій, але перенавчання для специфічних дефектів потребує додаткових зусиль і даних	Дозволяє створювати кастомні моделі для нових типів дефектів, але потребує часу на навчання та налаштування	Гнучка завдяки бібліотеці алгоритмів і підтримці кастомних моделей, але адаптація до нових дефектів потребує додаткового програмування

У той же час дані рішення мають і низку обмежень.

- Amazon Lookout for Vision:
  - залежність від хмарної інфраструктури може створювати проблеми в умовах обмеженого інтернет-з'єднання або вимог до безпеки даних;
  - висока вартість для малих і середніх підприємств, особливо при обробці великих обсягів даних;
  - обмежена гнучкість у налаштуванні архітектури моделей, оскільки платформа пропонує попередньо навчені рішення;
  - потреба в якісних анотованих даних для навчання, що може бути трудомістким для нових виробничих ліній.
- Microsoft Azure Anomaly Detector:
  - обмежена підтримка специфічних промислових дефектів, оскільки моделі орієнтовані на загальні аномалії;
  - залежність від хмарного підключення, що може бути проблематичним у віддалених виробництвах;
  - високі витрати на API-запити при великих обсягах даних;
  - менша гнучкість порівняно з кастомними моделями, оскільки API обмежує доступ до внутрішньої архітектури.
- Landing Visual AI:
  - обмежена підтримка специфічних промислових дефектів, оскільки моделі орієнтовані на загальні аномалії;
  - залежність від хмарного підключення, що може бути проблематичним у віддалених виробництвах;
  - затримки в обробці через необхідність передачі даних до хмарних серверів та отримання результатів;
  - менша гнучкість порівняно з кастомними моделями через спрощений інтерфейс без можливості глибокого налаштування;
  - залежність від стабільності сервісу постачальника та можливих змін у ціновій політиці.

- MVTec HALCON
  - висока вартість ліцензії, що може бути недоступною для малих підприємств;
  - висока складність впровадження та необхідність глибоких технічних знань для налаштування алгоритмів;
  - ресурсоемність системи, особливо при обробці зображень високої роздільної здатності;
  - залежність від кваліфікованого персоналу для постійної підтримки та оптимізації системи;
  - складність адаптації до нестандартних типів дефектів без ручного проектування алгоритмів.

### **Висновки за розділом 1**

У першому розділі роботи розглянуто теоретичні основи та аналіз сучасних систем комп'ютерного зору, з акцентом на їх принципи роботи, архітектуру та застосування для виявлення аномалій і дефектів на виробничих лініях. Описано поняття комп'ютерного зору, його ключові етапи та математичні основи, включаючи представлення зображень у вигляді матриць і використання згорткових нейронних мереж. Проведено огляд сучасних архітектур (VGG, ResNet, U-Net, YOLO, GAN тощо) та їх застосування в задачах виявлення аномалій. Проаналізовано традиційні методи обробки зображень, методи машинного та глибокого навчання, а також ефективність деяких комерційних рішень, з оцінкою їх точності, швидкості, масштабованості, доступності, адаптивності та обмежень, зокрема залежності від хмарної інфраструктури, вартості та потреби в анотованих даних.

## РОЗДІЛ 2

### РОЗРОБКА МОДЕЛІ ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ НА ЗОБРАЖЕННЯХ

#### 2.1 Аналіз та вибір методів для розробки моделі

У дослідженні розглядається задача виявлення аномалій на зображеннях різних видів продукції в промислових умовах, зокрема ідентифікація дефектів, таких як подряпини, тріщини, плями чи зміни текстури, на поверхнях матеріалів. Задача формулюється як навчання без вчителя для реконструкції нормальних зображень із подальшим виявленням аномалій за помилкою реконструкції. Такий підхід забезпечує універсальність моделі, дозволяючи адаптувати її до різних типів продукції без необхідності специфічних анотацій для кожного виду дефектів.

Основні вимоги до моделі:

- висока точність, щоб відповідати промисловим стандартам;
- висока швидкість обробки зображень для інтеграції в виробничі лінії;
- можливість роботи з обмеженим набором нормальних зображень без анотованих даних про дефекти;
- адаптивність до нових типів дефектів;
- доступність реалізації з використанням відкритих бібліотек для спрощення розробки.

Специфіка даних полягає у використанні лише нормальних зображень для навчання, щоб уникнути проблеми незбалансованості, коли аномалії є рідкісними або новими.

Для вибору кращого рішення методи порівнюються за критеріями: точність, швидкість, ресурси, адаптивність і доступність. Традиційні методи, такі як порогова сегментація, забезпечують високу швидкість і мінімальні вимоги до ресурсів, так як працюють на CPU, але їхня точність обмежена через низьку адаптивність до складних текстур і змін умов.

Методи класичного машинного навчання, наприклад SVM чи Isolation Forest, мають середню точність і швидкість, але потребують ручного проєктування ознак, що ускладнює їх універсальність для різних видів продукції.

Серед методів глибокого навчання згорткові автоенкодері (CAE) демонструють високу точність і адаптивність, оскільки здатні навчатися на нормальних зображеннях без анотацій, що вирішує проблему підбору даних для навчання. Вони потребують GPU, але використання попередньо навчених моделей зменшує обсяг даних і час навчання. Генеративно-змагальні мережі (GANomaly) забезпечують високу точність, але є складнішими в навчанні через необхідність балансування генератора і дискримінатора. Мережі для детекції (YOLO) і сегментації (U-Net) досягають високої точності, але потребують анотованих даних, що недоступні для даної задачі. Трансформери комп'ютерного зору (ViT) пропонують найвищу точність, але їхні обчислювальні вимоги роблять їх непрактичними для обмежених ресурсів.

На основі порівняльного аналізу для розробки моделі обрано згортковий автоенкодер із попередньо навченим ResNet-50 [5], як витягувачем ознак. Цей вибір обґрунтовується наступними факторами: по-перше, CAE ефективно вирішує задачу неконтрольованого навчання, моделюючи нормальні зображення та виявляючи аномалії за помилкою реконструкції, що відповідає вимогам універсальності для різних видів продукції. По-друге, використання ResNet-50 як витягувача ознак підвищує точність завдяки глибоким ієрархічним ознакам, зменшуючи потребу в великих обсягах даних для навчання. По-третє, метод забезпечує достатню швидкість обробки після оптимізації, що критично для промислових застосувань. По-четверте, CAE не потребує анотованих даних. Нарешті, реалізація в PyTorch [11] із використанням попередньо навчених моделей є доступною і відповідає сучасним стандартам розробки.

Обраний метод інтегрується в подальшу розробку через навчання на нормальних зображеннях із різних видів продукції, що забезпечує універсальність моделі. Використання ResNet-50 як витягувача ознак дозволяє ефективно обробляти зображення, а згортковий автоенкодер забезпечує виявлення аномалій шляхом порівняння реконструйованих і вихідних даних.

Згортковий автоенкодер із попередньо навченим ResNet-50, як витягувачем ознак, є оптимальним методом для розробки моделі виявлення аномалій завдяки високій точності, адаптивності до різних видів продукції та відсутності потреби в анотованих даних. Метод забезпечує баланс між обчислювальними вимогами і продуктивністю, що відповідає промисловим стандартам. Використання відкритих бібліотек і попередньо навчених моделей спрощує реалізацію, а універсальність підходу дозволяє застосовувати модель до різноманітних сценаріїв. Основним викликом є потреба в GPU для навчання, що планується компенсувати оптимізацією гіперпараметрів і використанням ефективних архітектур.

## **2.2 Архітектура системи комп'ютерного зору**

Архітектура системи комп'ютерного зору, розробленої для виявлення аномалій на зображеннях різних видів продукції, базується на принципах неконтрольованого навчання та реконструкції нормальних зображень. Система поєднує попередньо навчену згорткову нейронну мережу для витягування ознак із зображень і згортковий автоенкодер для моделювання нормального стану продукції, що дозволяє виявляти дефекти шляхом аналізу помилки реконструкції. Архітектура забезпечує універсальність, адаптивність до різних типів матеріалів і умов, а також відповідність промисловим вимогам щодо точності та швидкості. Система складається з кількох ключових компонентів: модуля введення даних, блоку витягування ознак, згорткового автоенкодера та модуля обробки результатів.

Модуль введення даних відповідає за отримання зображень через вибір файлів користувачем. Зображення подаються у форматі цифрових матриць у колірному просторі RGB, де кожен піксель містить інформацію про інтенсивність. Модуль забезпечує уніфікацію розміру зображень і їх перетворення в тензори, придатні для обробки нейронними мережами. Цей етап враховує вимоги до якості даних, такі як стабільне освітлення та висока роздільна здатність, і дозволяє користувачу гнучко обирати вхідні дані через інтерфейс, що підвищує доступність системи для промислових застосувань.

Блок витягування ознак відповідає за перетворення вхідних зображень у компактне представлення з високорівневими характеристиками. Для цього використовується попередньо навчена згортова нейронна мережа ResNet-50, яка ефективно вилучає ієрархічні ознаки завдяки глибокій архітектурі з залишковими з'єднаннями. ResNet-50 обробляє зображення, генеруючи карти ознак із проміжних шарів, які містять інформацію про текстуру, форму та інші візуальні характеристики, необхідні для аналізу різноманітних типів продукції. Використання попередньо навченої моделі зменшує потребу в великих обсягах даних, вирішуючи проблему обмеженої кількості анотованих зображень. Отримані карти ознак об'єднуються та нормалізуються для передачі до автоенкодера.

Центральним компонентом системи є згортковий автоенкодер (Convolutional Autoencoder), який моделює нормальний стан зображень шляхом їх стиснення в латентний простір і подальшої реконструкції. САЕ складається з енкодера, який стискає карти ознак від ResNet-50 у латентне представлення зниженої розмірності, і декодера, який відновлює ці карти, прагнучи мінімізувати різницю між вихідними та реконструйованими даними. Архітектура автоенкодера використовує згорткові шари для збереження просторових особливостей, що забезпечує ефективну обробку різноманітних текстур і типів продукції. САЕ підтримує неконтрольоване навчання, що дозволяє системі адаптуватися до нових видів матеріалів без анотацій.

Модуль обробки результатів відповідає за аналіз помилки реконструкції та представлення результатів користувачу через графічний інтерфейс. Система обчислює різницю між вихідними та реконструйованими картами ознак, ідентифікуючи аномалії. Графічний інтерфейс дозволяє користувачу обирати навчену модель і вхідні зображення, а також відображає результати аналізу у вигляді оригінального зображення, карти аномалій (з тепловою візуалізацією зон із високою помилкою) та виділеної зони дефектів. Інтерфейс забезпечує інтуїтивну взаємодію, підвищуючи доступність системи для користувачів без глибоких знань у комп'ютерному зорі. Результати можуть використовуватися для генерації сигналів у системах контролю якості або для подальшого аналізу.

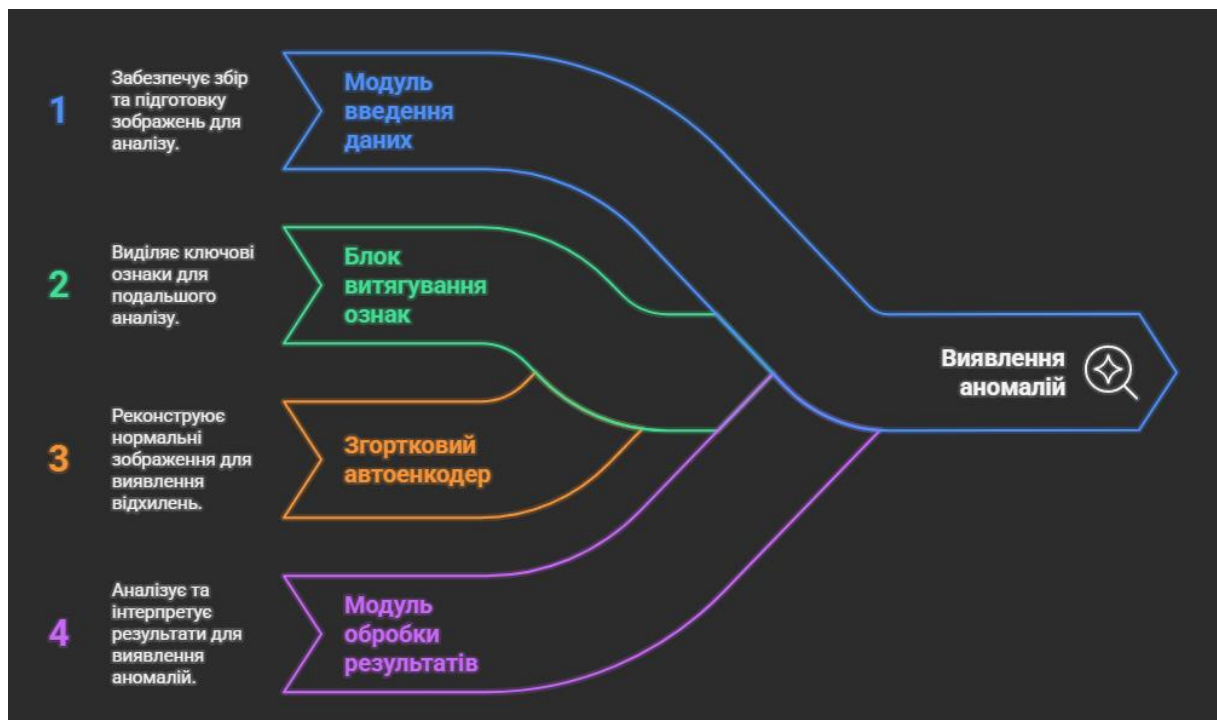


Рисунок 2.1 – Архітектура системи комп'ютерного зору.

