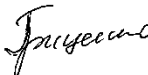


Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

«17» січня 2024 р.

 Гриценко Є. М.

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКТ
 Кафедра _____ КІТАР
 Рівень вищої освіти _____ другий (магістерський)
 Спеціальність _____ 151 Автоматизація та комп'ютерно-інтегровані технології
 Тип програми _____ Освітньо-професійна
 Освітня програма _____ Комп'ютеризовані та робототехнічні системи
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«03» листопада 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Гриценку Євгенію Миколайовичу
 (прізвище, ім'я, по батькові)

1. Тема роботи Розроблення системи контролю виробів 3D друку в роботизованому виробництві
 Затверджена наказом по університету від 03.11.2023 № 1288Ст
2. Термін подання студентом роботи до екзаменаційної комісії 22.01.2024
3. Вихідні дані до роботи _____
 - 3.1 Напруга живлення – 24 В;
 - 3.2 Метод системи контролю використаний у роботі – система контролю на основі нейронної мережі;
 - 3.3 Модель нейронної мережі яка була застосована у роботі – YOLOv5;
 - 3.4 Середовище для навчання моделі – Google Colab;
 - 3.5 Кількість зображень для навчання моделі – 3400 шт.
4. Перелік питань, що потрібно опрацювати в роботі _____
 - 4.1 Вступ;
 - 4.2 Аналіз технічного завдання;
 - 4.3 Розроблення автоматизованої системи контролю якості 3D друку;
 - 4.4 Навчання моделі нейронної мережі та її перевірка;
 - 4.5 Питання пов'язані з охороною праці;
 - 4.6 Висновки та перелік джерел посилань.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій Демонстраційний матеріал, представлений у форматі презентації PowerPoint (*.pptx). 14 – с. Формату А4.

6. Консультанти розділів роботи

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

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Термін виконання етапів роботи | Примітка |
|----|--|--------------------------------|----------|
| 1 | Аналіз технічного завдання | 06.11.2023 | Виконано |
| 2 | Аналіз методу контролю з застосуванням нейронних мереж | 13.11.2023 | Виконано |
| 3 | Вибір існуючого методу для застосування | 20.11.2023 | Виконано |
| 4 | Вибір обладнання для розробки макету | 27.11.2023 | Виконано |
| 5 | Розробка структурної схеми | 06.12.2023 | Виконано |
| 6 | Розробка алгоритму роботи програми | 15.12.2023 | Виконано |
| 7 | Вибір мови програмування та розробка програмного засобу для СК | 19.12.2023 | Виконано |
| 8 | Навчання моделі нейронної мережі та її перевірка | 04.01.2024 | Виконано |
| 9 | Оформлення пояснювальної записки | 16.01.2024 | Виконано |
| 10 | Подання кваліфікаційної роботи в ЕК | 22.01.2024 | Виконано |

Дата видачі завдання 03.11.2023 р.

Студент _____
(підпис)

Гриценко Є. М.
(прізвище, ініціали)

Керівник роботи _____
(підпис)

проф. Олександров Ю.М.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить: 84 с., 3 табл., 43 рис., 3 дод., 25 джерел.

3D ДРУК, 3D ПРИНТЕР, СИСТЕМА КОНТРОЛЮ, ЗГОРТКОВА
НЕЙРОННА МЕРЕЖА, MARLIN, YOLO.

Об'єкт дослідження – методи машинного навчання роботизованих систем на виробництві з прототипування.

Предмет дослідження – контроль якості друку 3D деталей з використанням нейронної мережі.

Мета роботи – удосконалення технології FDM 3D друку шляхом розроблення системи контролю 3D друкованих деталей у реальному часі для підвищення якості друку.

Методи дослідження – математичний аналіз, аналіз даних, машинне навчання з попереднім використанням тренувальної множини.

Наукова новизна полягає у розробці системи контролю для будь-якого 3D принтера, який має прошивку Marlin для розпізнавання дефектів у реальному час.

Практична цінність полягає у зменшенні витрат та часу на браковані деталі, які так чи інакше виникають під час 3D друку.

Отримані результати розробки можна застосовувати у комп'ютерно-інтегрованих системах на масштабній фермі 3D друку у якості системи контролю, що допоможе контролювати процес і мінімізувати затрати від браку.

THE ABSTRACT

Explanatory note: 84 pp., 3 tab., 43 fig., 3 app., 25 sources.

3D PRINTING, 3D PRINTER, CONTROL SYSTEM, CONVOLUTIONAL NEURAL NETWORK, MARLIN, YOLO.

The object of research – is machine learning methods for robotic systems in prototyping production.

The subject of research – is the quality control of 3D printed parts using a neural network.

Purpose - to improve the FDM 3D printing technology by developing a real-time control system for 3D printed parts to improve print quality.

Research methods – mathematical analysis, data analysis, machine learning with preliminary use of the training set.

The scientific novelty is the development of a control system for any 3D printer that has Marlin firmware for real-time defect recognition.

The practical value is to reduce the cost and time for defective parts that somehow arise during 3D printing.

The results of the development can be used in computer-integrated systems on a large-scale 3D printing farm as a control system that will help control the process and minimize the cost of defects.

ЗМІСТ

| | |
|--|----|
| Перелік умовних скорочень | 9 |
| Вступ..... | 10 |
| 1 Аналіз літератури за темою дослідження..... | 13 |
| 1.1 Аналіз структури систем контролю | 13 |
| 1.2 Аналіз СК з використанням машинного зору | 15 |
| 1.3 Аналіз методу контролю з застосуванням нейронних мереж | 20 |
| 1.4 Вибір існуючого методу для застосування в проекті | 22 |
| 1.5 СК дефектів друку 3D принтерів..... | 25 |
| 1.6 Постановка завдання на розробку | 27 |
| 1.7 Висновки до першого розділу | 27 |
| 2 Розроблення автоматизованої системи контролю якості 3D друку..... | 29 |
| 2.1 Вибір обладнання для розробки макету | 29 |
| 2.2 Розробка структурної схеми макету | 32 |
| 2.3 Розробка системи контролю деталей за допомогою штучної нейронної мережі..... | 34 |
| 2.3.1 Обґрунтування використання нейронних мереж | 34 |
| 2.3.2 Вибір моделі нейронної мережі..... | 36 |
| 2.4 Алгоритм розпізнавання дефектів..... | 40 |
| 2.5 Розробка підпрограми для зупинки 3D друку..... | 44 |
| 2.6 Висновки до другого розділу..... | 47 |
| 3 Навчання моделі нейронної мережі та проведення експерименту | 48 |
| 3.1 Вибір мови програмування та розробка програмного засобу для СК..... | 48 |
| 3.2 Підготовка набору даних для навчання..... | 52 |
| 3.3 Навчання моделі нейронної мережі | 55 |
| 3.4 Постановка експерименту | 63 |

| | |
|--|-----|
| | 8 |
| 3.4.1 Перевірка деталі з дефектом..... | 63 |
| 3.4.2 Перевірка деталі з відсутністю дефектів..... | 68 |
| 3.5 Охорона праці..... | 72 |
| 3.6 Висновки до третього розділу | 76 |
| Висновки | 78 |
| Перелік посилань..... | 80 |
| Додаток А Апробація результатів наукових досліджень..... | 84 |
| Додаток Б Лістинг програми..... | 103 |
| Додаток В Демонстраційний матеріал..... | 108 |

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ЗНМ – згорткові нейронні мережі;

СК – системи контролю;

FDM – моделювання методом пошарового наплавлення;

HOG – гістограми орієнтаційних градієнтів;

ORB – Oriented FAST and Rotated BRIEF;

PLA (Polylactic acid) – полілактидна кислота;

SIFT – масштабоінваріантне ознакове перетворення;

SURF – прискорені стійкі ознаки;

YOLO – You Only Look Once.

ВСТУП

На сьогоднішній день зростає кількість роботизованих підприємств. Разом з цим з'являється проблема з контролем продукції на виробництві.

Дана тема є актуальною сьогодні з кількох важливих причин. По-перше, зростання використання роботів у виробництві створює підвищену потребу в ефективних системах контролю, які забезпечують високу якість виробництва. По-друге, з введенням інтернету речей та обробки даних у реальному часі стало можливим збирати та аналізувати великі обсяги інформації, що дозволяє розробити більш точні та надійні системи контролю. По-третє, питання якості продукції стають все важливішими для підприємств у конкурентному світі, де вимагається висока точність та надійність. На додачу до цього, розвиток штучного інтелекту та машинного навчання дозволяє розширити можливості систем контролю та забезпечити їх адаптацію до змінних умов виробництва. Усі ці фактори роблять розробку систем контролю деталей на потоковому роботизованому виробництві актуальною та важливою задачею сьогодні.

У даній роботі було розроблено систему контролю 3D друкованих деталей. Дана тема є актуальною на сьогодні, адже велика кількість деталей друкуються на 3D принтерах і не тільки пластиком, адже існують кондитерські 3D принтери, принтери, що використовуються на будівництві тощо. Тому постає питання розробки системи контролю для даних пристроїв.

У наш час на багатьох 3D принтерах застосовується прошивка Marlin для керування даним пристроєм. Це доволі проста прошивка, яка має основні, потрібні звичайному користувачу, налаштування. Проте дана прошивка, не дивлячись на те, що вона простіша для звичайного користувача у порівнянні з новою Klipper, має ряд недоліків, одним з яких і є система контролю.

У даній роботі буде описано процес розробки такої система, що буде використовувати модель навченої нейронної мережі для контролю процесу друку.

Дана робота повинна вирішувати наступні питання:

- при виробництві через 3D друк можуть виникати різні дефекти, такі як неправильне з'єднання шарів, викиди матеріалу, деформації тощо. Виявлення цих дефектів у процесі друку дозволяє уникнути виробничих недоліків та покращити якість готового продукту;
- виявлення дефектів під час друку дозволяє реагувати на них у реальному часі. Це може зменшити кількість відходів матеріалу, часу та зусиль, необхідних для виробництва якісних виробів;
- шляхом виявлення та аналізу дефектів можна покращувати налаштування друкарського обладнання та матеріалів для досягнення кращих результатів;
- уникнення дефектів у процесі друку призводить до зменшення витрат на виготовлення та до підвищення якості продукції, що може бути важливим для підприємств та виробників.

Розробка програми для виявлення дефектів при 3D друці на принтерах з прошивкою Marlin може допомогти виробникам автоматизувати процес контролю якості, забезпечити більш точні та ефективні результати та покращити загальний процес виробництва.