### 2.3 Вибір програмних методів реалізації

Для створення системи комп'ютерного зору, яка виявляє дефекти на різних видах продукції, було обрані такі програмні інструменти, щоб вони були зручними і швидкими. Система має працювати точно, швидко

обробляти зображення і бути гнучкою. Для цього було обрано бібліотеки PyTorch, torchvision, tkinter, matplotlib, OpenCV [10] і numpy [9]. Вони прості у використанні, підтримують GPU і мають відкриту ліцензію, що важливо для доступності.

Бібліотека PyTorch була обрана як платформа для глибокого навчання — підходу, що використовує нейронні мережі для аналізу складних даних. PyTorch забезпечує гнучкість завдяки динамічному обчислювальному графу, що полегшує розробку й налагодження нейронних мереж, таких як згортковий автоенкодер. Підтримка апаратного прискорення на GPU дозволяє швидко обробляти зображення, що критично для промислових застосувань. Бібліотека дає змогу інтегрувати попередньо навчені моделі, наприклад ResNet-50, для витягування ознак, зменшуючи потребу в даних, що вирішує проблему обмеженої кількості анотацій. Відкрита ліцензія забезпечує доступність.

Для завантаження попередньо навчених моделей і підготовки зображень було використано бібліотеку torchvision, що входить до екосистеми PyTorch. Torchvision надає доступ до моделей, таких як ResNet-50, оптимізованих для обробки зображень у форматі RGB. Вона забезпечує функції для перетворення зображень у тензори, що необхідно для нейронних мереж. Використання готових моделей економить ресурси, що важливо через обмеженість даних. Бібліотека є простою у використанні та сумісною з PyTorch, що сприяє ефективній розробці.

Для роботи з багатовимірними масивами та числових обчислень було використано бібліотеку numpy. Вона забезпечує ефективну обробку цифрових матриць зображень, і підтримує швидкі операції, наприклад обчислення помилки реконструкції. Numpy є безкоштовною, сумісною з усіма вибраними бібліотеками й дозволяє адаптувати обчислення до різних типів матеріалів, що відповідає вимозі універсальності.

Для створення графічного інтерфейсу для взаємодії користувача з системою було обрано бібліотеку tkinter. Ця стандартна бібліотека Python

дозволяє розробляти кросплатформні інтерфейси. Tkinter забезпечує інтуїтивність, що підвищує доступність для користувачів без спеціалізованих знань. Вона легко інтегрується з іншими інструментами й не потребує додаткових залежностей, що спрощує розгортання системи.

Для відображення результатів, зокрема теплових карт та зображень, було використано бібліотеку `matplotlib`. Вона підтримує створення зображень, графіків і теплових карт, що необхідно для візуалізації помилки реконструкції. `Matplotlib` дозволяє показувати оригінальні зображення, карти аномалій і виділені дефекти, полегшуючи аналіз. Бібліотека є безкоштовною, сумісною з `tkinter` і `numpy`, і дає змогу налаштовувати візуалізацію для різних типів продукції, що забезпечує гнучкість.

Для виконання операцій із зображеннями, таких як створення бінарних масок і виділення аномалій, було обрано бібліотеку `OpenCV`. Вона є стандартом у комп'ютерному зорі, забезпечуючи швидку обробку. `OpenCV` оптимізовано для роботи на CPU, що зменшує залежність від GPU, і підтримує операції, необхідні для постобробки результатів. Відкрита ліцензія та сумісність із `numpy` і `matplotlib` сприяють інтеграції в систему й відповідають усім вимогам системи.

## **2.4 Вибір методу навчання моделі виявлення аномалій**

Для задачі виявлення аномалій на зображеннях промислової продукції, таких як подряпини, тріщини чи зміни текстури, необхідно обрати метод навчання, який забезпечує високу точність, швидкість обробки та можливість роботи з обмеженим набором даних без анотацій дефектів. З урахуванням специфіки задачі, де дефекти є рідкісними і можуть мати різноманітні форми, оптимальним вибором є неконтрольоване навчання або навчання без учителя. Цей підхід дозволяє моделі вивчати нормальний стан зображень без використання даних про аномалії, що усуває потребу в трудомісткому створенні анотованих наборів даних.

Неконтрольоване навчання базується на реконструкції нормальних зображень, де модель вчиться відтворювати вхідні дані, а аномалії виявляються за значними відхиленнями у помилці реконструкції. Такий метод є універсальним, оскільки не залежить від конкретних типів дефектів і може адаптуватися до різних видів продукції. Альтернативні підходи, такі як навчання з учителем, потребують великих анотованих наборів даних, що непрактично для промислових умов, де дефекти рідкісні або постійно з'являються нові типи. Напівконтрольоване навчання вимагає часткової анотації, що також ускладнює підготовку даних. Навчання з підкріпленням більше підходить для динамічних задач, таких як управління роботами, і не розглядається для аналізу зображень.

Для реалізації неконтрольованого навчання обрано підхід на основі згорткового автоенкодера (Convolutional Autoencoder), який використовує попередньо навчену модель ResNet-50 для витягування ознак. Згортковий автоенкодер навчається мінімізувати різницю між вихідними ознаками, отриманими з ResNet-50, і їхньою реконструйованою версією. Якщо зображення містить аномалію, автоенкодер, навчений лише на нормальних даних, не може точно відтворити дефектну область, що призводить до високої помилки реконструкції. Цей підхід дозволяє ефективно виявляти аномалії без необхідності знати їхні характеристики заздалегідь.

Для оцінки якості реконструкції використовується середньоквадратична похибка (MSE) як функція втрат.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2, \quad (2.1)$$

де  $n$  – кількість експериментів,  $Y_i$  – вхідне значення,  $\hat{Y}_i$  – передбачене значення.

MSE вимірює різницю між вхідними та реконструйованими ознаками, що забезпечує точне виявлення відхилень. Для оптимізації параметрів

моделі застосовується алгоритм Adam, який забезпечує швидку збіжність завдяки адаптивним темпам навчання.

Використання попередньо навченої моделі ResNet-50 як витягувача ознак дозволяє працювати з узагальненими та змістовними характеристиками, такими як текстура чи форма, замість сирих піксельних даних. Це підвищує точність моделі та зменшує потребу в великих обсягах навчальних даних, оскільки ResNet-50 уже навчена на наборі даних ImageNet. Неконтрольоване навчання на основі ознак, отриманих з ResNet-50, забезпечує універсальність моделі для різних типів зображень і промислових сценаріїв.

## 2.5 Проектування моделі

Проектування моделі спирається на теоретичні основи комп'ютерного зору та глибокого навчання, зокрема на концепцію автоенкодерів. Автоенкодери складаються з двох частин: енодера, який стискає вхідні дані в латентний простір зниженої розмірності, і декодера, який відновлює дані з цього простору.

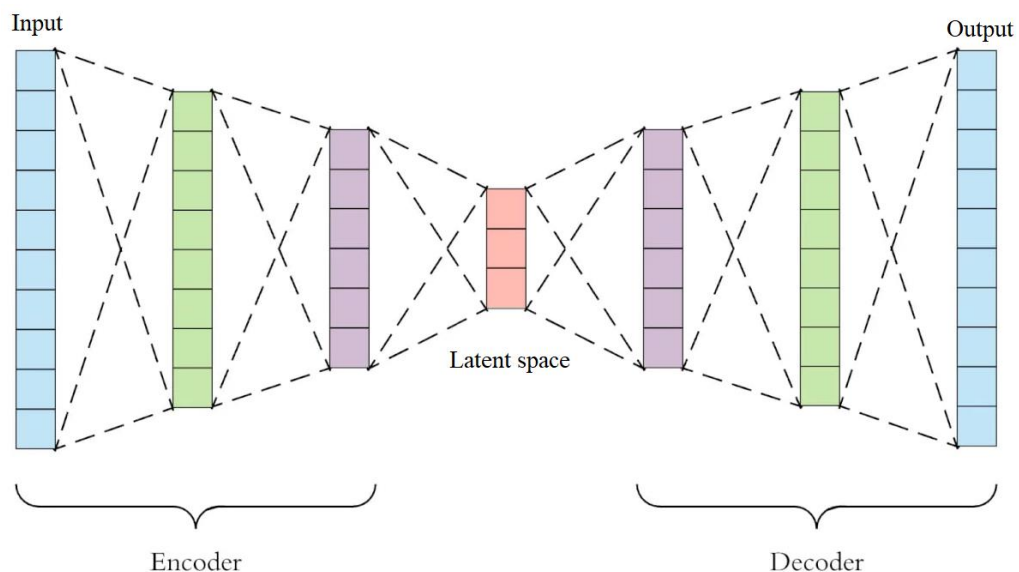


Рисунок 2.2 – Архітектура автоенкодеру.

Згорткові автоенкодери, на відміну від класичних повнозв'язних, використовують згорткові шари, що дозволяють зберігати просторову структуру зображень, що є критично важливим для аналізу текстур і форм у промислових зображеннях. Теоретичною перевагою такого підходу є здатність моделі узагальнювати нормальний стан об'єктів, уникаючи необхідності ручного визначення ознак, як у традиційних методах обробки зображень.

Архітектура моделі включає два ключові компоненти: блок витягування ознак і згортковий автоенкодер. Блок витягування ознак базується на ResNet-50, попередньо навченій моделі, яка широко використовується в задачах комп'ютерного зору завдяки своїй глибокій архітектурі та залишковим з'єднанням. Залишкові з'єднання (residual connections) дозволяють уникнути проблеми зникнення градієнта під час навчання глибоких мереж, що забезпечує стабільність і високу точність вилучення ознак. ResNet-50 приймає зображення у форматі RGB розміром 224x224 пікселі, які попередньо обробляються за допомогою трансформацій бібліотеки torchvision: масштабування до фіксованого розміру та перетворення в тензори. У реалізації на PyTorch параметри ResNet-50 заморожені, щоб уникнути їх оновлення, що зменшує обчислювальні витрати та дозволяє використовувати модель як витягувач ознак без додаткового навчання. Ознаки витягуються з проміжних шарів (layer2 і layer3) через механізм хуків, після чого об'єднуються за допомогою середнього пулінгу та адаптивного масштабування, формуючи карти ознак із 1536 каналами. Такий підхід забезпечує компактне представлення високорівневих характеристик, таких як текстура, краї чи форма, що є важливим для універсального аналізу різних типів продукції.

Згортковий автоенкодер, який є центральним компонентом моделі, спроектований для обробки карт ознак від ResNet-50. Енкодер стискає вхідні дані до латентного простору розмірністю 100, що дозволяє зменшити обсяг інформації, зберігаючи ключові характеристики нормальних

зображень. Декодер, у свою чергу, відновлює дані до початкової розмірності 1536 каналів. Архітектура енкодера складається з послідовності згорткових шарів, які поступово зменшують кількість каналів: від 1536 до 100. Кожен згортковий шар супроводжується нормалізацією для стабілізації навчання та активацією ReLU для додавання нелінійності. Декодер має симетричну структуру, збільшуючи кількість каналів у зворотному порядку, що забезпечує реконструкцію вхідних ознак.

Нормалізація шарів допомагає уникнути проблеми внутрішнього зсуву, що покращує збіжність моделі під час навчання. Функція активації ReLU забезпечує нелінійність, дозволяючи моделі краще моделювати складні патерни в даних. Латентний простір розмірністю 100 обрано як компроміс між компресією даних і збереженням достатньої кількості інформації для точної реконструкції.

Помилка реконструкції, яка є основою для виявлення аномалій, обчислюється як середньоквадратична похибка між вихідними та реконструйованими ознаками. MSE обрано через її простоту та ефективність у вимірюванні піксельних відхилень, що дозволяє чітко ідентифікувати аномалії. Оптимізація моделі здійснюється за допомогою алгоритму Adam із початковою швидкістю навчання 0.005, через його адаптивні темпи навчання, які враховують перші та другі моменти градієнтів, що особливо корисно для глибоких мереж із великою кількістю параметрів.

Таким чином, розроблена модель поєднує ResNet-50 для витягування ознак і згортковий автоенкодер для реконструкції, що забезпечує високу точність і адаптивність до промислових задач. Теоретичні принципи, такі як використання залишкових з'єднань, згорток і нормалізації, дозволяють оптимізувати продуктивність і зменшити обчислювальні витрати.

## 2.6 Навчальний набір даних

Формування навчального набору даних є ключовим етапом розробки системи комп'ютерного зору для виявлення аномалій на зображеннях промислової продукції. Оскільки система базується на безучительському навчанні, набір даних має містити лише нормальні зображення, щоб модель могла вивчити їхні характеристики та виявляти аномалії за відхиленнями у реконструкції. Для цього використано датасет MVTec AD [6], який є стандартним набором даних для задач виявлення аномалій у промислових умовах. Цей датасет вибрано через його високу якість, різноманітність категорій продукції та відповідність вимогам реальних виробничих сценаріїв.

Для навчання моделі використано виключно нормальні зображення (без дефектів), що відповідає підходу неконтрольованого навчання. Нормальні зображення в MVTec AD характеризуються високою роздільною здатністю, стабільними умовами освітлення та різноманітними текстурями, що дозволяє моделі вивчати широкий спектр візуальних характеристик. Тестові зображення з аномаліями, такими як подряпини, тріщини чи плями, використовуються лише для оцінки моделі, а не для навчання.