Мета роботи – удосконалення технології FDM 3D друку шляхом розроблення системи контролю 3D друкованих деталей у реальному часі для підвищення якості друку.

Об'єкт дослідження – методи машинного навчання роботизованих систем на виробництві з прототипування.

Предмет дослідження – контроль якості друку 3D деталей з використанням нейронної мережі.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз існуючих систем контролю;
- розробити структурну схему макету;
- провести підбір обладнання для макету;
- розробити алгоритм роботи програми для розпізнавання;
- створити власний набір даних для навчання моделі нейронної мережі;
- навчити модель нейронної мережі;
- перевірити систему в різних умовах;
- оформити кваліфікаційну роботу згідно ДСТУ 3008:2015 [1], а також з методичними вказівками з підготовки й оформлення кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології освітньої програми «Комп'ютеризовані та робототехнічні системи» [2].

Результати наукових досліджень були опубліковані у збірнику наукових статей [4] та наведені у додатку Додаток А.

1 АНАЛІЗ ЛІТЕРАТУРИ ЗА ТЕМОЮ ДОСЛІДЖЕННЯ

1.1 Аналіз структури систем контролю

Системи контролю (СК) – це комплексне об'єднання обладнання, програмного забезпечення та процесів, створене для вимірювання, моніторингу, регулювання та керування певними аспектами деякого системного об'єкта або процесу. Ці системи можуть бути застосовані в різних галузях, таких як виробництво, транспорт, автоматизація, енергетика, медицина, телекомунікації та багато інших [3].

Основні функції СК включають в себе:

- збір інформації. Вони вимірюють і збирають дані про стан системи або процесу за допомогою датчиків, сенсорів та інших пристроїв;
- аналіз і обробка інформації. СК обробляють зібрані дані, використовуючи різні алгоритми та методи аналізу, щоб зрозуміти стан системи і виявити аномалії;
- прийняття рішень. На основі аналізу, СК можуть приймати рішення, такі як включення або вимкнення пристроїв, зміна параметрів роботи або відправлення сповіщень операторам;
- керування системою. Вони можуть використовувати зібрані дані та прийняті рішення для автоматичного керування системою або процесом, забезпечуючи оптимальний режим роботи;
- моніторинг і звітність. СК можуть надавати інформацію операторам або керівництву для моніторингу та аналізу продуктивності та ефективності;
- автоматизація та оптимізація процесів. СК можуть сприяти автоматизації процесів, зниженню витрат, підвищенню якості та безпеки виробництва.

СК деталей на виробництві відіграють важливу роль у забезпеченні якості продукції та відповідності стандартам. Існують різні види таких систем, кожний з яких має свої переваги та недоліки.

Ось деякі з них:

- візуальний контроль. Використовується для перевірки візуальних параметрів, таких як розмір, форма, колір і текстура деталей. Перевагами цього методу є низька вартість впровадження і здатність виявляти видимі дефекти. Однак він обмежується об'єктивністю спостерігача та не завжди ефективний для незримих дефектів;

- вимірювальні СК. Використовуються для точного вимірювання розмірів та геометричних параметрів деталей. Вони дозволяють виявити мікроскопічні дефекти, але можуть бути дорогими та вимагати точності при налаштуванні;

- СК за допомогою рентгенівських променів. Використовують рентгенівські промені для проникнення в матеріал і виявлення дефектів, які не видимі на поверхні. Це ефективний метод для виявлення внутрішніх дефектів, але може бути дорогим і вимагає використання радіаційної технології;

- ультразвукові СК. Використовують ультразвук для виявлення дефектів, особливо в металевих деталях. Цей метод ефективний для виявлення тріщин та інших дефектів, але вимагає навчання операторів та налаштування обладнання;

- системи машинного зору та штучного інтелекту. Використовують комп'ютерне зорове сприйняття та алгоритми машинного навчання для автоматичного виявлення дефектів. Ці системи можуть бути швидкими та надійними, але вимагають великої кількості даних для навчання.

Перевагами СК деталей є підвищення якості продукції, зменшення відходів та зниження витрат на перевірку якості. Однак у них можуть бути недоліки, такі як високі витрати на впровадження, потреба в кваліфікованих операторах, та обмеження в ефективності виявлення дефектів, які не завжди видимі або піддаються легкому вимірюванню.

В даній роботі буде розглянуто СК за допомогою машинного зору.

1.2 Аналіз СК з використанням машинного зору

СК за допомогою машинного зору (Machine Vision) використовують комп'ютерне зорове сприйняття для автоматичного аналізу та інтерпретації зображень або відео для виявлення дефектів, вимірювання розмірів, розпізнавання об'єктів та багато інших завдань. Ця технологія дозволяє системам виробництва та автоматизації оптимізувати процеси та підвищити якість продукції. Ось докладніша інформація про СК за допомогою машинного зору:

- засоби візуального сприйняття. Системи машинного зору використовують камери і сенсори для збору візуальних даних. Це можуть бути звичайні кольорові камери, термальні камери, камери з високою швидкістю тощо;

- аналіз зображення. При зборі зображень, система обробляє їх для виділення основних особливостей і аналізу. Це включає в себе виявлення контурів, кольорів, текстур, геометричних параметрів і багато іншого;

- виявлення дефектів. Однією з основних функцій систем машинного зору є виявлення дефектів виробів або матеріалів. Вони можуть виявляти подряпини, тріщини, бульбашки повітря, незбалансованість тощо;

- вимірювання і розпізнавання об'єктів. Системи можуть вимірювати розміри об'єктів, визначати їх форму, розпізнавати маркування та інші характеристики;

- підсумковий аналіз. Після обробки зображень системи можуть зробити рішення щодо допустимості виробу, дати сигнали для роботів або інших систем керування, або створити звіти для операторів або керівництва;

– інтеграція з іншими системами. СК за допомогою машинного зору часто інтегруються з іншими системами керування та виробництва для автоматичного реагування на знайдені дефекти або для управління робочими процесами.

Комп'ютерний зір може використовуватись у різних сферах та для розв'язання різноманітних задач. Однією з таких задач є ідентифікація об'єктів.

Після того як об'єкт буде ідентифіковано, з ним будуть проводитись інші дії які дозволяє ця технологія, наприклад:

- розпізнавання кольору;
- розпізнавання форми;
- розпізнавання меж об'єкту;
- розпізнавання координат об'єкту.

Дані системи дуже часто можуть бути використані в якості контролю на виробництві. Це дає змогу зменшити таку критичну виробничу проблему як людський фактор. Також це дає змогу зменшити вплив людини на контроль продукції та використовувати її як оператора, а не працівника [4].

До таких систем можна віднести систему виявлення дефектних деталей на виробничій лінії. Дана система дасть змогу підвищити темп виробництва та зменшити кількість випуску бракованої продукції.

Для вирішення задачі розпізнавання дефектів на деталях можуть застосовуватись різні методи [5].

Ось деякі з них:

– метод шаблону (Template Matching). Template Matching дозволяє виявляти, де на вихідному зображенні знаходиться вказаний шаблон (рис. 1.1). Він працює порівнюючи піксель за пікселем величину шаблону з вихідним зображенням та вказує на області, де збігаються значення;

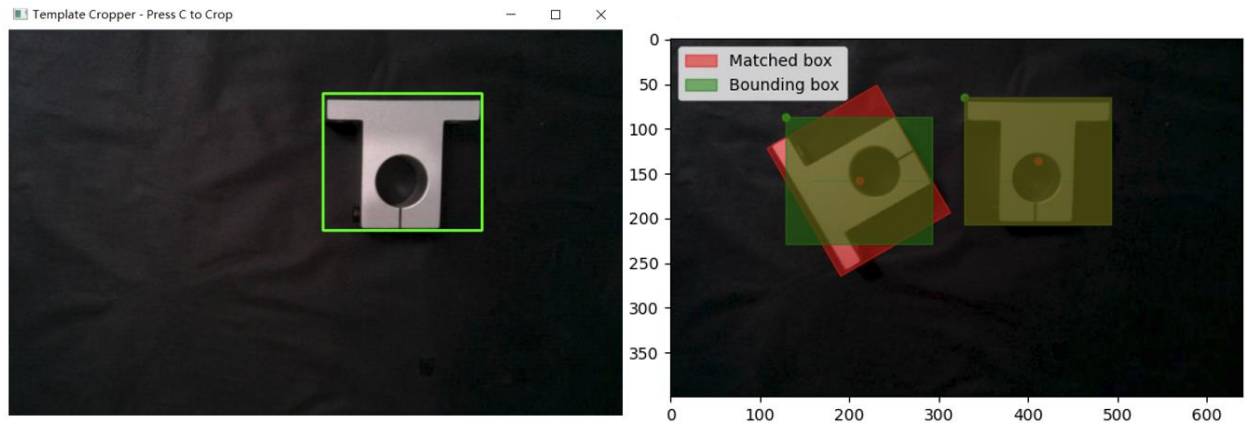


Рисунок 1.1 – Метод шаблону [6]

– Scale-Invariant Feature Transform (SIFT). SIFT – це алгоритм, який визначає ключові точки на зображенні (рис. 1.2). Кожна ключова точка має свій дескриптор, який описує оточуючі області. Цей метод дозволяє виявляти деталі незалежно від масштабу та орієнтації [8];

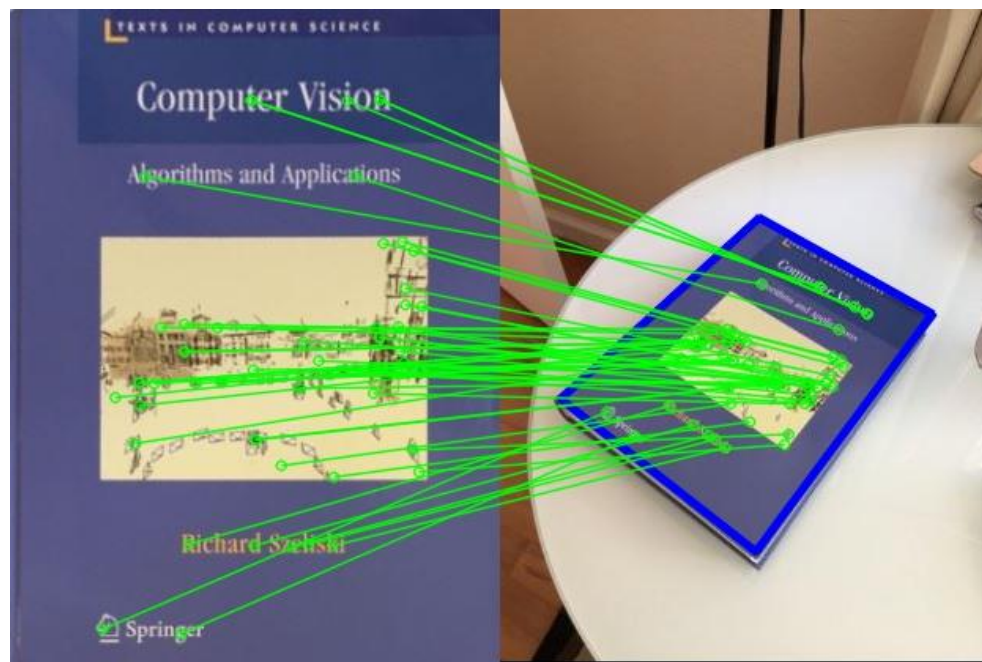


Рисунок 1.2 – Метод SIFT [7]