Обсяг навчального набору даних становить приблизно 200-300 зображень, що є достатнім для згорткового автоенкодера, особливо з урахуванням використання попередньо навченої ResNet-50, яка зменшує потребу в великих обсягах даних. Для підвищення узагальнення моделі використано техніку поділу даних на навчальну та валідаційну підмножини, що дозволяє контролювати перенавчання. У майбутніх розробках можна розширити набір даних за допомогою аугментації, щоб підвищити стійкість моделі до реальних умов.

## 2.7 Алгоритми підготовки та обробки зображень

Алгоритми підготовки та обробки зображень розроблено для забезпечення сумісності даних із моделлю, підвищення якості зображень і

виявлення аномалій у системі комп'ютерного зору. Процес включає попередню обробку, трансформацію даних для навчання та постобробку результатів.

Для підготовки даних зображення попередньо обробляються за допомогою бібліотеки `torchvision`. Зокрема, застосовуються трансформації розміру зображень до 224x224 пікселів та їхнього перетворення в тензори. Ці трансформації забезпечують сумісність із вхідним форматом ResNet-50, яка очікує триканальні зображення RGB розміром 224x224. Набір даних розділено на навчальну та тестову підмножини у співвідношенні 70% до 30% за допомогою функції `random_split`, що дозволяє оцінити ефективність моделі на невикористаних даних.

## 2.8 Стратегія валідації моделі

Валідація моделі проводиться на тестовій підмножині нормальних зображень, щоб оцінити якість реконструкції, та на тестових зображеннях із аномаліями, щоб перевірити здатність моделі виявляти дефекти. Для оцінки якості реконструкції використовується середня валідаційна втрата, яка обчислюється як середнє значення MSE на тестових партіях. Для оцінки виявлення аномалій застосовується порогова сегментація, де карта помилок реконструкції порівнюється з пороговим значенням. Зображення вважається аномальним, якщо середня помилка реконструкції перевищує поріг. Цей підхід дозволяє кількісно оцінити чутливість моделі до аномалій.

Теоретичне обґрунтування валідації базується на принципі оцінки узагальнюючої здатності моделі. Тестова підмножина нормальних зображень дозволяє перевірити, чи модель адекватно відтворює нормальний стан, тоді як зображення з аномаліями оцінюють її здатність виявляти відхилення. Порогова сегментація є стандартним методом для задач виявлення аномалій, оскільки вона проста у реалізації та адаптивна до різних типів даних шляхом налаштування порогу. У коді результати валідації візуалізуються за допомогою `matplotlib`, де відображаються

графіки навчальної та валідаційної втрат, що допомагає оцінити стабільність навчання. Наприклад, зменшення обох втрат із кожною епохою свідчить про успішне навчання, тоді як розбіжність між ними може вказувати на перенавчання.

## **Висновки за розділом 2**

У другому розділі роботи описано розробку моделі для виявлення аномалій на зображеннях промислової продукції та підготовку навчального набору даних. Задачу сформульовано як задачу неконтрольованого навчання з використанням згорткового автоенкодера з попередньо навченим ResNet-50 для витягування ознак, що забезпечує високу точність, швидкість і універсальність без потреби в анотованих даних. Архітектура системи включає модулі введення даних, витягування ознак, згорткового автоенкодера та обробки результатів із графічним інтерфейсом для візуалізації аномалій. Для реалізації обрано бібліотеки PyTorch, torchvision, numpy, tkinter, matplotlib і OpenCV через їхню гнучкість, швидкість і відкриту ліцензію. Навчання базується на нормальних зображеннях із датасету, використовуючи MSE як функцію втрат і алгоритм Adam для оптимізації. Дані підготовлено шляхом трансформації до розміру 224x224 пікселів і поділу на навчальну та тестову підмножини. Валідація моделі проводиться за допомогою MSE та порогової сегментації, оцінюючи якість реконструкції та здатність виявляти дефекти, з візуалізацією результатів для аналізу стабільності навчання.

## РОЗДІЛ 3

### ПРОГРАМНА РЕАЛІЗАЦІЯ ТА АНАЛІЗ ЕКСПЕРИМЕНТАЛЬНИХ РЕЗУЛЬТАТІВ

#### 3.1 Програмна реалізація моделі виявлення аномалій

Програмна реалізація моделі для виявлення аномалій на зображеннях промислової продукції розроблена з використанням мови програмування Python [12] та бібліотек для глибокого навчання та обробки зображень. Основою моделі є згортковий автоенкодер у поєднанні з попередньо навченою нейронною мережею ResNet-50. Реалізація базується на неконтрольованому навчанні, що дозволяє моделі вивчати нормальний стан зображень із датасету і виявляти аномалії за помилкою реконструкції. Код структуровано для забезпечення високої швидкості обробки, адаптивності до різних типів матеріалів і зручності інтеграції з графічним інтерфейсом користувача.

Бібліотека PyTorch забезпечує гнучкість завдяки динамічному обчислювальному графу та підтримку апаратного прискорення на GPU. PyTorch обрано через її широку підтримку в задачах глибокого навчання, зручність налагодження та сумісність із сучасними бібліотеками комп'ютерного зору. Додатково використано бібліотеку torchvision для завантаження попередньо навченої моделі ResNet-50 та обробки зображень та numpy для числових обчислень.

Модель виявлення аномалій складається з двох основних компонентів: витягувача ознак і автоенкодера. Витягувач ознак реалізовано через клас `resnet_feature_extractor`, який використовує попередньо навчену модель ResNet-50 із ваговими коефіцієнтами ImageNet (`ResNet50_Weights.DEFAULT`). Параметри ResNet-50 заморожені (`param.requires_grad = False`), щоб уникнути їх оновлення та зменшити обчислювальні витрати. Ознаки витягуються з проміжних шарів (`layer2` і `layer3`) за допомогою хуків (`register_forward_hook`), після чого об'єднуються

через середній пулінг (`nn.AvgPool2d`) та адаптивне масштабування (`nn.AdaptiveAvgPool2d`), формуючи карти ознак із 1536 каналами. Цей підхід дозволяє отримати компактне представлення високорівневих характеристик, таких як текстура чи форма, що є важливим для універсального аналізу.

Згортковий автоенкодер реалізовано через клас `FeatCAE`, який складається з енкодера та декодера. Енкодер стискає вхідні карти ознак (1536 каналів) до латентного простору розмірністю 100 через три згорткові шари: від 1536 до 868, потім до 200 і до 100. Кожен шар включає нормалізацію (`nn.BatchNorm2d`) та активацію `ReLU` (`nn.ReLU`) для стабільності та нелінійності. Декодер має симетричну структуру, відновлюючи дані від 100 до 1536 каналів.

Модуль обробки результатів включає постобробку та візуалізацію. На етапі тестування зображення обробляються аналогічно: масштабуються, перетворюються в тензори та передаються в `ResNet-50` для витягування ознак. Автоенкодер реконструює ці ознаки, а помилка реконструкції обчислюється як середньоквадратична похибка. Карта помилок масштабується до розміру `224x224` за допомогою білінійної інтерполяції (`nn.functional.interpolate`). Для виділення аномалій застосовується порогова сегментація, що створює бінарну маску (`anomaly_map > threshold`). Маска масштабується до початкового розміру зображення через `cv2.resize` і накладається на оригінальне зображення для виділення аномальних зон. Результати візуалізуються за допомогою `matplotlib` у вигляді трьох зображень: оригінального, теплової карти помилок (з колірною схемою `"jet"`) та аномальної зони.

### 3.2 Навчання та валідація моделі

Навчання та валідація моделі є ключовими етапами, які визначають її точність і здатність до узагальнення. Модель навчається за допомогою неконтрольованого підходу, використовуючи лише нормальні зображення з

датасету. Основна мета навчання – мінімізувати помилку реконструкції ознак, які виділяються за допомогою ResNet-50. Під час валідації перевіряється, наскільки добре модель відновлює зображення і чи може вона виявляти аномалії.

Для експерименту було обрано два датасети – “carpet” і “screw”. Навчання моделі відбувалося з різною кількістю епох, щоб побачити, як довготривале навчання впливає на кінцеві результати. Розмір батча встановлено на 16 зображень, що дозволяє ефективно використовувати обчислювальні ресурси і забезпечує стабільне оновлення ваг моделі.

Валідація проводилась на окремій тестовій вибірці нормальних зображень для оцінки якості реконструкції. Відбувалося відстежування як training loss під час навчання, так і validation loss на тестових даних. Training loss показує, наскільки добре модель вчиться відновлювати навчальні зображення, тоді як validation loss дає уявлення про те, чи не перенавчається модель і чи зможе вона добре працювати на нових даних.

Розглянемо навчання на датасеті “carpet”. Першим експериментом було проведено 5 епох навчання. Графік зміни функції втрат у процесі навчання нейронної мережі протягом 5 епох представлено на рис. 3.1.

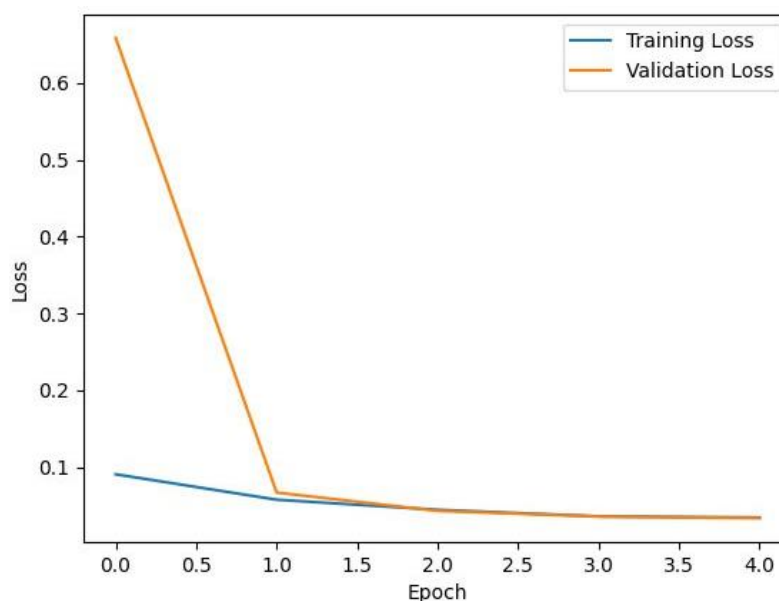


Рисунок 3.1 – Графік втрат під час навчання у 5 епох.

Модель швидко навчається, Training Loss та Validation Loss стабілізуються та майже однакові, це ознака хорошого навчання, модель добре працює та не перенавчилася.

На другому експерименті було проведено 50 епох навчання. Розглянемо графік на рис. 3.2.

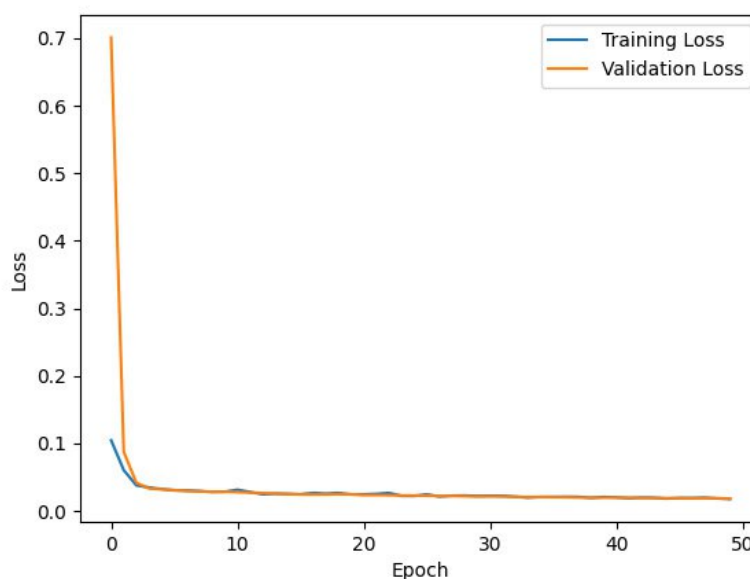


Рисунок 3.2 – Графік втрат під час навчання у 50 епох.

На початку навчання значення обох Loss дуже різко зменшується, це показник того, що модель швидко навчається розпізнавати базові закономірності. Протягом всього навчання Training Loss та Validation Loss залишаються стабільно низькими, перенавчання немає, обидві криві співпадають.

Результат проведених 100 епох навчання представлено на рис. 3.3.

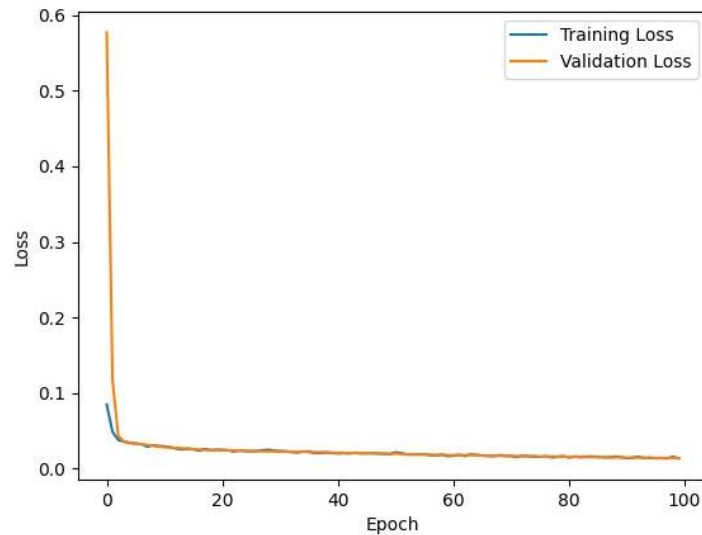


Рисунок 3.3 – Графік втрат під час навчання у 100 епох.

Результати схожі на навчання у 50 епох, модель добре навчається, немає перенавчання.

Перейдемо до навчання на датасеті “screw”. Датасет характеризується більшою деталізацією та складністю мікроструктури порівняно з датасетом “carpet”, що зажадало більш детального аналізу та розширеного набору експериментів під час навчання моделі. Тому було проведено експерименти з навчання на 10, 50 та 100 епохах. Динаміку зміни втрат протягом навчання мережі за 10 епох представлено на рис. 3.4.

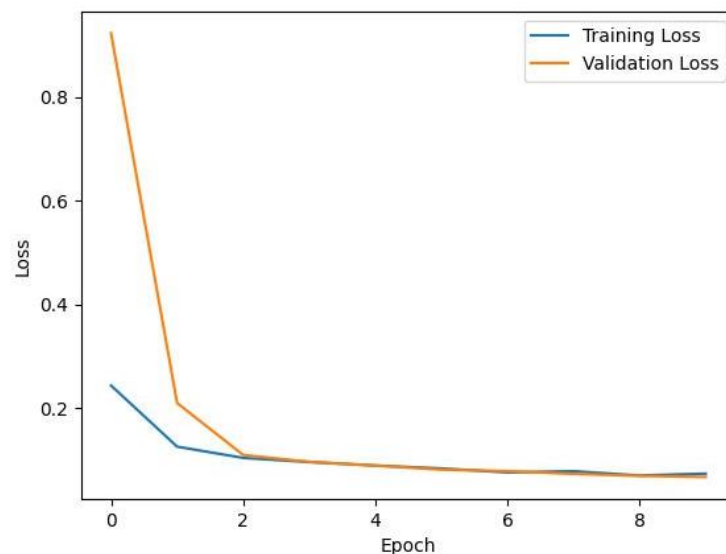


Рисунок 3.4 – Графік втрат під час навчання у 10 епох.

Видно, що тренувальні втрати починаються з рівня приблизно 0.25 і швидко знижуються до 0.12 вже на другій епосі, після чого стабілізуються на низькому рівні. Валідаційні втрати мають більш значне зниження - з початкових 0.9 до 0.2 на другій епосі, а потім поступово зменшуються до 0.08. Загалом графік показує успішне навчання моделі з хорошою конвергенцією, хоча валідаційні втрати на початку були значно вищими за тренувальні.

Процес навчання протягом 50 епох демонструє графік на рис. 3.5.

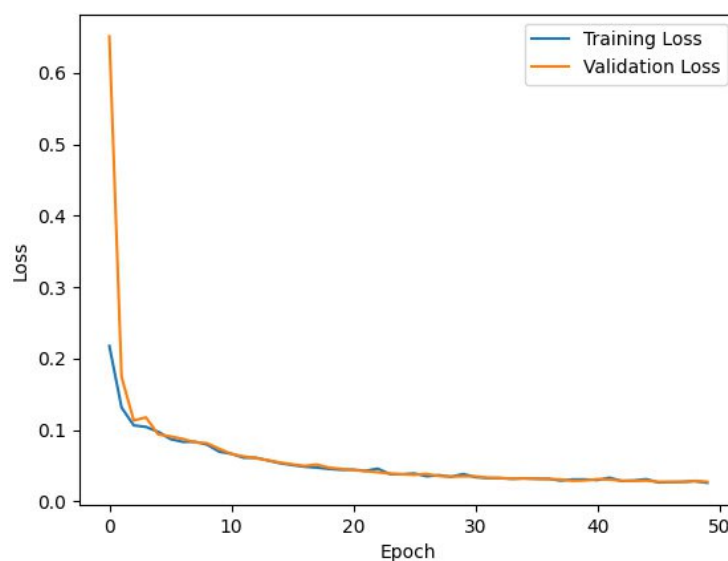


Рисунок 3.5 – Графік втрат під час навчання у 50 епох.

Спостерігається схожа поведінка – різке падіння втрат на перших епохах з подальшою стабілізацією. Тренувальні втрати знижуються з 0.22 до приблизно 0.02, а валідаційні – з 0.65 до 0.03. Найбільш активна фаза навчання відбувається в перші 15-20 епох, після чого обидві криві виходять на плато. Це свідчить про те, що модель досягла оптимального стану і подальше навчання не приносить значних покращень.

На графіку навчання протягом 100 епох бачимо найбільш детальну картину процесу навчання (рис. 3.6).

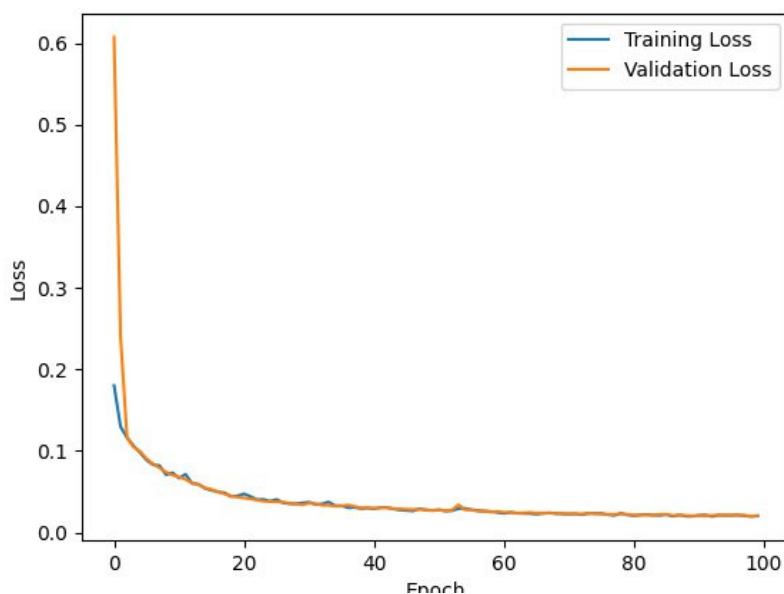


Рисунок 3.6 – Графік втрат під час навчання у 100 епох.

Початкові значення втрат складають 0.2 для тренувальних і 0.6 для валідаційних даних. Найінтенсивніше зниження відбувається протягом перших 25-30 епох, після чого криві практично не змінюються і залишаються на рівні близько 0.02. Такий тривалий період стабільності підтверджує, що модель повністю навчилася і досягла мінімуму функції втрат.

### 3.3 Інтерфейс користувача системи виявлення аномалій

Інтерфейс користувача системи виявлення аномалій розроблений з урахуванням потреб промислових користувачів, які можуть не мати спеціальних знань у програмуванні або машинному навчанні, але потребують простого інструменту для швидкого аналізу зображень на предмет дефектів. Для створення інтерфейсу було використано бібліотеку tkinter у Python, вона дозволяє легко створювати графічний інтерфейс і працює на різних операційних системах. Основна ідея полягала в тому, щоб користувач міг завантажити навчену модель, вибрати зображення для перевірки і отримати результати у зрозумілому вигляді: оригінальне

зображення, теплову карту помилок і бінарну маску з виділеними аномаліями.

Інтерфейс реалізовано в класі `AnomalyDetectorApp`, який створює головне вікно програми з набором елементів управління (рис. 3.7).

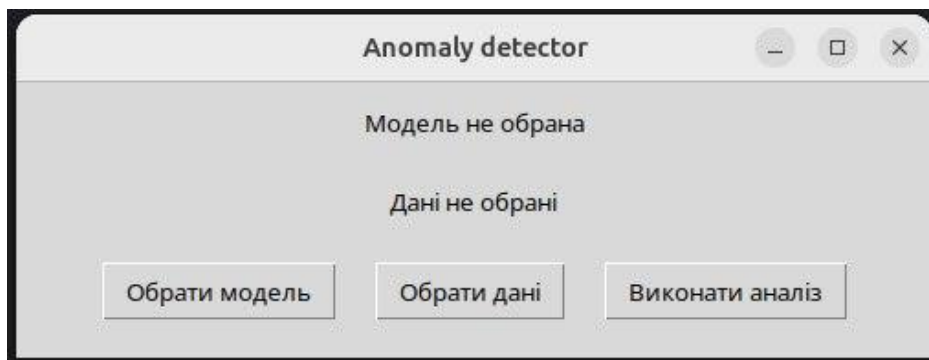


Рисунок 3.7 – Головне меню програми.

Програма має три основні кнопки: `select_model` для завантаження файлу моделі у форматі `.pth`, `select_data` для вибору зображення та `execute` для запуску процесу аналізу. Важливою особливістю програми є система перевірок – користувач не може запустити аналіз, поки не буде вибрана модель (рис. 3.8) і не завантажені дані для обробки (рис. 3.9).

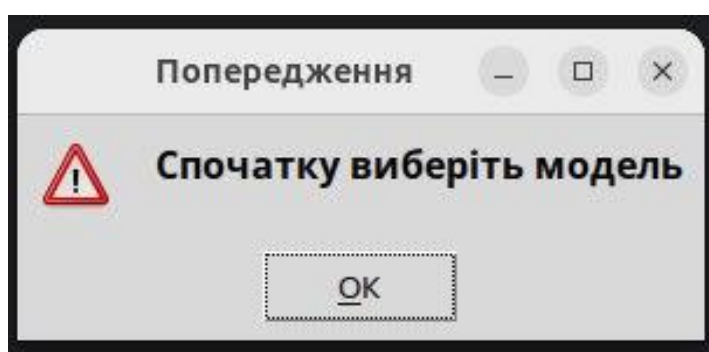


Рисунок 3.8 – Вікно попередження про необхідність вибору моделі.

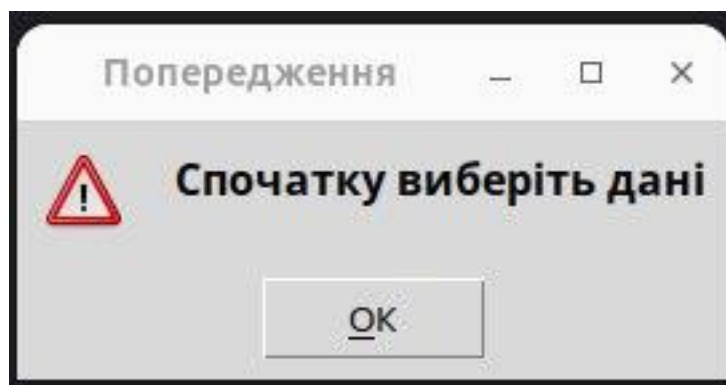


Рисунок 3.9 – Вікно попередження про необхідність вибору даних.

При натисканні кнопки "Обрати модель" чи "Обрати дані" відкривається діалогове вікно, яке дозволяє користувачу здійснювати пошук по файловій системі та обрати файли. Вікно має фільтр файлів, який показує лише відповідні файли, що спрощує пошук потрібного файлу серед інших документів у папці (рис. 3.9).

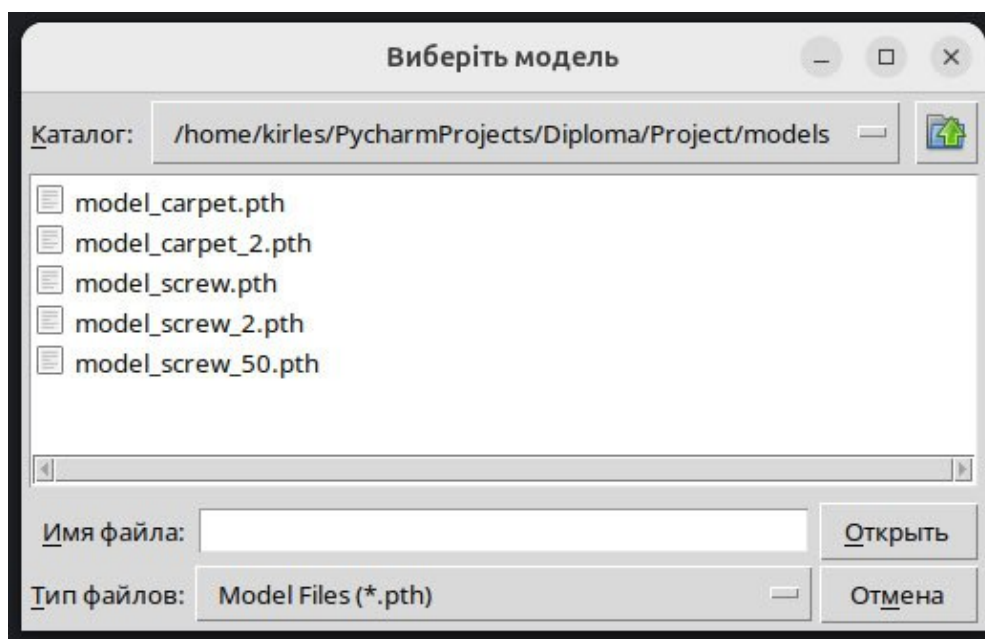


Рисунок 3.9 – Діалогове вікно вибору моделі.

Також було додано текстові поля, які показують назву вибраної моделі та кількість обраних для обробки зображень (рис. 3.10).