– Speeded-Up Robust Features (SURF). SURF подібний до SIFT, але швидший і менше обчислювально вимогливий. Він також визначає ключові точки та дескриптори для розпізнавання деталей на зображенні [8];

– Oriented FAST and Rotated BRIEF (ORB). ORB – це ще один метод для визначення ключових точок та дескрипторів (рис. 1.3). Він є швидким і підходить для розпізнавання деталей на зображеннях;

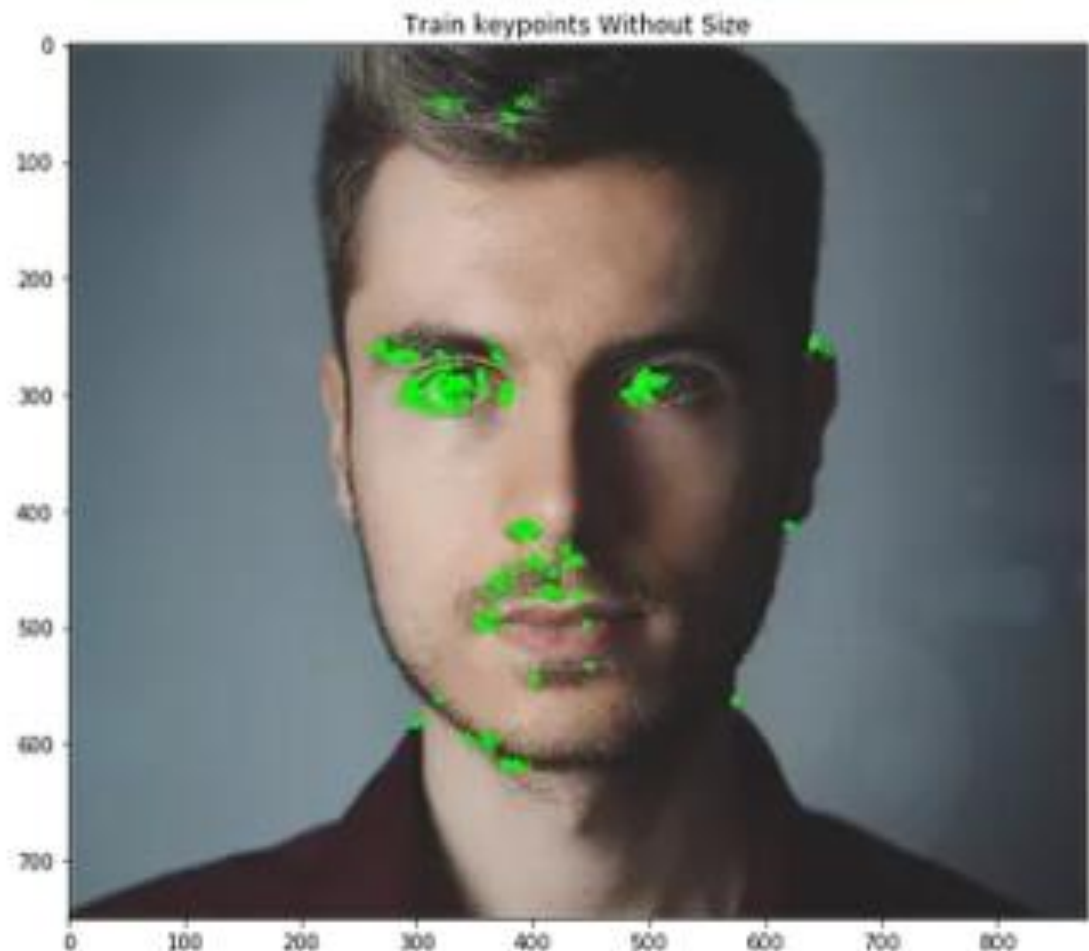


Рисунок 1.3 – Метод ORB [9]

– каскади Хаара (Haar Cascades). Каскади Хаара використовуються для детектування об'єктів або облич на зображеннях (рис. 1.4).

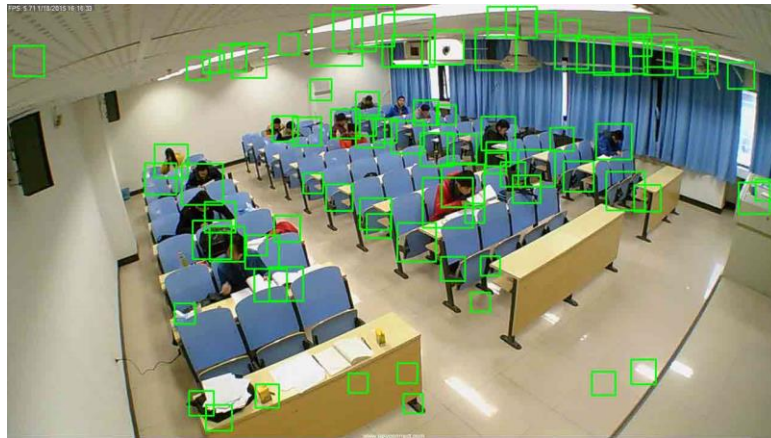


Рисунок 1.4 – Метод каскадів Хаара [10]

Вони працюють на основі признаков Хаара, які вказують на певні структурні елементи зображення;

- детектор кутів Харріса (Harris Corner Detector). Детектор кутів Харріса виявляє кути на зображенні, які можуть бути використані для визначення областей зі змінним освітленням або граней об'єктів [10];

- гістограми орієнтаційних градієнтів (HOG). Гістограми орієнтаційних градієнтів використовуються для аналізу текстур та форм об'єктів на зображеннях (рис. 1.5) [11].

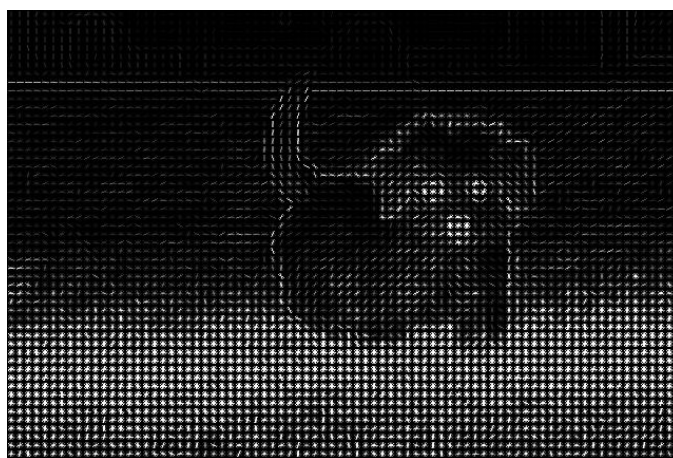


Рисунок 1.5 – Метод HOG

Вони розглядають градієнти пікселів для визначення орієнтації та текстури;
– машинне навчання і нейромережі (рис. 1.6).

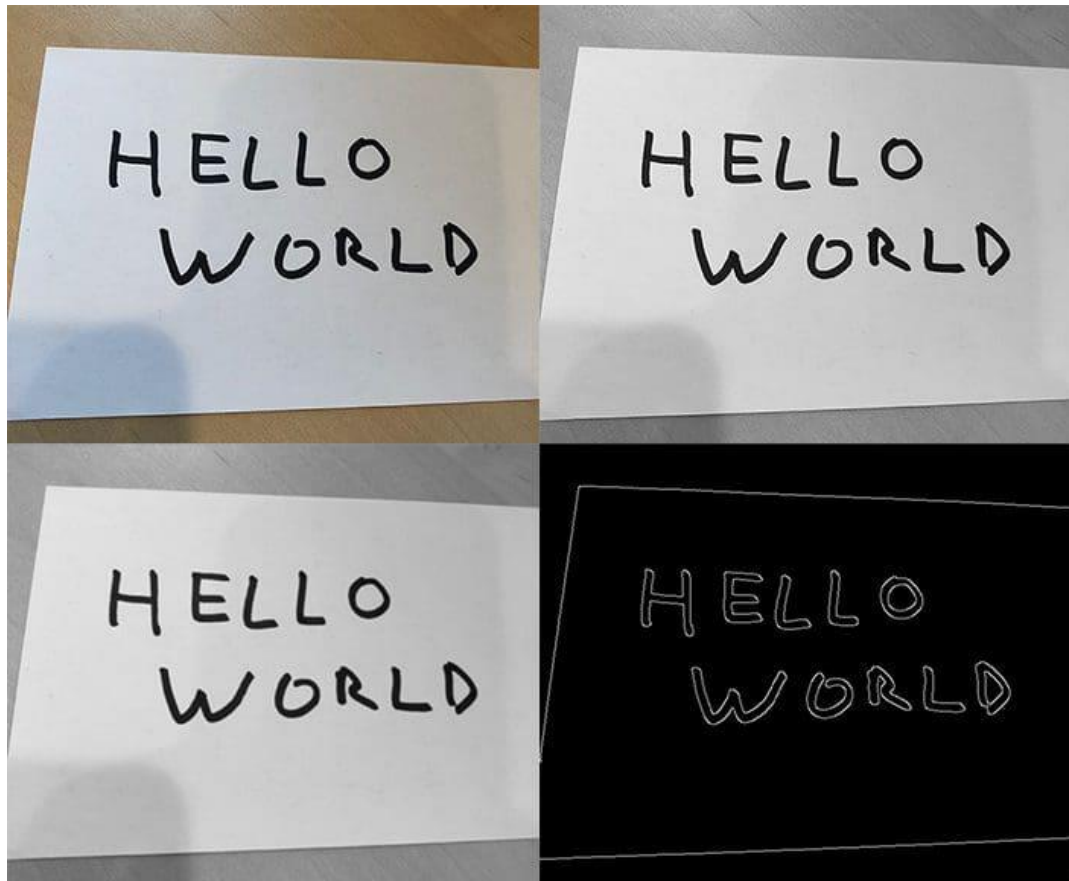


Рисунок 1.6 – Метод з застосуванням нейронних мереж

1.3 Аналіз методу контролю з застосуванням нейронних мереж

СК якості та виявлення браку виробництва з використанням нейронних мереж стали дуже популярними завдяки своїй здатності автоматизувати процеси виробництва та виявляти дефекти з високою точністю. Нейронні мережі використовуються для аналізу відео, зображень, акустичних сигналів та інших типів даних для виявлення аномалій і браку в продукції. СК якості виробництва

використовуються для виявлення дефектів та аномалій в продукції з метою забезпечення високої якості та відповідності стандартам.