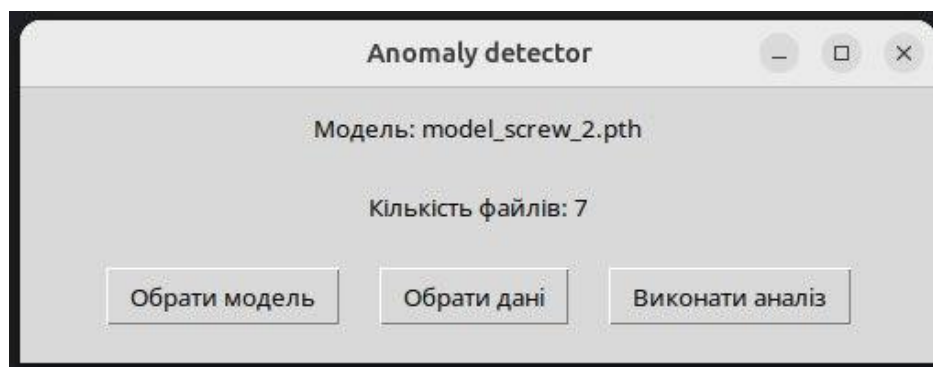


Рисунок 3.10 – Головне меню програми з обраними даними.

При натисканні кнопки «Виконати аналіз» запускається аналіз зображень, результати відображаються у вигляді трьох зображень поруч одне з одним: спочатку показується оригінальне зображення, потім теплова карта, яка показує рівень помилки реконструкції в різних частинах зображення, і нарешті бінарна маска, що чітко виділяє знайдені аномальні зони (рис. 3.11).

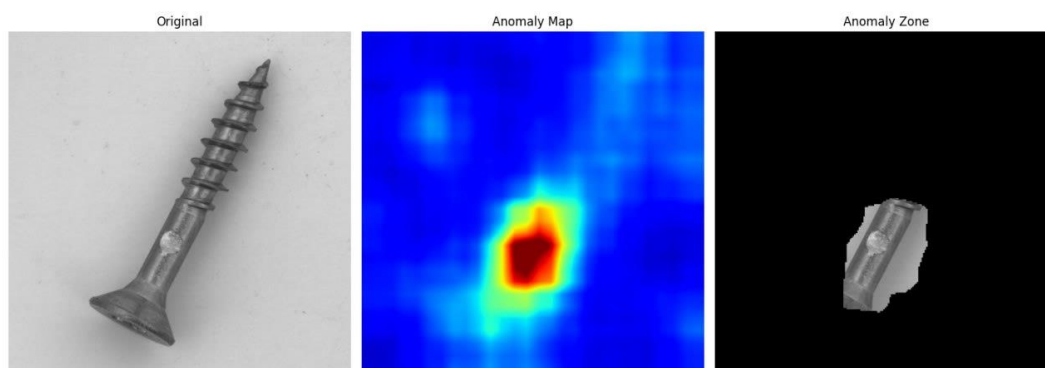


Рисунок 3.11 – Результат виконання аналізу.

### 3.4 Проведення та аналіз експериментальних досліджень

Після успішного навчання моделей було проведено серію експериментів для оцінки їх ефективності у виявленні аномалій. Тестування здійснювалося на зображеннях двох типів: нормальних (без дефектів) та аномальних (з дефектами). Для кожної навченої моделі аналізувалися

результати роботи алгоритму, що включали візуалізацію карт аномалій та виділення зон з підвищеним рівнем аномальності.

### 3.4.1 Експериментальні дослідження на датасеті «carpet»

Результати тестування моделі з однією епохою навчання на нормальному зображенні показують незадовільну ефективність алгоритму реконструкції. Карта аномалій характеризується великою червоною зоною, що вказує на критично високий рівень помилки відтворення навіть для стандартних зразків без дефектів (рис. 3.12).

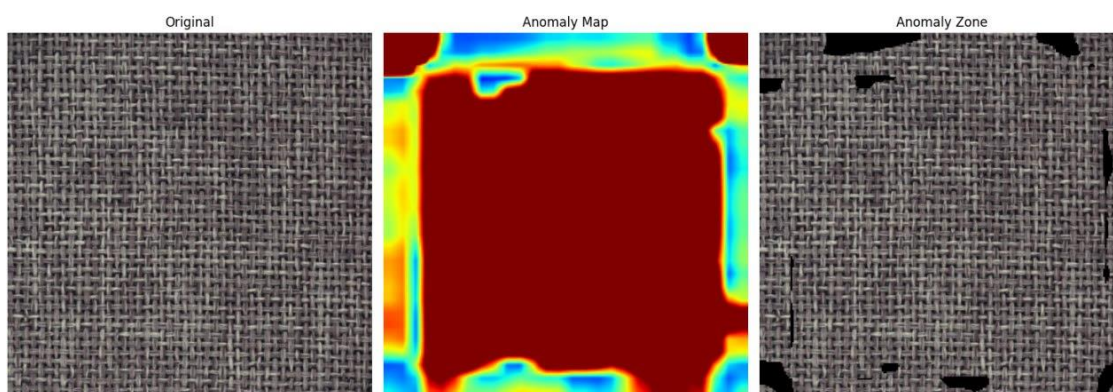


Рисунок 3.12 – Результат виконання аналізу на тестовому зображенні без дефекту на моделі навченою протягом 1 епохи.

Модель, навчена протягом 5 епох, демонструє значне покращення якості реконструкції порівняно з попереднім експериментом. Для тестового зображення без аномалії карта аномалій має синє забарвлення по всій площі зображення, що вказує на низьку помилку реконструкції та правильне розпізнавання нормальної текстури (рис. 3.13).

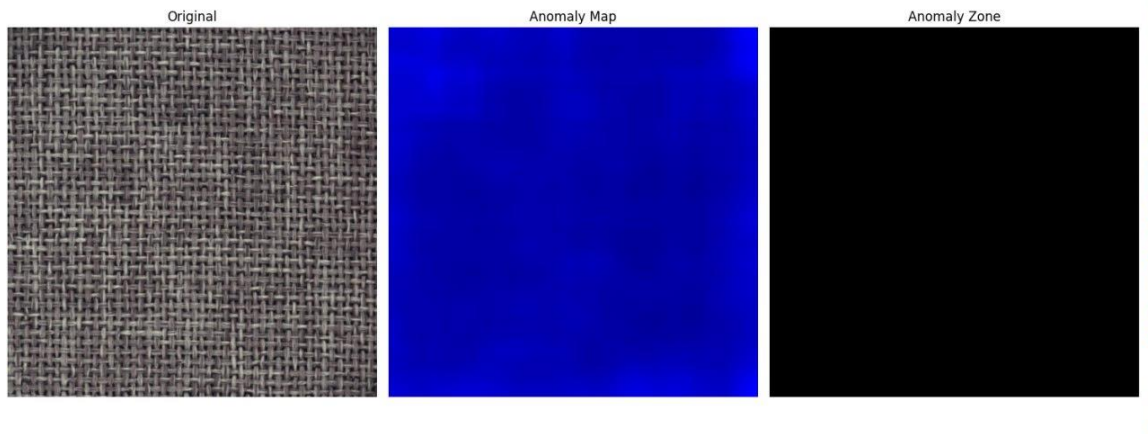


Рисунок 3.13 – Результат виконання аналізу на тестовому зображенні без дефекту на моделі навченою протягом 5 епох.

При аналізі зображення з дефектом модель показує відмінні результати виявлення аномалій. На карті аномалій чітко виділяється аномальна зона, що точно локалізує місце розташування дефекту (рис. 3.14).

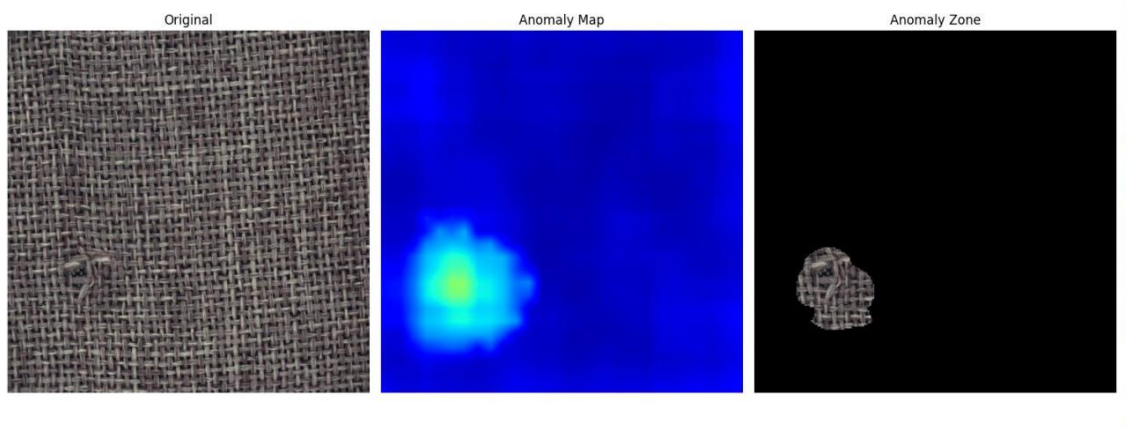


Рисунок 3.14 – Результат виконання аналізу на тестовому зображенні з дефектом на моделі навченою протягом 5 епох.

Нормальні ділянки зображення залишаються синіми, що демонструє здатність моделі розрізняти аномальні та нормальні області. Модель успішно навчилася відтворювати характеристики нормальних зразків з датасету та виявляти аномалії на поганих зразках.

### 3.4.2 Експериментальні дослідження на датасеті «screw»

При тестуванні моделі, навченою протягом 5 епох, на тестовому нормальному зображенні результати показують низьку якість реконструкції. Карта аномалій має переважно червоне забарвлення по всій площі зображення, що свідчить про високу помилку відтворення. Модель неправильно інтерпретує нормальні елементи та поверхні як аномальні зони (рис. 3.15).

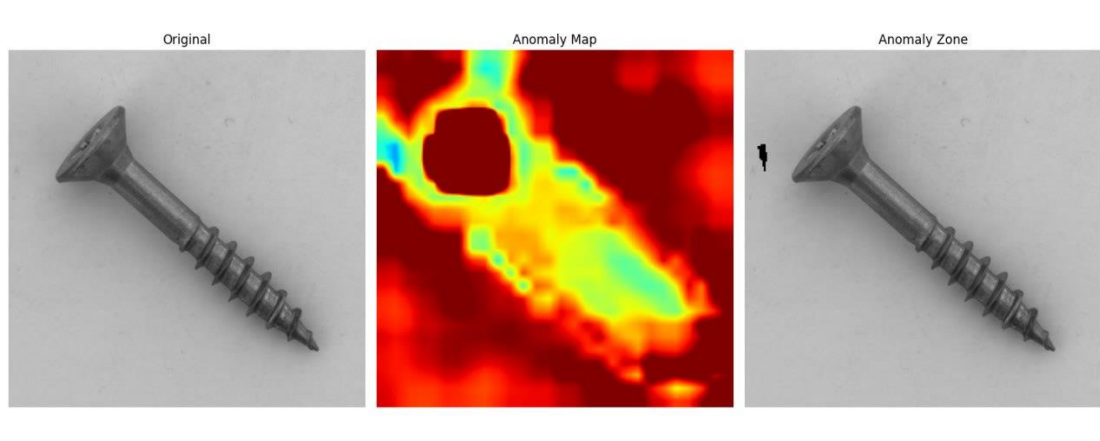


Рисунок 3.15 – Результат виконання аналізу на тестовому зображенні без дефекту на моделі навченою протягом 5 епох.

Модель з 50 епохами навчання при тестуванні на нормальному зображенні показує помітне покращення, проте результати залишаються ненайкращими. На карті аномалій спостерігається змішане забарвлення, переважно сині зони з окремими жовтими зонами (рис. 3.16).

При аналізі зразка з дефектами модель виявляє надмірну чутливість, класифікуючи практично весь об'єкт як аномальний. Карта аномалій охоплює не лише область справжнього дефекту, але й значну частину нормальних ділянок. Така поведінка свідчить про те, що модель ще не досягла оптимального балансу між чутливістю до дефектів та точністю локалізації (рис. 3.17).

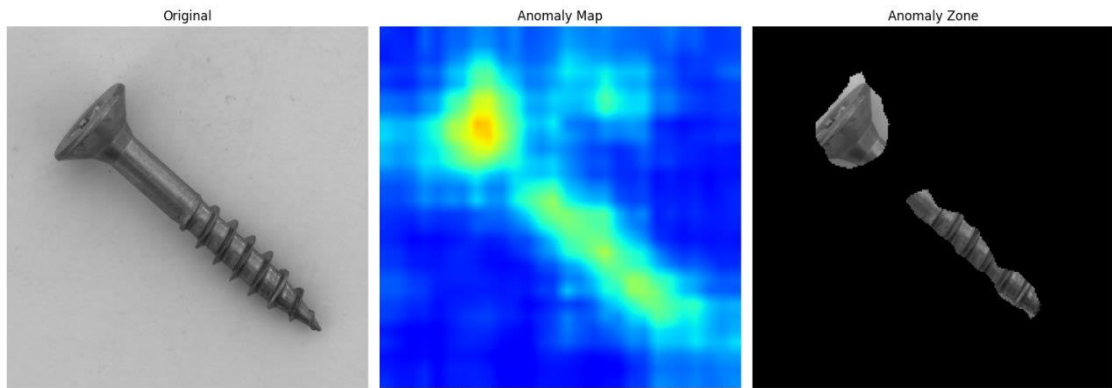


Рисунок 3.16 – Результат виконання аналізу на тестовому зображенні без дефекту на моделі навченою протягом 50 епох.