Нейронні мережі є ключовим елементом сучасних СК якості. Вони використовуються для обробки даних та виявлення браку.

Для того, щоб навчити нейронну мережу розпізнавати дефекти в першу чергу необхідно підготувати дані для її навчання. Це може включати в себе збір відео або зображень продукції, які підлягають аналізу.

Архітектура нейронної мережі для аналізу зображень – це, зазвичай, глибока згортоква нейронна мережа (ЗНМ). ЗНМ використовують різновид багат шарових перцептронів, розроблений так, щоби вимагати використання мінімального обсягу попередньої обробки.

Вони відомі також як інваріантні відносно зсуву або просторово інваріантні штучні нейронні мережі, виходячи з їхньої архітектури спільних ваг та характеристик інваріантності відносно паралельного перенесення [12].

Згорткові мережі взяли за основу біологічний процес, а саме схему з'єднання нейронів зорової кори тварин. Окремі нейрони кори реагують на стимули лише в обмеженій області зорового поля, відомій як рецептивне поле. Рецептивні поля різних нейронів частково перекриваються таким чином, що вони покривають усе зорове поле [13].

ЗНМ використовують відносно мало попередньої обробки, в порівнянні з іншими алгоритмами класифікації зображень. Це означає, що мережа навчається фільтрів, що в традиційних алгоритмах конструювали вручну. Ця незалежність у конструюванні ознак від апріорних знань та людських зусиль є великою перевагою.

Вони мають застосування в розпізнаванні зображень та відео, рекомендаційних системах та обробці природної мови [14-15].

1.4 Вибір існуючого методу для застосуванні в проекті

Для вирішення задачі розпізнавання дефектів на об'єктах необхідно чітко визначити алгоритм дій.

Попередній алгоритм може виглядати так:

- фільтрація зображення. Необхідно застосувати фільтри такі як розмиття або фільтр Гауса, для зниження шуму і поліпшення якості зображення. Це допоможе усунути небажані деталі та поліпшити контрастність об'єктів;
- порогова обробка. Необхідно використати алгоритм порогової обробки для відділення об'єктів від фону на основі яскравості або кольору пікселів. Ці параметри необхідно підібрати;
- виділення контурів. Застосувати алгоритм виділення контурів (алгоритм Кенні), для виявлення меж об'єкту на зображенні. Це дозволить проаналізувати форму об'єкта;
- використати HOG-дескриптори або нейронні мережі.

Після того як був складений алгоритм можна перейти до розбору методів за допомогою яких можна визначити дефекти на об'єктах.

Для розпізнавання об'єкта можна використати метод каскадів (признаків) Хаара. Каскади Хаара є алгоритмом комп'ютерного зору, який зазвичай використовується для виявлення і розпізнавання об'єктів на зображеннях. Він заснований на виділенні характерних особливостей (таких як межі, кути або текстури) і використанні їх для класифікації об'єктів. Даний метод підходить для розпізнавання дефектів, але точність буде залежати від наступних факторів:

- якість зображення. Для досягнення високої точності детектування дефектів, важливо мати зображення з високою роздільною здатністю і гарною чіткістю. Якщо зображення низької якості або має артефакти, це може знизити точність визначення дефектів. А так як всі дії будуть проводитись у режимі реального часу, то з цього витікає ще один мінус – вартість обладнання;

- тренування класифікатора. Каскад Хаара вимагає попереднього тренування з використанням великого набору розмічених зображень, що містять як позитивні приклади дефектів, так і негативні приклади без дефектів. Якість тренування і розмір навчального набору можуть істотно впливати на точність визначення;

- тип дефектів. Каскад Хаара найкраще працює для виявлення об'єктів з яскраво вираженими текстурними або геометричними особливостями. Якщо дефекти на деталях мають слабо виражені або неоднорідні характеристики, точність визначення може бути знижена;

- параметри детектора. Налаштування параметрів каскаду Хаара, як-от розмір вікна сканування, порогові значення і число рівнів детектування, може вплинути на точність і швидкість виявлення дефектів. Необхідно провести експерименти й оптимізувати ці параметри для конкретного завдання.

Також слід додати, що зона (місце) де виконується детектування повинно бути гарно освітленим. В противному випадку тіні на об'єкті можуть розцінюватись як дефекти, що не дасть ніякої переваги перед неавтоматизованими СК.

Згідно з описом даного методу – його можна використати в якості контролю форми деталі.

Також як зазначалося вище для детектування дефектів можна використати нейронні мережі. Цей метод буде більш точним. Для цього необхідно навчити нейронну мережу. Цей процес доволі довготривалий і це є одним з мінусів їх використання. Використання нейронних мереж для виявлення дефектів на об'єктах має свої плюси та мінуси.

Плюси використання нейронних мереж:

- автоматизація. Нейронні мережі можуть автоматизувати процес виявлення дефектів, що дозволяє швидше і ефективніше проводити контроль якості виробництва;

- здатність до узагальнення. Нейронні мережі можуть навчатися розпізнавати широкий спектр дефектів, включаючи ті, які можуть бути важко програмно визначити;

- адаптація до змін. Мережі можуть бути налаштовані для розпізнавання нових видів дефектів без необхідності перепрограмувати систему з нуля;

- можливість роботи з великими обсягами даних. Нейронні мережі можуть швидко обробляти великі обсяги даних, що важливо для виробничих процесів з високими темпами виробництва.

Мінусами застосування нейронних мереж є:

- потреба в великому обсязі даних. Навчання нейронних мереж вимагає значної кількості даних з достатньою кількістю прикладів дефектів. Якщо відповідних даних недостатньо, модель може бути недоученою;

- складність налаштування. Вибір правильної архітектури та гіперпараметрів для нейронної мережі може бути складним завданням і вимагає фахових знань;

- інтерпретованість. Великі нейронні мережі часто важко інтерпретувати, що ускладнює зрозуміння причини рішень, прийнятих моделлю;

- витрати на обчислення. Тренування та робота нейронних мереж може вимагати значних обчислювальних ресурсів і електроенергії;

- залежність від якості даних. Якість навчальних даних безпосередньо впливає на якість моделі. Якщо дані містять помилки або шум, модель може навчитися неправильно;

- не завжди підходить для всіх дефектів. Деякі дефекти, зокрема ті, які пов'язані із структурними особливостями матеріалу або об'єкта, можуть бути важко виявити за допомогою нейронних мереж.

Загалом стає зрозуміло, що для більш точного виявлення дефектів краще використати нейронні мережі в якості детектора, або розробити алгоритм при якому два методи будуть працювати одночасно.

1.5 СК дефектів друку 3D принтерів

Як вже стало відомо автоматизовані СК у наш час дуже розповсюджені і можуть використовуватись як на підприємствах так і у звичайних побутових речах. Системи візуального контролю а тому числі. Але системи візуального контролю якості друку 3D принтерів не настільки розповсюджені на цей час, як і сам друк на 3D принтері.

З відомих рішень які застосовуються можна виділити модуль який є у прошивці Klipper.

Klipper – це програмне забезпечення з відкритим кодом, яке використовується для керування 3D принтерами та іншими обладнаннями для виготовлення. Він був розроблений спільнотою і доступний на безкоштовній основі. Він зазвичай встановлюється на мікроконтролерах, які керують рухом принтера та іншими процесами, пов'язаними з 3D друку.

Основна ідея Klipper полягає в тому, що весь обробник команд для 3D принтера виконується на більш потужному комп'ютері (наприклад, на Raspberry Pi), і команди передаються на мікроконтролер принтера через зручний протокол. Це дозволяє підвищити точність та швидкість друку, оскільки мікроконтролер може обробляти команди руху більш швидко, а також робити інші корисні функції.

Klipper також підтримує багато типів 3D принтерів і інших CNC-машин, і його функціональність може бути розширена за допомогою різних плагінів. Він є популярним вибором серед ентузіастів 3D друку, оскільки надає більше можливостей для налаштування та покращення роботи принтера.

Також Klipper може підтримувати і візуальний контроль якості друку, та в разі чого зупиняти друк, аби не витратити пластик на зіпсовану деталь і не пошкодити обладнання. Зазвичай дану прошивку використовують студії 3D друку, або так звані ферми. Там обладнання може бути дуже багато, а слідкувати

за цим всім процесом дуже складно, адже друк може тривати від кількох годин до кількох днів в залежності від розміру деталі. Приклад дефектів 3D друку зображений на рисунку 1.7.

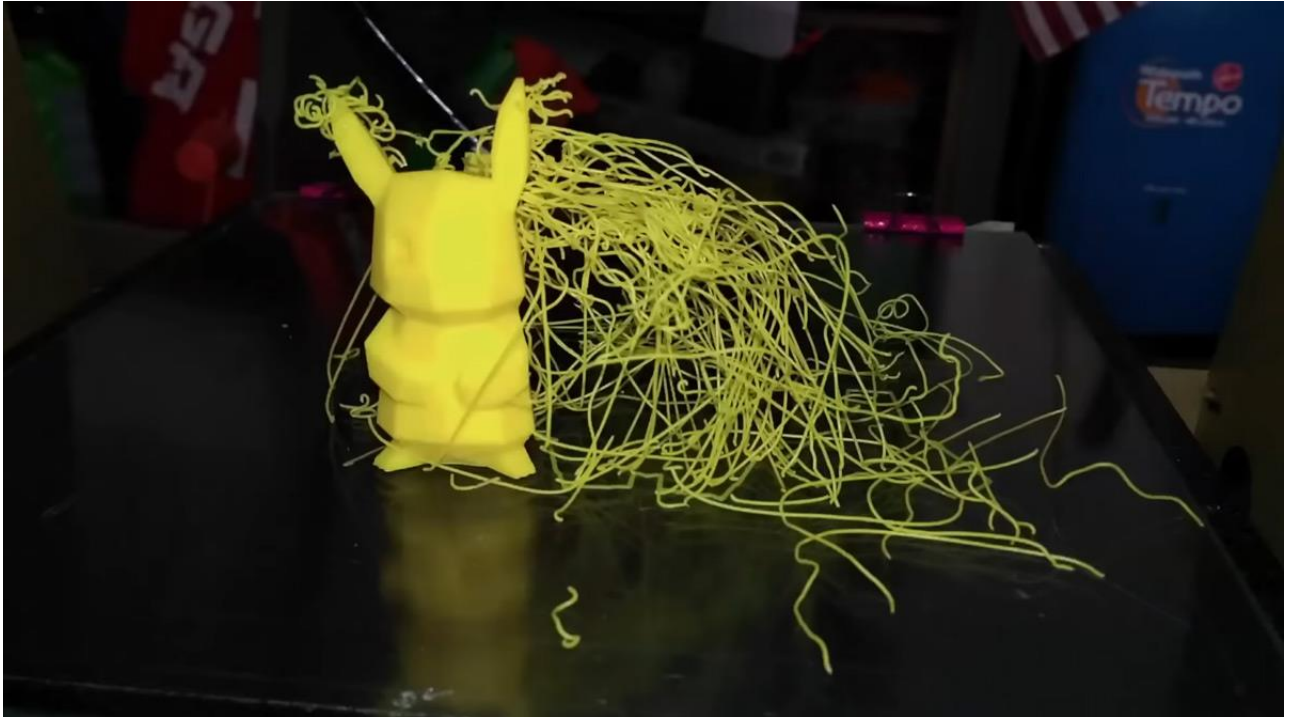


Рисунок 1.7 – Дефекти при друці на 3D принтері

Проблемою даної прошивки є те, що звичайний користувач, який або тільки починає знайомитись з даною технологією, або не дуже досвідчений скоріше за все буде мати складнощі зі встановленням, адже для її роботи необхідно перепрограмувати основний мікроконтролер. Також для роботи з Klipper необхідно мати хост такий як стаціонарний комп'ютер (використовується рідко), або одноплатний комп'ютер такий як Raspberry Pi.

Для прошивки Marlin таких відомих рішень немає. Хоча дана прошивка поставляється майже з усіма 3D принтерами з заводу. Можуть бути звичайно різні рішення від різних компаній, але в основному команди нічим не відрізняються.

Постає завдання розробити програмний модуль для принтерів під прошивку Marlin, який би визначав дефекти при 3D друці.

Найпоширеніші дефекти 3D друку:

- великий зазор між соплом і столом. Наслідком цієї проблеми є те, що деталь буде друкуватися у повітрі і скоріше за все зіпсує перший шар, після чого деталь відвалиться і принтер буде просто витрачати пластик друкуючи його в повітрі;

- недостатній зазор між соплом і столом. Наслідком можуть бути пропуски на першому шарі. При даній проблемі сопло принтера опущено дуже низько і починає друкувати не тією висотою шару яка потрібна розмазуючи видавлений пластик по поверхні столу внаслідок чого може зламатися обладнання.

1.6 Постановка завдання на розробку

Враховуючи те що, метою кваліфікаційної роботи є удосконалення технології FDM 3D друку шляхом розроблення системи контролю 3D друкованих деталей у реальному часі, для досягнення поставленої мети необхідно вирішити наступні завдання:

- розробити структурну схему макету;
- розробити алгоритм визначення дефектів для друкованих деталей;
- підібрати нейронну мережу для використання у макеті;
- розробити підпрограму для зупинки друку;
- підібрати обладнання для роботи макету.

1.7 Висновки до першого розділу

У даному розділі був проведений аналіз технічного завдання та розглянуті деякі існуючі аналоги даної СК.

Після вивчення існуючих аналогів систем контролю виробництва, а також проведення аналізу власного технічного завдання, виявлено декілька ключових аспектів.

Було виявлено, що існують різні системи контролю якості виробництва, які використовують різні методи, такі як комп'ютерний зір, нейронні мережі, аналіз зображень тощо. Вони зазвичай пропонують широкий спектр можливостей, але можуть бути дорогими та складними у впровадженні.

Було виявлено недоліки існуючих систем контролю 3D друку. Однією з таких систем є додаток до нестандартної для багатьох принтерів прошивки Klipper. Також було визначено, що існуючих рішень для прошивки Marlin не існує, тому було вирішено розробити СК саме для цієї прошивки, оскільки вона використовується у переважній більшості пристроїв за замовчуванням.

2 РОЗРОБЛЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ КОНТРОЛЮ ЯКОСТІ 3D ДРУКУ

2.1 Вибір обладнання для розробки макету

Макет СК дефектів при 3D друці буде складатися з трьох модулів – це камера, хост та 3D принтер.

В якості камери для розпізнавання буде використовуватись веб-камера Logitech C922 (рис. 2.1). Короткі характеристики наведені у таблиці 2.1. Камера буде стояти навпроти стола 3D принтера, що дозволить повністю охопити площу друку.

В більшості випадків камера є основною складовою при контролі, тому необхідно дібрати освітлення та ракурс для того, щоб система працювала без збоїв та хибних спрацьовувань.

У даному макеті камера буде стояти напроти принтера, так як конструкція принтера не дозволяє розмістити її всередині корпусу. Камера буде охоплювати всю поверхню столу, що дозволить контролювати максимальну кількість об'єктів, що друкуються.

Використання веб-камери у даному випадку забезпечить високу якість відеозапису в роздільній здатності 1080p. Чіткість та деталізація зображення дозволяють здійснювати більш точний та деталізований аналіз друку для виявлення потенційних дефектів.

Також є функції автофокусу та корекції освітлення, що дозволяють отримувати чітке зображення навіть при зміні освітлення в приміщенні, що є важливим для точного аналізу друку протягом його виконання.



Рисунок 2.1 – Камера для проекту [16]

Таблиця 2.1 – Короткі характеристики камери

| | |
|---------------------------|---|
| Роздільна здатність відео | 1920x1080 |
| Фокусування | Автофокус |
| Живлення | USB |
| Поле огляду | 78° |
| Частота кадрів | 60 кадрів/с при 720p, 30 кадрів/с при 1080p |

У зв'язку з тим, що використання нейромережі для розпізнавання об'єктів потребує обробки великої кількості даних та досить великої обчислювальної потужності, стає необхідним використовувати хост. В якості хоста буде використовуватись звичайний персональний комп'ютер. Короткі характеристики наведені у таблиці 2.2.

Таблиця 2.2 – Короткі характеристики хоста

| | |
|-----|---|
| CPU | i5-11600KF, 6 ядер, 12 віртуальних ядер, базова тактова частота 3,9 ГГц |
| GPU | Geforce GTX 1080TI, обсяг відеопам'яті – 11 Гб, частота пам'яті – 11124 МГц, частота ядра – 1569 МГц, роздільність шини пам'яті – 352 біт |
| RAM | 16 Гб DDR4, частота – 3200 МГц |

Об'єктом для контролю та керування буде виступати 3D принтер FlyingBear Ghost 6 (рис. 2.2), його характеристики наведені у таблиці 2.3.



Рисунок 2.2 – 3D принтер для проекту [17]

Таблиця 2.3 – Характеристики 3D принтера [17]

| 3D принтер | |
|---------------------------------------|-----------------------------|
| Технологія друку | FDM |
| Матеріал друку | PLA, ABS, Wood, HIPS, PETG |
| Кількість друкуючих сопел | 1 шт |
| Діаметр сопел | 0,4 мм |
| Горизонтальна роздільна здатність | 0,01 мм |
| Вертикальна роздільна здатність друку | 0,002 мкм |
| Кінематика | CoreXY |
| Прошивка | Marlin |
| Типова/максимальна швидкість друку | до 150 мм/с |
| Драйвери крокових двигунів | TMC-2225 |
| Екструдер | Direct extruder |
| Термоблок | E3D-V6 |
| Рекомендована температура екструдера | 210 °C (максимальна 260 °C) |
| Напруга живлення | 24 В, 300 Вт |
| Габарити області друку | |
| Довжина | 255 мм |
| Ширина | 210 мм |
| Висота | 210 мм |

2.2 Розробка структурної схеми макету

Структурна схема дозволить у подальшому розробити алгоритм та програмний засіб для реалізації СК. Структурна схема зображена на рисунку 2.3.

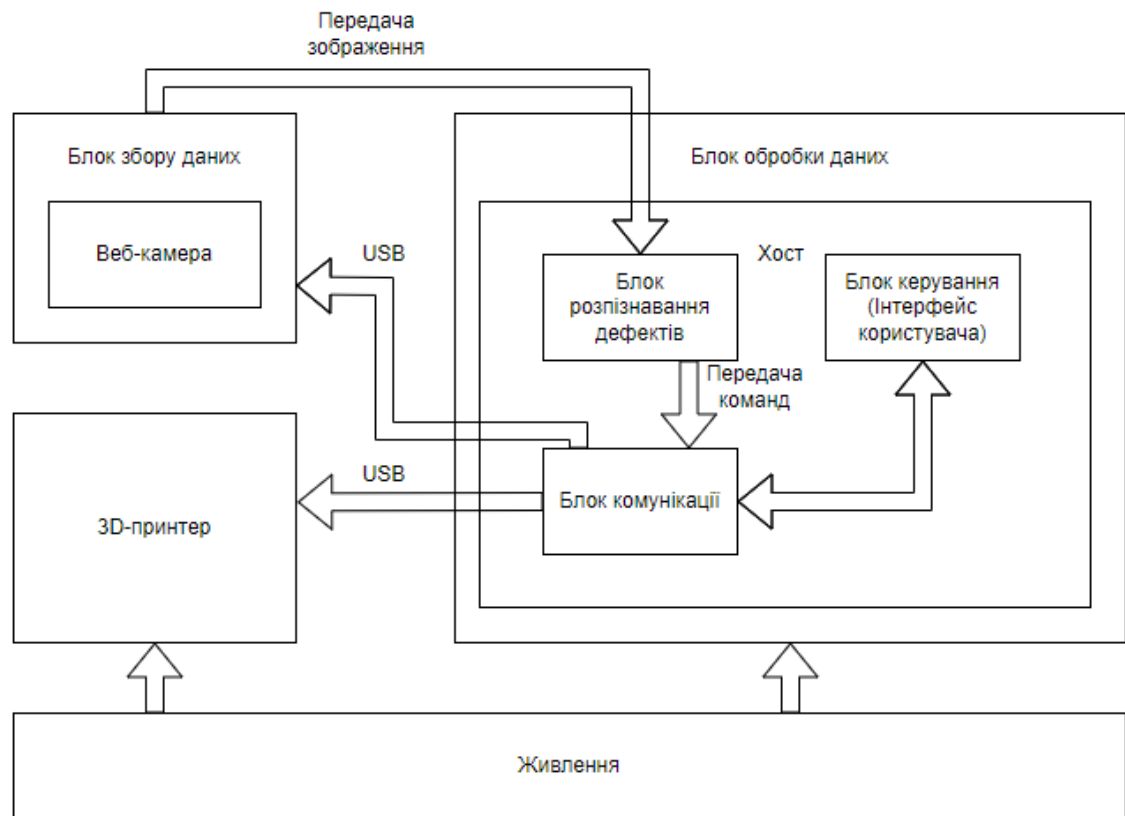


Рисунок 2.3 – Структурна схема макету

На даній схемі візуально зображено модулі, які будуть відповідати за роботу макета. До цих модулів входять:

- блок збору даних – це веб-камера, яка стоїть напроти принтера та передає зображення на блок обробки даних;
- блок обробки даних – це хост, або комп’ютер;
- 3D принтер – пристрій, який виготовляє необхідні запчастини;
- живлення – блок, який дозволяє працювати всій системі;
- блок розпізнавання дефектів – програмний модуль, який встановлений на хості. Саме він веде спостереження за 3D принтером і виявляє дефекти. Після виявлення дефекту надсилає команди на блок комунікації;
- блок керування (інтерфейс користувача) – програмний модуль, який надає змогу користувачеві слідкувати за роботою системи;

– блок комунікації – це інтерфейси, за допомогою яких відбувається комунікація між камерою та хостом і принтером та хостом.

Макет зображений на рисунку 2.4.

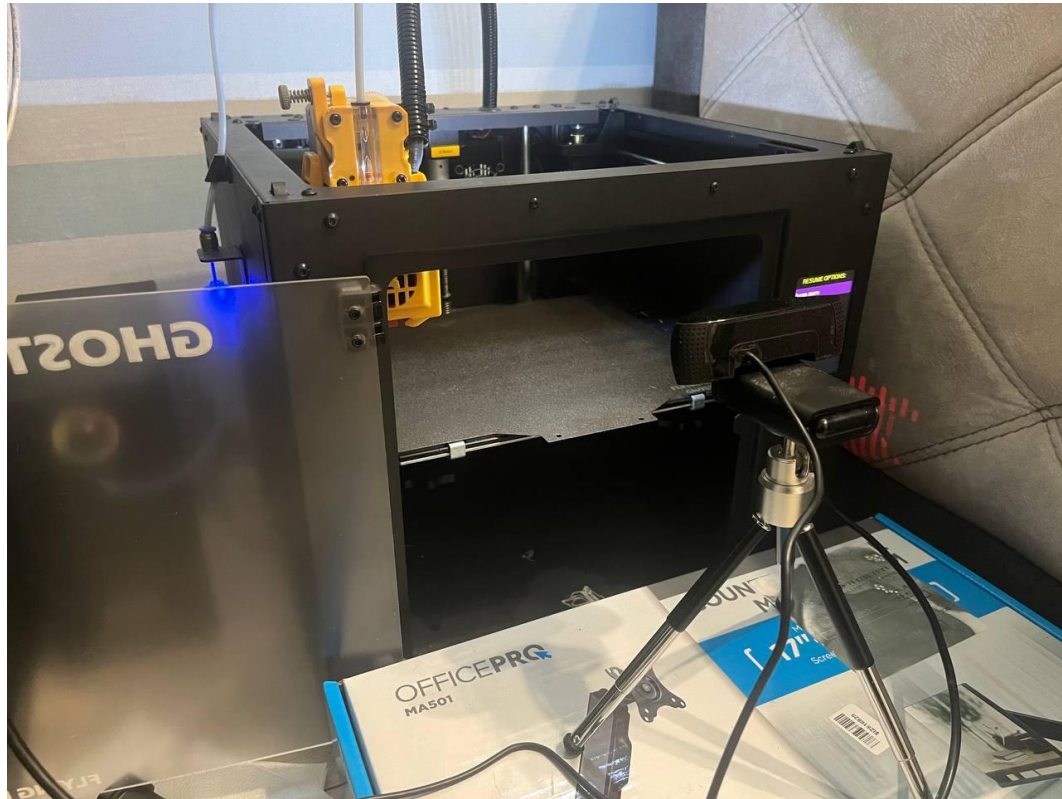


Рисунок 2.4 – Макет, який буде тестуватися

2.3 Розробка системи контролю деталей за допомогою штучної нейронної мережі

2.3.1 Обґрунтування використання нейронних мереж

Так як в даній роботі було зазначено, що розпізнавати потрібно дефекти при друці на 3D принтері, необхідно вирішити, який метод детектування необхідно використати.

Найпростішим методом у даній ситуації є метод каскадів (признаків Хаара), але він має ряд недоліків, через які він не підходить, хоча по швидкодії він один з найкращих:

- необхідно визначати у реальному часі (каскади Хаара гарно підходять для розпізнавання об'єктів на зображеннях);
- дуже сильно залежить від якості зображення;
- потребує значного використання ресурсів для визначення у реальному часі;
- велика кількість похибок та випадкові виявлення при змінному освітленні.

Методом, що більш підходить для вирішення цього завдання, є використання штучних нейронних мереж. Загалом нейронні мережі складніші за інші методи і потребують великої кількості даних та часу, але в порівнянні з іншими методами вони суттєво виграють у розпізнаванні об'єктів.

Ось деякі переваги використання нейронної мережі:

- вища точність. Глибокі нейронні мережі, зокрема згорткові нейронні мережі (CNN), мають здатність до вищої точності при виявленні об'єктів порівняно з більш традиційними методами, такими як каскад Хаара. Вони можуть розпізнавати об'єкти зі складними текстурями та формами;
- здатність до навчання. Нейронні мережі можуть бути навчені виявляти різні типи об'єктів, що робить їх дуже гнучкими та придатними для різноманітних завдань виявлення об'єктів;
- здатність до адаптації. Нейронні мережі можуть адаптуватися до змінних умов, таких як зміна освітлення, масштабу та орієнтації об'єкта, що дозволяє їм ефективно функціонувати в різних сценаріях;
- швидкість роботи. Завдяки оптимізації та апаратній підтримці, нейронні мережі можуть працювати в реальному часі на сучасних обчислювальних пристроях, включаючи мобільні та вбудовані системи;

- можливість виявлення багатьох типів об'єктів. За допомогою глибоких нейронних мереж можна виявляти різні типи об'єктів одночасно на одному зображенні. Це корисно в задачах, де потрібно аналізувати багато об'єктів різних типів;

- здатність до реалізації складних завдань. Нейронні мережі можуть виконувати багато завдань, включаючи сегментацію об'єктів, визначення об'єктів у 3D просторі та інші завдання, що виходять за межі простого виявлення;

- постійне покращення. Дослідження в області нейронних мереж та комп'ютерного зору постійно розвиваються, що дозволяє вдосконалювати точність та продуктивність систем виявлення об'єктів.

2.3.2 Вибір моделі нейронної мережі

На сьогоднішній день існує багато моделей нейронних мереж для комп'ютерного зору, які використовуються для завдань розпізнавання об'єктів, визначення класів, сегментації зображень та багато інших задач. Ось кілька основних моделей:

- Convolutional Neural Networks (CNNs). Згорткові нейронні мережі ЗНМ використовують різновид багатошарових перцептронів, розроблений таким чином, щоби вимагати використання мінімального обсягу попередньої обробки. Плюсом можна вважати те, що вони ефективні у виявленні та відображенні локальних шаблонів, що робить їх відмінними для роботи із зображеннями. Мають здатність автоматичного вивчення властивостей об'єктів на різних рівнях абстракції.

До мінусів можна віднести те, що вони можуть мати проблеми з інтерпретацією роботи на високому рівні. Не завжди ефективні для обробки зображень з різних кутів або різної якості [18];

– Residual Neural Networks (ResNets). Перевага даних моделей в тому, що вони дозволяють тренувати дуже глибокі нейронні мережі без проблеми з вирішенням проблеми зникнення градієнту.

Недоліками є те, що при додаванні великої кількості шарів може відбутися перенавчання моделі або ускладнення процесу її навчання [19];

– Inception Networks (Inception/GoogleNet). Перевага в тому, що вони використовують модулі різних розмірів фільтрів для виявлення об'єктів на різних масштабах, що дозволяє зменшити кількість параметрів та покращити швидкість обчислень.

До недоліків можна віднести складність розуміння та інтерпретації роботи мережі через використання різних розмірів фільтрів [20];

– Recurrent Neural Networks (RNNs). Перевагами є те, що вони добре працюють з послідовними даними, такими як текст чи часові ряди, тому можуть застосовуватися до завдань комп'ютерного зору, де потрібно аналізувати послідовність подій на зображеннях.

Недоліками є те, що вони мають проблеми з довгостроковою залежністю, можуть бути важкими для тренування через проблему зникнення градієнту [21];

– Transformer-based models (наприклад, Vision Transformers – ViT). Переваги в тому, що даний вид використовує механізми уваги для аналізу зображень, дозволяючи вирішувати завдання комп'ютерного зору за допомогою трансформаційних архітектур, які раніше використовувалися переважно для обробки тексту.

Недоліками даних моделей є те, що вони потребують більше обчислювальних ресурсів для тренування, можуть бути менш ефективними для менших наборів даних [22];

– Object Detection Models (наприклад, YOLO - You Only Look Once, Faster R-CNN, SSD – Single Shot Multibox Detector). До переваг можна віднести те, що

це – спеціалізовані моделі для виявлення об'єктів на зображеннях, здатні вирішувати проблему детектування об'єктів у реальному часі.

Недоліками є те, що вони можуть вимагати більш високих обчислювальних ресурсів для тренування та роботи.

В даному проекті буде використовуватись модель YOLO. Основні особливості моделі YOLO:

- YOLO використовує одну нейронну мережу, що працює одразу з усією картинкою, роблячи прогнози та виявляючи об'єкти в кожному сегменті зображення;

- дана модель пропонує швидкий процес обробки зображень та виявлення об'єктів порівняно з деякими іншими алгоритмами об'єктного виявлення;

- YOLO використовує одне пряме проходження мережі, щоб одночасно передбачити координати та класи об'єктів на зображенні.