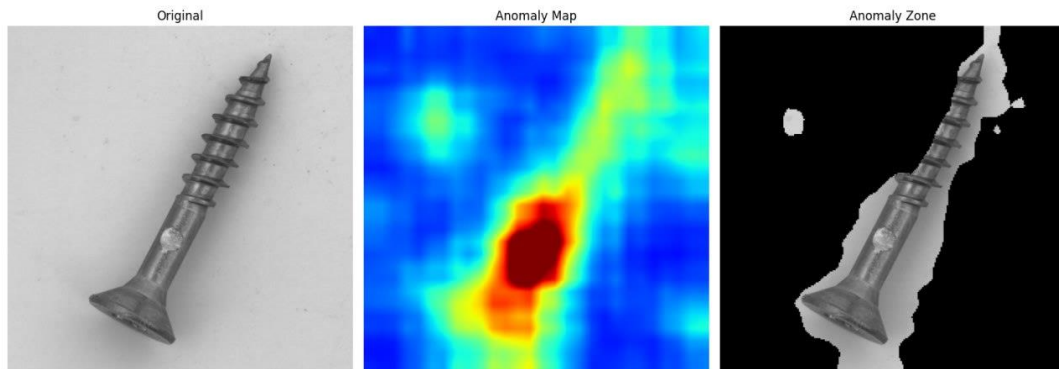


Рисунок 3.17 – Результат виконання аналізу на тестовому зображенні з дефектом на моделі навченою протягом 50 епох.

Модель зі 100 епохами навчання демонструє відмінні результати при аналізі нормального зразку. Карта аномалій має переважно синє забарвлення по всій площі зображення, що вказує на низьку помилку реконструкції та правильне розпізнавання стандартних характеристик виробу. Зона аномалій залишається чорною, підтверджуючи відсутність виявлених дефектів. Це свідчить про успішне навчання моделі розпізнавати нормальні зразки болтів (рис. 3.18).

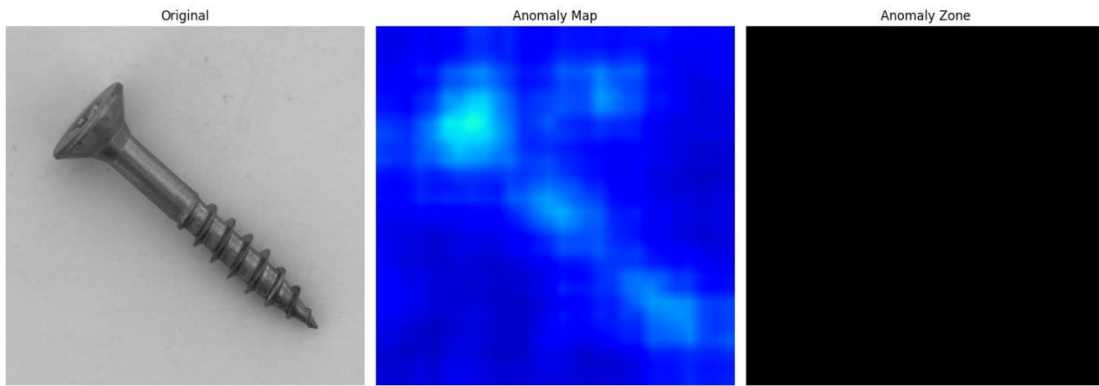


Рисунок 3.18 – Результат виконання аналізу на тестовому зображенні без дефекту на моделі навченою протягом 100 епох.

При тестуванні на дефектному зразку модель показує оптимальну ефективність детекції. На карті аномалій чітко локалізується червона зона, що точно відповідає розташуванню реального дефекту, тоді як нормальні ділянки залишаються синіми. У зоні аномалій виразно відображається виявлений дефект без захоплення нормальних областей (рис. 3.19).

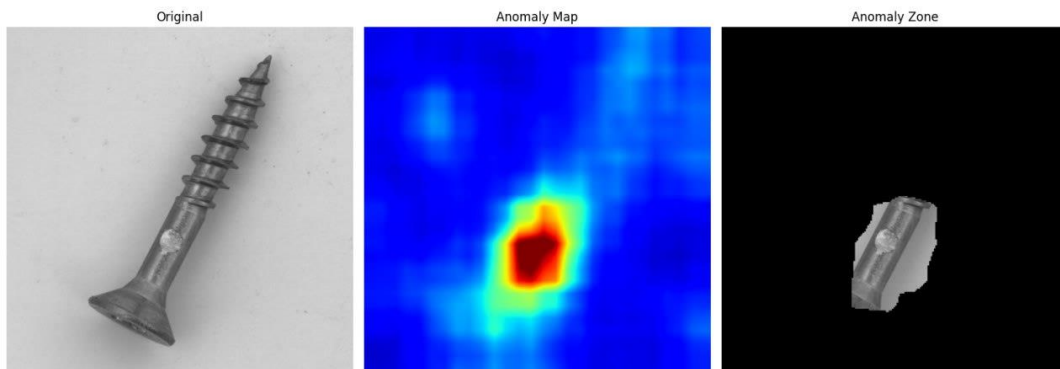


Рисунок 3.19 – Результат виконання аналізу на тестовому зображенні з дефектом на моделі навченою протягом 100 епох.

### Висновки за розділом 3

У третьому розділі роботи описано програмну реалізацію та аналіз експериментальних результатів системи виявлення аномалій на зображеннях промислової продукції. Модель, побудована на основі

згорткового автоенкодера з ResNet-50 як витягувачем ознак, реалізована в Python з використанням бібліотек PyTorch, torchvision, numpy, tkinter, matplotlib та OpenCV. Реалізація включає витягування ознак із проміжних шарів ResNet-50, стиснення їх автоенкодером у латентний простір та обчислення помилки реконструкції для виявлення аномалій. Навчання проводилося на датасетах "carpet" і "screw" з різною кількістю епох (5, 50, 100), демонструючи стабільне зниження тренувальних і валідаційних втрат без ознак перенавчання. Для "carpet" модель досягла хорошої якості реконструкції вже після 5 епох, ефективно виявляючи дефекти, тоді як для складнішого датасету "screw" оптимальні результати отримані після 100 епох, з чіткою локалізацією аномалій. Інтерфейс користувача, створений за допомогою tkinter, забезпечує інтуїтивне завантаження моделі та зображень, відображення оригінальних зображень, теплових карт і бінарних масок аномалій. Експерименти підтвердили високу точність і адаптивність моделі.

## ВИСНОВКИ

У кваліфікаційній роботі було розглянуто використання систем комп'ютерного зору для розв'язання задачі виявлення аномалій і дефектів на зображеннях промислової продукції з необхідністю роботи з обмеженими даними без анотацій.

Актуальність розробки полягає в підвищенні ефективності контролю якості на виробничих лініях, що дозволяє автоматизувати процеси, зменшуючи вплив людського фактору та забезпечуючи стабільну точність. У роботі використаний підхід неконтрольованого навчання на основі згорткового автоенкодера з попередньо навченою моделлю ResNet-50, що забезпечує універсальність і високу точність виявлення аномалій без потреби в анотованих даних. Практичне значення одержаних результатів полягає у створенні системи комп'ютерного зору, яка досягає високої точності за менший час навчання, та розробці зручного графічного інтерфейсу для промислових користувачів.

У першому розділі проведено аналіз і порівняння існуючих методів і моделей комп'ютерного зору, включаючи традиційні методи обробки зображень, класичне машинне навчання та сучасні підходи глибокого навчання. Виявлено, що методи глибокого навчання, зокрема згорткові нейронні мережі, значно перевершують традиційні методи за точністю та адаптивністю, хоча й потребують більших обчислювальних ресурсів.

У другому розділі розроблено модель на основі згорткового автоенкодера з ResNet-50 для витягування ознак, описано архітектуру системи, процеси підготовки даних і методи навчання. Обґрунтовано вибір неконтрольованого навчання, що дозволяє моделі працювати з нормальними зображеннями та виявляти аномалії за помилкою реконструкції і забезпечує універсальність для різних типів продукції. Використано бібліотеки PyTorch, torchvision, numpy, tkinter, matplotlib і OpenCV, а також датасет MVTec AD для тестування навчання та валідації.

У третьому розділі описано програмну реалізацію моделі, яка включає витягування ознак із ResNet-50, стиснення даних автоенкодером і обчислення помилки реконструкції для виявлення аномалій. Проведено експериментальні дослідження на датасетах “carpet” і “screw” з різною кількістю епох (5, 50, 100), що показали стабільне зниження втрат і високу точність після 100 епох, особливо для складнішого датасету “screw”. Розроблено інтуїтивний графічний інтерфейс на базі tkinter, який забезпечує зручну взаємодію користувача з системою, відображення оригінальних зображень, теплових карт і бінарних масок аномалій.

Аналіз результатів показав, що запропонована модель досягає високої точності виявлення аномалій за менший час навчання порівняно з альтернативними методами, зокрема завдяки використанню попередньо навченої ResNet-50 і оптимізації гіперпараметрів. У середньому модель продемонструвала низькі значення функції втрат і чітку локалізацію дефектів після 100 епох. Система має значні переваги перед існуючими аналогами, завдяки універсальності, адаптивності та доступності реалізації.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. AMD ROCm Platform [Електронний ресурс]. – режим доступу: URL: <https://www.amd.com/en/products/software/rocm.html> (дата звернення – 18.05.2025)
2. Baur C. Autoencoders for unsupervised anomaly segmentation in brain MR images: A comparative study / C. Baur, S. Denner, N. Navab, S. Albarqouni // *Medical Image Analysis*. – 2021. – Vol. 69. – Article 101952.
3. Chandola V. Anomaly detection: A survey / V. Chandola, A. Banerjee, V. Kumar // *ACM Computing Surveys*. – 2009. – Vol. 41, No. 3. – P. 1–58.
4. Chen Y. Deep learning-based anomaly detection for manufacturing systems / Y. Chen, X. Zhou, T. S. Huang // *IEEE Transactions on Industrial Informatics*. – 2021. – Vol. 17, No. 8. – P. 5244–5253.
5. He K. Deep Residual Learning for Image Recognition / K. He, X. Zhang, S. Ren, J. Sun // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. – 2016. – P. 770–778.
6. MVTec Research Datasets [Електронний ресурс]. – режим доступу: URL: <https://www.mvtec.com/company/research/datasets> (дата звернення – 20.05.2025)
7. Napoletano P. Anomaly detection in nanofibrous materials by CNN-based self-similarity / P. Napoletano, F. Piccoli, R. Schettini // *Sensors*. – 2018. – Vol. 18, No. 1. – Article 209.
8. NVIDIA CUDA Documentation [Електронний ресурс]. – режим доступу: URL: <https://docs.nvidia.com/cuda/index.html> (дата звернення – 18.05.2025)
9. NumPy documentation [Електронний ресурс]. – режим доступу: URL: <https://numpy.org/doc/2.2/> (дата звернення – 17.05.2025)
10. OpenCV Documentation [Електронний ресурс]. – режим доступу: URL: <https://docs.opencv.org> (дата звернення – 17.05.2025)

11. PyTorch Tutorials: [Електронний ресурс]. – режим доступу: URL: <https://pytorch.org/tutorials/> (дата звернення – 16.05.2025)
12. Python 3.12 Documentation [Електронний ресурс]. – режим доступу: URL: <https://docs.python.org/3.12/> (дата звернення – 15.05.2025)
13. Rawat W. Deep convolutional neural networks for image classification: A comprehensive review / W. Rawat, Z. Wang // Neural Computation. – 2017. – Vol. 29, No. 9. – P. 2352–2449.
14. Yi J. Deep learning based anomaly detection in manufacturing processes / J. Yi, T. Yoon // Journal of Manufacturing Systems. – 2021. – Vol. 60. – P. 572–584.
15. Zhou Y. Anomaly detection and localization in manufacturing systems using deep learning / Y. Zhou, X. Liang, W. Zhang, L. Zhang, X. Xie // IEEE Transactions on Industrial Electronics. – 2022. – Vol. 69, No. 2. – P. 1824–1834.
16. Тимчишин Р. М. Сучасні підходи до розв'язання задач комп'ютерного зору / Р.М. Тимчишин, О.Є. Волков, О.Ю. Господарчук, Ю.П. Богачук // Control systems and computers. — 2018. - № 6. - С. 46-73.

## ДОДАТКИ

### Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Харківський національний університет імені В. Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту  
Кафедра комп'ютерних систем та робототехніки  
Рівень вищої освіти (освітньо-кваліфікаційний рівень) Бакалавр  
Галузь знань: 12 – Інформаційні технології  
Спеціальність: 123 «Комп'ютерна інженерія»  
Освітня програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри комп'ютерних  
систем та робототехніки  
к. ф.-м. н., доц. ХРУСЛОВ М. М.  
«02» жовтня 2024 року

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

#### **Буштець Кирило Віталійович**

(прізвище, ім'я, по батькові студента)

1. Тема роботи «Модель виявлення аномалій на виробничих лініях за допомогою систем комп'ютерного зору»

керівник роботи Стрілець Вікторія Євгенівна, кандидат технічних наук,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від *16 квітня 2025 року № 4101-5/962*

2. Строк подання студентом роботи *30 травня 2025 року*

3. Перелік питань, які потрібно розробити:

1. Аналіз існуючих технологій, підходів, методів і моделей виявлення аномалій на виробничих лініях.
2. Створення моделі виявлення аномалій на виробничих лініях на основі системи комп'ютерного зору.
3. Тестування та оцінка ефективності розробленої моделі виявлення аномалій на виробничих лініях.
4. Аналіз подальших можливостей розвитку запропонованої моделі.