Дана модель може працювати у режимі реального часу. Тобто вона добре підходить для задачі детектування дефектів у реальному часі, адже може обробляти вихідний відео потік.

Однією з основних переваг YOLO є її точність виявлення об'єктів. Це означає, що модель може ідентифікувати різноманітні дефекти з великою точністю, що дозволяє вчасно втручатися для уникнення виробничих недоліків.

Одним з недоліків можна вважати те, що дана модель схильна до хибних спрацювань, особливо в умовах поганої освітленості або поганої якості камери, а також її не рекомендується застосовувати у складних сценах, тобто з великою кількістю об'єктів в кадрі.

Проте, використання моделі YOLO є відносно простим у реалізації та налаштуванні порівняно з іншими складними моделями глибокого навчання. Це дозволяє легко адаптувати модель для виявлення конкретних дефектів у 3D друці шляхом тренування на відповідних даних. Архітектура моделі YOLOv5 зображена на рисунку 2.5;

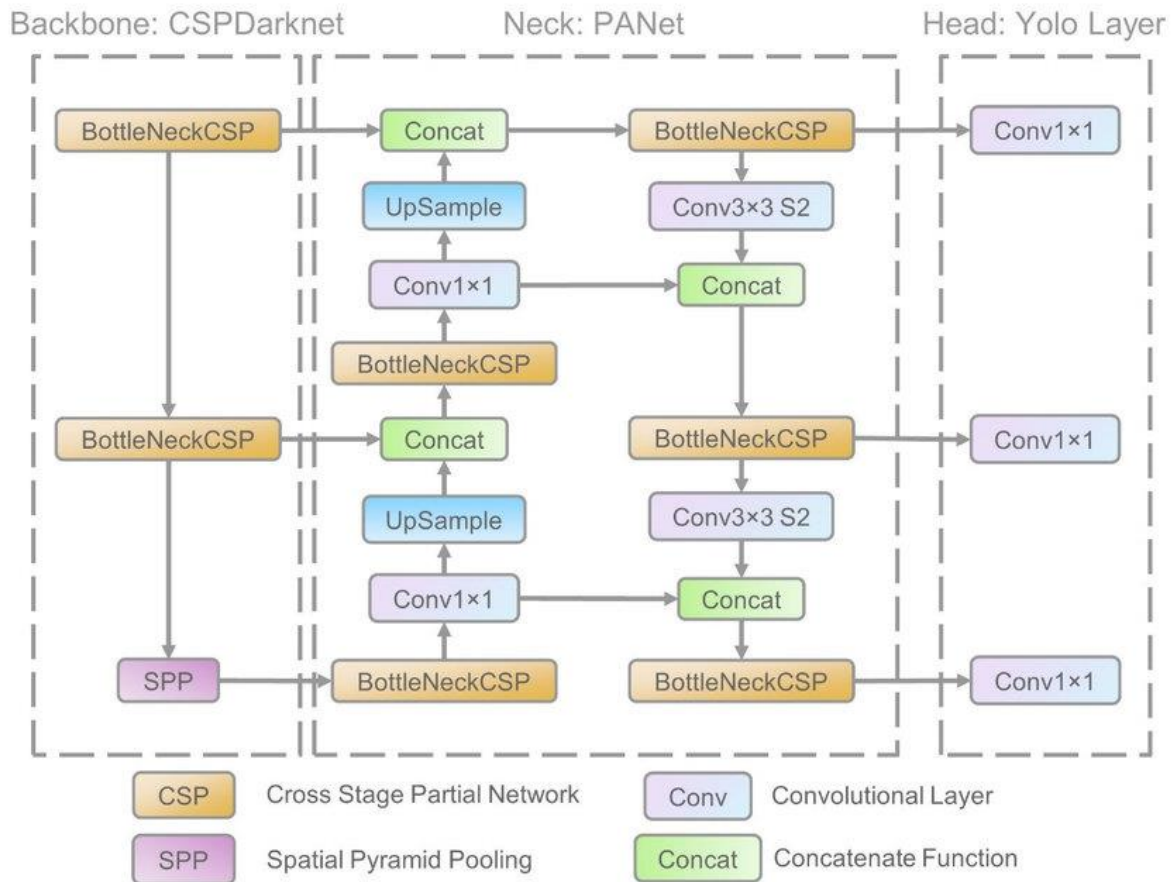


Рисунок 2.5 – Архітектура YOLOv5 [23]

– **BottleNeckCSP** (Bottleneck CSP) – це модуль, який поєднує в собі багато коротших шляхів (shortcuts) із стрижнями (bottleneck blocks), що дозволяє оптимізувати обчислювальні ресурси та поліпшувати ефективність нейронної мережі;

– **Concat** (Concatenate) – це операція конкатенації, яка об'єднує дані або фільтри з різних шляхів обробки в одному тензорі. Використовується для об'єднання інформації з різних частин мережі;

– **UpSample** (Upsampling) – ця операція використовується для збільшення розмірів тензору. У YOLOv5 це може бути зроблено, наприклад, за допомогою білінійного апсемплінгу (bilinear upsampling), щоб збільшити роздільну здатність об'єктів на вищому рівні мережі;

– Conv1x1 (1x1 Convolution) – це згортковий шар з ядром розміром 1x1. Використовується для зменшення або збільшення кількості каналів та комбінування інформації між ними;

– SPP (Spatial Pyramid Pooling) – це модуль, який дозволяє моделі працювати з об'єктами різних розмірів, захоплюючи інформацію на різних масштабах (пулінг на просторових пірамідах);

– Conv3x3 S2 (3x3 Convolution with Stride 2) – це згортковий шар з ядром розміром 3x3 та з кроком 2. Використовується для зменшення розмірності зображення (спаду) з використанням згорткових операцій [23].

Для того, щоб модель працювала правильно в умовах даного проекту, необхідно навчити її на власному наборі даних. Перед навчанням моделі є декілька важливих кроків які необхідно виконати, а саме:

- підготувати власний набір даних (dataset);
- розмітити кожне зображення з набору даних, щоб отримати координати дефектів;
- створити файлову структуру необхідну для проекту;
- стиснути дані.

У даному випадку для навчання моделі буде використовуватись сервіс Google Colab. Цей сервіс дозволяє використовувати необхідні для навчання моделей нейронних мереж ресурси. Наприклад для навчання буде використаний графічний прискорювач Nvidia Tesla V100, який спеціально був розроблений та спрямований на підвищення продуктивності в навчанні нейронних мереж.

2.4 Алгоритм розпізнавання дефектів

Для подальшої розробки програми для розпізнавання дефектів був розроблений алгоритм її роботи. Цей алгоритм демонструє як буде поводити себе система, якщо дефект виявлено.

Даний алгоритм використовує попередньо навчену модель штучної нейронної мережі.

При старті перевіряється зв'язок з послідовним портом, через який підключений принтер. Це зроблено для того, щоб надіслати команду екстреного припинення друку.

Після цього встановлюється зв'язок з камерою. Це зроблено для того, щоб уникнути помилок під час виконання програми. Після того як зв'язок з камерою встановлено, зображення виводиться у вікні на комп'ютері. Це зроблено для того, щоб користувач бачив як працює програма.

У програми є постійний зв'язок з попередньо навченою моделлю нейронною мережею. При появі дефекту виконується умова, після якої навколо дефекту малюється прямокутник з надписом, на якому зазначається вид дефекту. Після чого виконується умова, яка перевіряє скільки часу в кадрі знаходиться об'єкт з дефектом і, якщо такий об'єкт наявний впродовж більш як 30 секунд, то програма надсилає команду для екстреного закінчення друку на 3D принтер. Ця перевірка зроблена задля забезпечення більшої точності. Так як модель нейронної мережі може бути не в змозі відрізнити в деяких моментах дефект від звичайної деталі, виконується ця умова. Так як у даному проекті швидкість реакції не повинна бути миттєвою, адже сам процес 3D друку досить довгий, було вирішено зробити час на перевірку в 30 секунд, що повинно виключити майже всі хибні спрацювання.

Після того як перевірку буде пройдено програма повинна надіслати команду про екстрену зупинку друку у послідовний порт 3D принтера, закрити зв'язок з принтером, завершити зв'язок з камерою та завершити свою роботу з виводом інформації про виявлений дефект та відсотком впевненості моделі.

Алгоритм зображений на рисунку 2.6.