## 4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Затвердження теми роботи з керівником	05.09.2024 - 01.10.2024
2	Пошук та аналіз наукової літератури за темою кваліфікаційної роботи.	02.10.2024 - 24.10.2024
3	Аналіз технологій та підходів до виявлення аномалій на виробничих лініях.	02.10.2024 - 24.10.2024
4	Огляд методів і моделей виявлення аномалій на основі систем комп'ютерного зору.	25.10.2024 - 09.11.2024
5	Вибір програмних технологій для розробки моделі.	10.11.2024 - 24.11.2024
6	Розробка моделі виявлення аномалій із використанням підходів машинного навчання.	25.10.2024 - 08.11.2024
7	Збір даних для навчання моделі	09.11.2024 - 29.11.2024
8	Програмна реалізація та навчання моделі виявлення аномалій на виробничих лініях.	30.11.2024 - 31.12.2024
9	Тестування та оцінка ефективності моделі виявлення аномалій на виробничих лініях.	01.01.2025 - 01.02.2025
10	Аналіз можливостей подальшого вдосконалення запропонованої моделі.	01.03.2025 - 30.04.2025-
11	Підготовка і оформлення звітних матеріалів та додатків кваліфікаційної роботи. Оформлення списку літератури	01.04.2025 - 30.04.2025-
12	Оформлення пояснювальної записки кваліфікаційної роботи відповідно вимогам до звітів про НДР.	01.05.2025 - 30.05.2025-
13	Оформлення звіту про переддипломну практику	01.05.2025 - 30.05.2025-
14	Представлення кваліфікаційної роботи керівнику та рецензенту	30.05.2025

5. Дата видачі завдання **02 жовтня 2024 року.**

Студент

Буштець К.В.

  
 підпис

Керівник роботи

Стрілець В.Є.

  
 підпис

Затверджую

« \_\_\_\_\_ » \_\_\_\_\_ 2024 р.

**Технічне завдання  
на розробку програмного виробу  
«Модель виявлення аномалій на виробничих лініях за  
допомогою систем комп'ютерного зору»**

Назва розділу	Назва та зміст підрозділу
1. Вступ	<p>1.1. Назва програмного виробу: «Модель виявлення аномалій на виробничих лініях за допомогою систем комп'ютерного зору»</p> <p>1.2. Галузь застосування: Модель виявлення аномалій на виробничих лініях за допомогою систем комп'ютерного зору застосовується в промисловій автоматизації та контролі якості. Вона може використовуватися для моніторингу виробничих процесів у режимі реального часу, виявлення дефектів продукції, відхилень у розміщенні або формі компонентів, а також для запобігання браку ще на етапі виробництва. Такі системи активно впроваджуються в автомобільній, електронній, харчовій, фармацевтичній, текстильній та інших галузях, де необхідна висока точність контролю якості і зменшення людського фактору.</p>
2. Підстава для розробки	<p>2.1. Навчальний план за спеціальністю: 123 - Комп'ютерна інженерія, Харківський національний університет імені В. Н. Каразіна, Навчально-науковий інститут комп'ютерних наук та штучного інтелекту.</p> <p>2.2. Завдання на кваліфікаційну роботу: №4101-5/962 від 16.04.2025 Розробка Моделі виявлення аномалій на виробничих лініях за допомогою систем комп'ютерного зору.</p>
3. Призначення розробки	<p>3.1. Мета розробки програмного виробу: Метою розробки програмного виробу є підвищені ефективності виявлення дефектів продукції на виробничих лініях за допомогою систем</p>

	<p>комп'ютерного зору. Реалізація цієї системи дозволить знизити кількість дефектної продукції, скоротити час на перевірку виробів, зменшити витрати на ручний контроль та підвищити загальну продуктивність виробничого процесу.</p> <p>Для досягнення цієї мети програмний виріб має вирішувати такі завдання:</p> <ul style="list-style-type: none"> <li>– автоматичний аналіз зображень;</li> <li>– виявлення відхилень та дефектів;</li> <li>– фіксація та збереження інформації про виявлені аномалії;</li> <li>– надання звіту про результати перевірки;</li> <li>– інтеграція з існуючим обладнанням виробничої лінії.</li> </ul> <p>3.2. Призначення програмного виробу: Цей програмний виріб призначений для автоматичного виявлення аномалій на виробничих лініях за допомогою комп'ютерного зору з метою підвищення якості продукції та оптимізації процесу технічного контролю.</p> <p>3.3. Вихідні дані для розробки :</p> <ul style="list-style-type: none"> <li>– Використання трансферного навчання на попередньо навченій моделі, наприклад, ResNet50, для покращення результатів класифікації при обмеженій кількості даних.</li> <li>– Для побудови моделі використовується високоякісний датасет зображень виробів, отриманий з виробничої лінії, з маркуванням аномалій.</li> </ul>
<p>4. Технічні вимоги до програмного виробу</p>	<p>4.1. Вимоги до функціональних характеристик:</p> <ul style="list-style-type: none"> <li>– автоматичне завантаження та обробка зображень в режимі реального часу;</li> <li>– виявлення аномалій та дефектів на зображеннях за допомогою попередньо навченої моделі нейронної мережі;</li> <li>– візуальне відображення зображення з підсвічуванням виявлених дефектів;</li> <li>– можливість збереження результатів у локальну або хмарну базу даних;</li> <li>– час пошуку дефекту одного зображення не повинен перевищувати 2 секунд на персональному комп'ютері середнього рівня.</li> </ul> <p>4.2. Вимоги до надійності:</p>

- модель має бути стійкою до невеликих змін у зображенні;
- система повинна перевіряти формат, роздільну здатність та цілісність зображень перед їх обробкою. У разі виявлення пошкоджених або нечитабельних файлів – виводиться повідомлення про помилку та здійснюється пропуск кадру;
- програмний виріб повинен проходити тестування на безперервну роботу;
- програмний виріб має зберігати цілісність моделей та вихідних даних навіть у разі збоїв у роботі.

#### 4.3. Вимоги до умов експлуатації:

- Кліматичні умови: температура навколишнього середовища: від +10 °С до +35 °С; відносна вологість повітря: не більше 80% при температурі +25 °С; відсутність агресивного пилу, прямих сонячних променів та джерел сильного електромагнітного випромінювання поблизу робочого місця.
- Програмний виріб доступний для завантаження з репозиторію. Для запуску достатньо завантажити архів з файлами, розпакувати його на локальний диск і запустити відповідний виконуваний файл або скрипт.
- Програма працює автономно на локальному комп'ютері та не потребує додаткових налаштувань або мережевих з'єднань з віддаленими серверами.
- Після запуску користувач може одразу приступити до аналізу зображень з виробничих ліній шляхом завантаження фото або відео у визначене вікно інтерфейсу.
- Робота здійснюється в офлайн-режимі — після початкового встановлення програмний виріб не потребує доступу до мережі Інтернет.
- Для ефективного використання програми користувачу рекомендується ознайомитися з короткою інструкцією з експлуатації.

#### 4.4. Вимоги до складу параметрів технічних засобів:

- Програмний виріб може бути запущений на персональному комп'ютері з будь-якою з поширених операційних систем: Windows,

	<p>Linux/Unix, Mac OS X, OS/2 або Amiga, за умов наявності встановленого інтерпретатора Python версії 3.x або вище.</p> <ul style="list-style-type: none"> <li>– Мінімальні апаратні вимоги включають підключення до електромережі, щонайменше 8 ГБ оперативної пам'яті, 1 ГБ вільного місця на диску та процесор із тактовою частотою від 3.0 до 3.5 ГГц.</li> <li>– Для покращення швидкодії обробки зображень рекомендується використовувати комп'ютер із дискретною відеокартою. У разі використання GPU NVIDIA потрібна підтримка бібліотеки CUDA. Для графічних процесорів AMD необхідно встановити платформу ROCm, сумісну з відповідною моделлю відеокарти.</li> </ul> <p>4.5. Вимоги до інформаційної та програмної сумісності</p> <ul style="list-style-type: none"> <li>– Програма працює автономно під керуванням сучасних ОС: Windows 10/11, Linux (Ubuntu 20.04+), macOS 10.15+. Потрібен Python 3.12+ та бібліотеки PyTorch 2.7.0, NumPy 2.0.1 тощо.</li> <li>– Вхідні дані — зображення у форматах .jpg/.png, вихід — зображення у форматах .jpg/.png</li> <li>– Код написаний на Python, легко вбудовується до більших систем. Захист даних не реалізовано, оскільки продукт призначено для локального навчального та дослідницького використання.</li> </ul> <p>4.6. Вимоги до маркування та упаковки</p> <p>Вимоги до маркування та упаковки не висувуються, оскільки програмний продукт є цифровим рішенням і не потребує фізичної упаковки для використання в навчальних, дослідницьких або виробничих умовах.</p> <p>4.7. Вимоги до транспортування та зберігання</p> <p>Вимоги до транспортування та зберігання відсутні, оскільки продукт є програмним забезпеченням, яке може бути передане в електронному вигляді та не потребує спеціальних умов зберігання.</p> <p>4.8. Спеціальні вимоги</p> <p>Спеціальні вимоги до програмного продукту не передбачені. Система розроблена як універсальне рішення для виявлення аномалій у виробничих процесах і не вимагає додаткового апаратного або</p>
--	--

	<p>програмного забезпечення, окрім стандартних компонентів, зазначених у вимогах до експлуатації.</p>
5. Вимоги до програмної документації.	<p>Програмна документація до виробу "Модель виявлення аномалій на виробничих лініях за допомогою систем комп'ютерного зору":</p> <ol style="list-style-type: none"> <li>1. Це Технічне завдання на розробку програмного виробу (представити у вигляді Додатка Б до пояснювальної записки до кваліфікаційної роботи).</li> <li>2. Програму та методика випробувань розробленого програмного виробу (представити у вигляді Додатка В до пояснювальної записки до кваліфікаційної роботи).</li> <li>3. Опис програмного виробу (представити в розділі пояснювальної записки до кваліфікаційної роботи).</li> <li>4. Текст програми (представити в Додатку Г до пояснювальної записки до кваліфікаційної роботи).</li> </ol>
6. Техніко-економічні показники	<ol style="list-style-type: none"> <li>1. Планується досягти точності виявлення аномалій на виробничих лініях на рівні близько 90%, враховуючи складність завдання та варіації виробничих умов.</li> <li>2. Розробка моделі триватиме орієнтовно 3 місяці. Для її створення будуть використовуватися безкоштовні програмні засоби та доступне апаратне забезпечення. У разі необхідності масштабування для обробки великих обсягів даних чи в реальному часі, можливі витрати на продуктивніші графічні процесори (GPU), такі як NVIDIA або AMD, що підтримують CUDA чи ROCm.</li> <li>3. Очікується, що модель матиме подібний або вищий рівень ефективності порівняно з наявними рішеннями. Вона буде доступною для інтеграції в промислові та дослідницькі системи завдяки спрощеній реалізації та можливості використання на різних етапах виробничого процесу. Крім того, модель може бути адаптована для освітніх чи дослідницьких застосунків, що робить її більш доступною та легко впроваджуваною.</li> </ol>
7. Стадії та етапи розробки	<p>Розробка моделі класифікації архітектурних споруд проходить у три основні стадії відповідно до ДЕСТ 19.102-77:</p> <p>Стадії:</p> <ol style="list-style-type: none"> <li>1. Розробка технічного завдання— формування вимог, постановка задачі, погодження з керівником.</li> <li>2. Робоче проектування— включає програмування, налагодження, розробку документації та випробування.</li> </ol>

	<p>3. Впровадження розробленого виробу— підготовка, передача програми та документації до експлуатації.</p> <p>Етапи:</p> <ol style="list-style-type: none"> <li>1. Розробка програми (підбір і підготовка датасету, створення моделі, тренування та валідація моделі, коригування за потреби, тестування, створення звітів за результатами роботи моделі).</li> <li>2. Розробка програмної документації.( створення пояснювальної записки, візуалізацій результатів).</li> <li>3. Впровадження (випробування) програми (передача готової моделі для практичного або навчального використання).</li> </ol>
<p>8. Порядок контролю та приймання</p>	<ol style="list-style-type: none"> <li>1. Перевірка ходу розроблення заданих проектних документів керівнику кваліфікаційної роботи здійснювати 1 раз на 3 тижні.</li> <li>2. Випробування програмного виробу.</li> <li>3. Захист розробленого програмного виробу провести на засіданні ДЕК.</li> <li>4. Розроблені проектні документи подати в електронному вигляді.</li> </ol>

Виконавець

Буштець Кирило Віталійович



Замовник

Стрілець Вікторія Євгенівна



**Додаток В**

**Програма та методика випробувань програмного виробу  
«Модель виявлення аномалій на виробничих лініях за  
допомогою систем комп'ютерного зору»**

**1. Об'єкт випробувань****1.1 Найменування програмного виробу:**

Модель виявлення аномалій на виробничих лініях за допомогою систем комп'ютерного зору.

**1.2 Область застосування:**

Модель виявлення аномалій на виробничих лініях за допомогою систем комп'ютерного зору застосовується в промисловій автоматизації та контролі якості. Вона може використовуватися для моніторингу виробничих процесів у режимі реального часу, виявлення дефектів продукції, відхилень у розміщенні або формі компонентів, а також для запобігання браку ще на етапі виробництва. Такі системи активно впроваджуються в автомобільній, електронній, харчовій, фармацевтичній, текстильній та інших галузях, де необхідна висока точність контролю якості і зменшення людського фактора.

**1.3 Умовне призначення розробки (за потреби).**

Цей програмний виріб призначений для автоматизації процесу виявлення аномалій на виробничих лініях за допомогою комп'ютерного зору. Він розроблений для використання в промислових умовах, а також може бути застосований у навчальних та дослідницьких цілях в галузі штучного інтелекту та автоматизації виробничих процесів.

## **2. Мета випробувань**

Метою випробувань є перевірка відповідності функціональних характеристик розробленого програмного виробу вимогам, зазначеним у Технічному завданні. Зокрема, це підтвердження здатності моделі автоматично виявляти аномалії на виробничих лініях за допомогою комп'ютерного зору, з урахуванням різних варіацій виробничих умов та факторів, що можуть впливати на ефективність класифікації. Випробування мають також на меті оцінку ефективності використання нейронних мереж для зниження часу на виявлення аномалій, забезпечуючи стабільну точність результатів. Це дозволяє зменшити людський фактор, зменшити час реакції на потенційні проблеми та забезпечити стабільний рівень якості класифікації для подальшого аналізу та коригування виробничих процесів.

## **3. Загальні положення**

### **3.1 Підстави щодо випробувань**

*Підставою для проведення випробувань є: наказ ХНУ №\_\_\_ від \_\_\_\_\_ про затвердження тем кваліфікаційних робіт бакалаврів.*

### **3.2 Місце та тривалість випробувань**

*Приймальні випробування проводяться на базі комп'ютерного класу кафедри.*

### **3.3. Обсяг випробувань**

Приймальні випробування програмного виробу проводяться в обсязі відповідно до цієї Програми та методики випробувань.

### **3.4 Організації, які беруть участь у випробуваннях**

*Приймальні випробування проводяться за участю Замовника та Виконавця..*

#### **4. Вимоги до програмного виробу**

- Програмний виріб повинен автоматично виконувати виявлення аномалій на виробничих лініях на основі вхідного зображення.
- Програмний виріб має підтримувати обробку зображень у форматах .jpg, .jpeg та .png.
- Програмний виріб повинен мати можливість виведення результатів на екран, а також збереження результатів у файл для подальшого аналізу чи архівації.
- Користувач повинен мати можливість завантажити зображення, вказавши шлях до файлу.
- Максимальний обсяг оперативної пам'яті, що використовується під час класифікації одного зображення, не повинен перевищувати 2000 МБ.
- Час класифікації одного зображення не повинен перевищувати 3 секунд на персональному комп'ютері середнього рівня.

#### **5. Вимоги до програмної документації**

Програмна документація до виробу "Модель виявлення аномалій на виробничих лініях за допомогою систем комп'ютерного зору":

- 1) Це Технічне завдання на розробку програмного виробу (представити у вигляді Додатка Б до пояснювальної записки до кваліфікаційної роботи).
- 2) Програму та методику випробувань розробленого програмного виробу (представити у вигляді Додатка В до пояснювальної записки до кваліфікаційної роботи).
- 3) Опис програмного виробу (представити в розділі пояснювальної записки до кваліфікаційної роботи).

4) Текст програми (представити в Додатку Г до пояснювальної записки до кваліфікаційної роботи).

## **6. Засоби та порядок випробувань**

### **6.1 Засоби випробувань**

Для проведення випробувань буде використовуватись наступне технічне та програмне забезпечення:

#### **Технічні засоби:**

- Сервери або робочі станції з достатньою кількістю оперативної пам'яті та потужними графічними процесорами (GPU) для прискорення обчислень.
- Вихідні зображення архітектурних об'єктів для тестування та виявлення аномалій.

#### **Програмні засоби:**

- Python — основна мова програмування для реалізації моделі глибокого навчання.
  - PyTorch — фреймворк для створення та навчання нейронної мережі, використовуючи архітектуру ResNet50 для класифікації архітектурних стилів.
  - PyCharm— середовище для розробки, в якому буде здійснюватися написання коду та його виконання.
  - NumPy, Pandas — для обробки даних та аналізу результатів.
  - Matplotlib — для візуалізації результатів тестування та графічного представлення ефективності моделі.
  - OpenCV — для обробки зображень.
- Та інші.

### **6.2 Порядок проведення випробувань**

#### **1-й підетап. Перевірка комплектності програмної документації**

Перевіряється наявність усіх необхідних документів згідно з Технічним завданням. Перевірка комплектності технічних та програмних засобів здійснюється за критерієм відповідності їх складу вимогам середовища виконання

### **2-й підетап. Перевірка якості програмної документації**

Якість програмної документації відповідає вимогам ДЕСТ 19.301–79 ЄСПД.

### **2-й етап – функціональне випробування програмного виробу**

На цьому етапі перевіряється працездатність моделі та ступінь виконання функціональних вимог.

## **1. Перевірка працездатності програмного виробу**

### **Методика проведення Тесту 1:**

- Запуск тестування моделі здійснюється у середовищі PyCharm.
- Вибирається тестове одне зображення з вказаної директорії.
- Програма виконує:
  - завантаження моделі PyTorch;
  - попередню обробку зображення;
  - виконання передбачення;
  - вимір часу виконання та обсягу використаної пам'яті;
  - виведення зображення із виділеною зоною аномалії;

### **Очікуваний результат:**

- Виведення зображення із виділеною зоною аномалії
- Вивід часу виконання та спожитої пам'яті

### **Фактичний результат:**

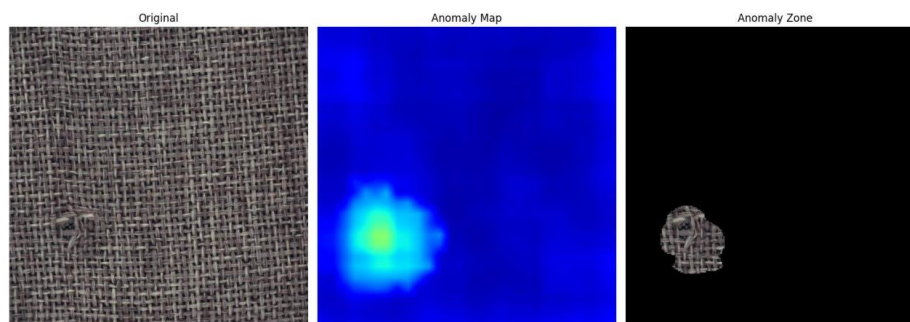


Рисунок В.1 – Результат виявлення зони аномалії. Тест 1

```
[INFO] Time elapsed: 1.502 seconds
[INFO] Memory usage: 1042.03 MB

Process finished with exit code 0
```

Рисунок В.2 – Результат виконання тесту 1 по швидкості і часу

**Висновок:** у результаті виконання Тесту 1 функціональна вимога щодо виявлення аномалії виконана. А також було виміряно час виконання обробки одного зображення та використання пам'яті, що не перевищує вимоги.

### Методика проведення Тесту 2:

- Запуск тестування моделі здійснюється у середовищі PyCharm.
- Вибирається тестове одне зображення з вказаної директорії.
- Програма виконує:
  - завантаження моделі PyTorch;
  - попередню обробку зображення;
  - виконання передбачення;
  - вимір часу виконання та обсягу використаної пам'яті;
  - виведення зображення із виділеною зоною аномалії;

### Очікуваний результат:

- Виведення зображення із виділеною зоною аномалії
- Вивід часу виконання та спожитої пам'яті

## Фактичний результат:

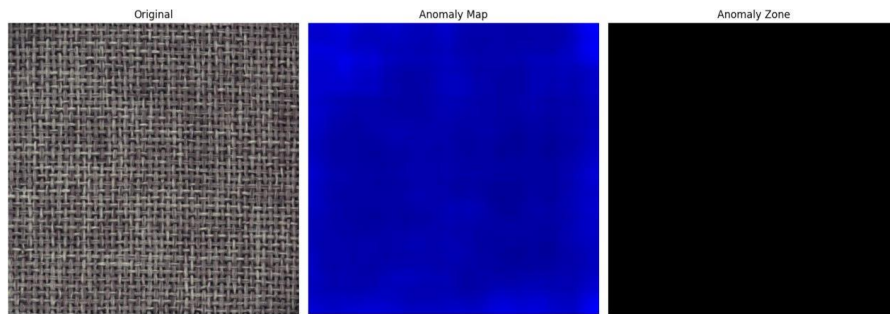


Рисунок В.3 – Результат виявлення аномалії. Тест 2

```
[INFO] Time elapsed: 1.346 seconds
[INFO] Memory usage: 1042.85 MB

Process finished with exit code 0
```

Рисунок В.4 – Результат виконання тесту 2

**Висновок:** у результаті виконання Тесту 2 функціональна вимога щодо виявлення аномалії виконана. А також було виміряно час виконання обробки одного зображення та використання пам'яті, що не перевищує вимоги.

### Методика проведення Тесту 3:

- Запуск тестування моделі здійснюється у середовищі PyCharm.
- Вибирається тестове одне зображення з вказаної директорії.
- Програма виконує:
  - завантаження моделі PyTorch;
  - попередню обробку зображення;
  - виконання передбачення;
  - вимір часу виконання та обсягу використаної пам'яті;
  - виведення зображення із виділеною зоною аномалії;

### Очікуваний результат:

- Виведення зображення із виділеною зоною аномалії

- Вивід часу виконання та спожитої пам'яті

### Фактичний результат:

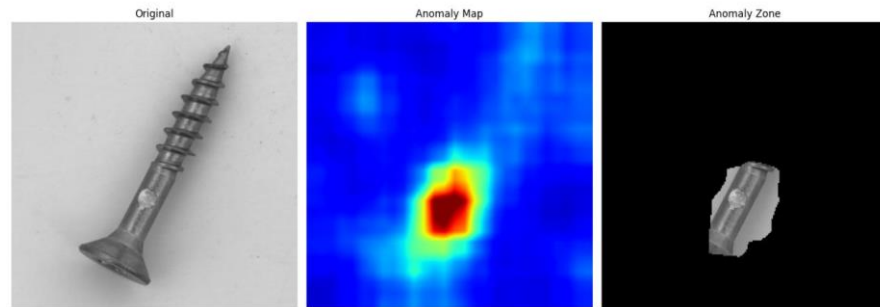


Рисунок В.3 – Результат виявлення аномалії. Тест 3

```
[INFO] Time elapsed: 1.343 seconds  
[INFO] Memory usage: 1042.48 MB  
  
Process finished with exit code 0
```

Рисунок В.6 – Результат виконання тесту 3

**Висновок:** у результаті виконання Тесту 3 функціональна вимога щодо виявлення аномалії виконана. А також було виміряно час виконання обробки одного зображення та використання пам'яті, що не перевищує вимоги.

### Висновок за результатами проведених випробувань

Програмний виріб вважається таким, що пройшов випробування в цілому, якщо успішно виконані всі Тести 1, 2, 3. Загалом, тестування показало, що модель працює ефективно та надає точні результати виявлення аномалії. Це дозволяє зробити висновок про її готовність до використання в реальних умовах або для подальшого доопрацювання. Рекомендується проводити подальше тестування для оцінки роботи моделі на нових даних та вдосконалення результатів.

Виконавець: Буштець Кирило Віталійович



### Фрагмент лістингу коду моделі виявлення аномалій

```

class resnet_feature_extractor(nn.Module):
    def __init__(self):
        super(resnet_feature_extractor, self).__init__()
        self.model = resnet50(weights=ResNet50_Weights.DEFAULT)

        self.model.eval()
        for param in self.model.parameters():
            param.requires_grad = False

        def hook(module, input, output) -> None:
            self.features.append(output)

        self.model.layer2[-1].register_forward_hook(hook)
        self.model.layer3[-1].register_forward_hook(hook)

    def forward(self, input):
        self.features = []
        with torch.no_grad():
            _ = self.model(input)

        self.avg = nn.AvgPool2d(3, stride=1)
        fmap_size = self.features[0].shape[-2]
        self.resize = nn.AdaptiveAvgPool2d(fmap_size)

        resized_maps = [self.resize(self.avg(fmap)) for fmap in
self.features]
        patch = torch.cat(resized_maps, 1)

        return patch

class FeatCAE(nn.Module):
    def __init__(self, in_channels = channel, latent_dim = ld,
is_bn=True):
        super(FeatCAE, self).__init__()

        layers = []
        layers += [nn.Conv2d(in_channels, (in_channels + 2 *
latent_dim) // 2, kernel_size=1, stride=1, padding=0)]
        if is_bn:
            layers += [nn.BatchNorm2d(num_features=(in_channels
+ 2 * latent_dim) // 2)]
        layers += [nn.ReLU()]
        layers += [nn.Conv2d((in_channels + 2 * latent_dim) //
2, 2 * latent_dim, kernel_size=1, stride=1, padding=0)]
        if is_bn:
            layers += [nn.BatchNorm2d(num_features=2 *
latent_dim)]
        layers += [nn.ReLU()]

```

```

        layers += [nn.Conv2d(2 * latent_dim, latent_dim,
kernel_size=1, stride=1, padding=0)]

        self.encoder = nn.Sequential(*layers)

        layers = []
        layers += [nn.Conv2d(latent_dim, 2 * latent_dim,
kernel_size=1, stride=1, padding=0)]
        if is_bn:
            layers += [nn.BatchNorm2d(num_features=2 *
latent_dim)]
        layers += [nn.ReLU()]
        layers += [nn.Conv2d(2 * latent_dim, (in_channels + 2 *
latent_dim) // 2, kernel_size=1, stride=1, padding=0)]
        if is_bn:
            layers += [nn.BatchNorm2d(num_features=(in_channels
+ 2 * latent_dim) // 2)]
        layers += [nn.ReLU()]
        layers += [nn.Conv2d((in_channels + 2 * latent_dim) //
2, in_channels, kernel_size=1, stride=1, padding=0)]
        self.decoder = nn.Sequential(*layers)

    def forward(self, x):
        x = self.encoder(x)
        x = self.decoder(x)
        return x

```