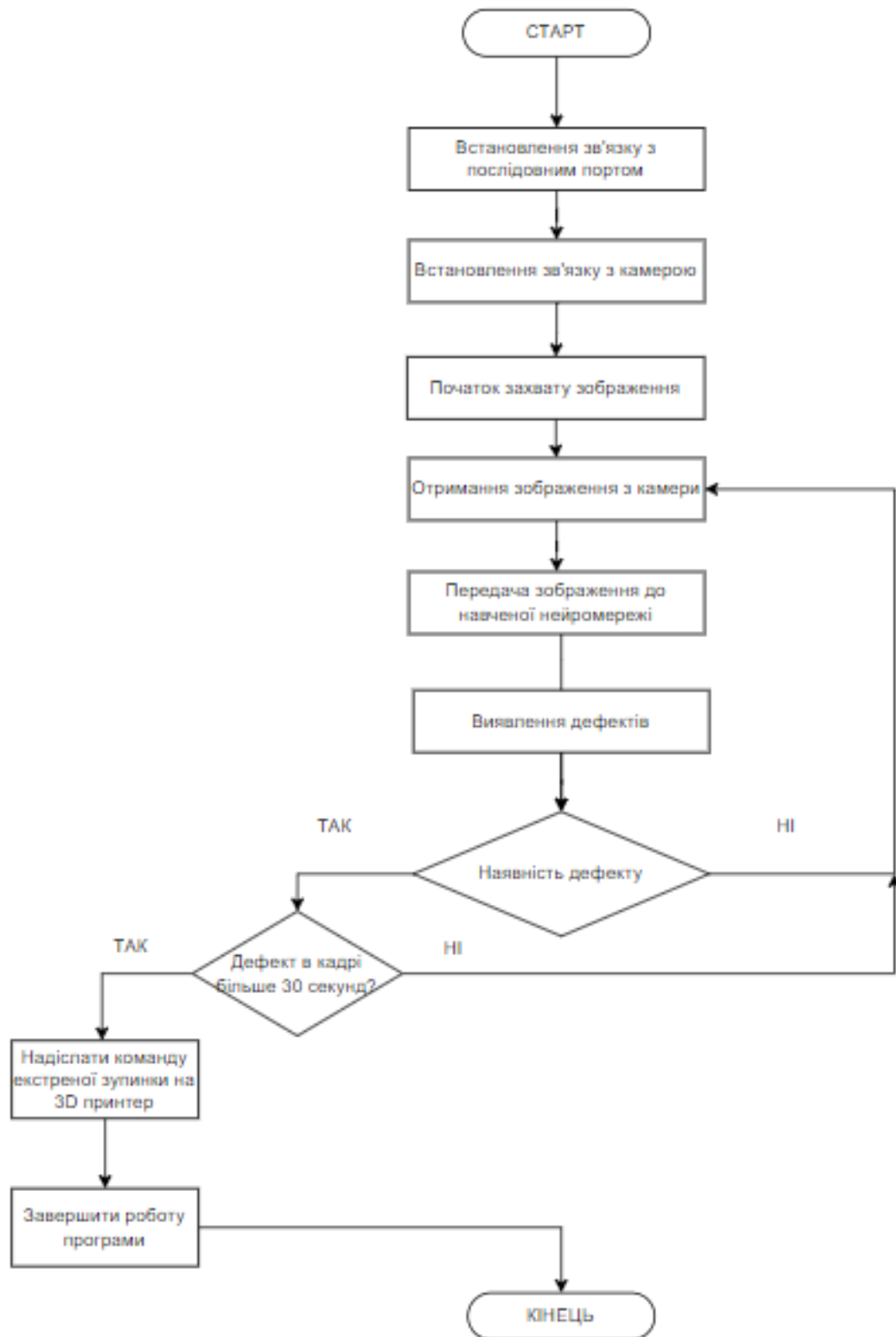


Рисунок 2.6 – Алгоритм роботи макету

В даному алгоритмі є пункт, який називається «Виявлення дефектів». Даному пункту слід приділити особливу увагу і привести його алгоритм роботи окремо. Цей пункт передбачає, що передача зображень на вхід нейронної мережі вже відбулася, тобто необхідно привести алгоритм роботи, яку і описує пункт «Виявлення дефектів». Загальний алгоритм роботи моделі нейронної мережі, яка використовує архітектуру YOLO наведений на рисунку 2.7.

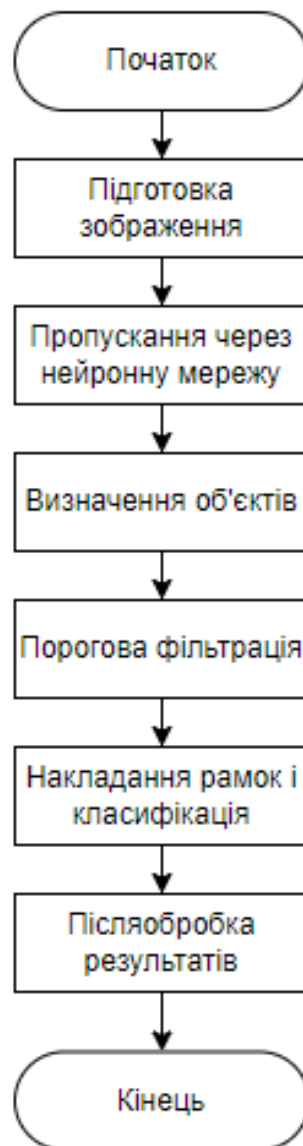


Рисунок 2.7 – Алгоритм роботи моделі нейронної мережі з архітектурою YOLO

Підготовка зображення – це процес коли вхідне зображення піддається попередній обробці, такій як зміна розміру, нормалізація пікселів, перетворення до формату, прийнятного для моделі тощо.

Визначення об'єктів включає наступні етапи:

- модель розпізнає області на зображенні, в яких ймовірно можуть знаходитися об'єкти, використовуючи анкори (anchors) та класифікатори;
- кожна клітина в остаточному обробленому зображенні мережі відповідає певному регіону на вихідному зображенні. Модель виводить прогнозовані прямокутні рамки (bounding boxes) разом із ймовірностями належності цих рамок до певних класів об'єктів;
- для кожної рамки модель видає набір координат (центр, ширина, висота) та ймовірність того, що область містить об'єкт певного класу.

Порогова фільтрація:

- результати визначення об'єктів фільтруються за певним порогом ймовірності, який визначає, наскільки впевнена модель у виявленні об'єкта;
- рамки з низькими ймовірностями можуть бути відкинуті.

Накладання рамок і класифікація:

- для кожної обробленої рамки встановлюється клас об'єкту, якому вона найбільш відповідає, на основі ймовірності та класифікаторів;
- остаточні рамки, що містять об'єкти, можуть бути позначені на вихідному зображенні для подальшого використання або візуалізації.

2.5 Розробка підпрограми для зупинки 3D друку

Так як 3D принтер буде працювати на прошивці Marlin, необхідно дослідити яким чином буде зупинятися процес друку при виявленні помилки.

У Marlin є декілька команд G-коду, які призупиняють друк, але працюють вони по різному.

M0 або M1 – команда, яка використовується для призупинення 3D друку. Коли принтер отримує команду M0, він зупиняє роботу тимчасово, чекаючи на підтвердження від користувача, перш ніж продовжити друк або виконати інші команди. Команда M0 використовується для створення паузи у процесі друку. Коли принтер досягає цієї команди в своєму програмному коді, він припиняє виконання і чекає на втручання оператора. Це може бути корисним у випадках, коли потрібно змінити матеріал, перевірити друкований об'єкт, внести корективи або здійснити інші дії під час процесу друку.

M112 – команда для негайного припинення друку. Коли принтер отримує команду M112, він припиняє будь-які поточні операції друку негайно. Ця команда може бути використана в разі виявлення помилки, неполадки у роботі принтера або в ситуаціях, коли необхідно негайно зупинити процес друку з безпекових причин. Після виконання команди M112 принтер переходить у режим припинення роботи. Це означає, що друкарська головка або інші рухомі частини принтера зупиняються, а друкарський процес припиняється. Важливо зауважити, що після виклику команди M112 принтер може вимагати перезавантаження або перезавантаження для відновлення нормального функціонування [24].

Виходячи з цього, в підпрограмі для припинення друку буде використовуватись команда M112. Тобто після отримання даних від основної програми про те, що дефект виявлено, запуститься підпрограма для відміни, яка відправить команду M112 побайтово за допомогою послідовного порту 3D принтера, друк зупиниться і на екрані 3D принтера з'явиться повідомлення, яке зображено на рисунку 2.8.

Повідомлення свідчить, що принтер необхідно перезавантажити для подальшої роботи.

Команда M0 у даному випадку не підходить тому, що вона призупиняє друк. Тобто вона спрацьовує, чекає втручання користувача і потім друк продовжується. Втручання користувача у даному випадку це просто необхідність

натиснути один раз по ємкісному екрану принтера, після чого друк відновлюється. Команда M112 не дає змоги продовжити друк, що в даному випадку є більш вірним рішенням.



Рисунок 2.8 – Результат спрацювання команди M112

Лістинг підпрограми зупинення друку приведений нижче на рисунку 2.9.

```
# Налаштування послідовного порту
serial_port = 'COM3'
baud_rate = 9600
# установка з'єднання з принтером за допомогою COM
try:
    ser = serial.Serial(port=serial_port, baudrate=baud_rate)
    print(f"Connected to {serial_port} at {baud_rate} bps")
except serial.SerialException as e:
    print(f"Error: {e}")
    exit()
# команда припинення друку
custom_message = "\nM112"
# відправка команди принтеру
ser.write(bytes(custom_message, 'utf-8'))
```

Рисунок 2.9 – Підпрограма для зупинки друку

2.6 Висновки до другого розділу

У результаті виконання другого розділу проведено підбір обладнання. Після чого було проаналізовано переваги та недоліки використання нейронної мережі та методу каскадів Хаара у даному проекті. Визначено основні моделі нейронних мереж, розглянуто недоліки і переваги кожної з моделей. В результаті для реалізації даного проекту вибрано модель нейронної мережі YOLO.

Розроблено структурну схему макету, яка включає в себе 3 основні блоки, такі як блок обробки даних, блок збору даних та 3D принтер.

Після чого був розроблений алгоритм роботи програми для визначення дефектів при 3D друці, та розібраний алгоритм роботи нейронної мережі.

Розроблено підпрограму для зупинки друку при виявленні дефекту та порівняно команди такі як M0 та M112.

У наступному розділі необхідно буде підготувати дані для навчання моделі, навчити модель нейронної мережі на власному наборі даних та перевірити коректність її роботи у різних умовах.

3 НАВЧАННЯ МОДЕЛІ НЕЙРОННОЇ МЕРЕЖІ ТА ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

3.1 Вибір мови програмування та розробка програмного засобу для СК

У даному проекті в якості мови програмування для застосунку виступає Python. Основною причиною того, чому саме ця мова програмування використовується у даному проекті є те, що Python має широкий вибір бібліотек для машинного навчання і обробки зображень, таких як PyTorch, TensorFlow, OpenCV тощо. Це дозволяє легко і ефективно використовувати моделі, такі як YOLO, та обробляти зображення в реальному часі.

Програма починається зі встановлення зв'язку з послідовним портом до якого підключено принтер. Дана функція представлена на рисунку 3.1.

```
# Функція для вибору COM-порту
def choose_serial_port():
    port = input("Введіть назву COM-порту (наприклад, COM3): ")
    return port

# Функція для підключення до послідовного порту
def connect_serial(port, baud_rate):
    try:
        ser = serial.Serial(port=port, baudrate=baud_rate)
        print(f"Підключено до {port} з швидкістю {baud_rate} біт/с")
        return ser
    except serial.SerialException as e:
        print(f"Помилка: {e}")
        exit()
```

Рисунок 3.1 – Функція для встановлення зв'язку з COM-портом

Ця функція призначена для отримання від користувача назви COM-порту, яку він вводить з клавіатури. Після введення функція повертає отриману назву

порту. Це корисно в сценаріях, де програма повинна взаємодіяти з конкретним послідовним портом, і користувач може обирати його за допомогою введення. Це дозволяє користувачеві вибрати порт перед подальшим використанням програми для взаємодії з обраним послідовним портом.

Після цього виконуються функції, які завантажують модель yolo та відкривають захват з камера (рис. 3.2).

```
# Функція для завантаження моделі YOLOv5
def load_yolov5_model(model_path):
    return torch.hub.load('ultralytics/yolov5', 'custom', path=model_path)

# Функція для відкриття вебкамери
def open_webcam(camera_index):
    return cv2.VideoCapture(camera_index)
```

Рисунок 3.2 – Завантаження моделі та відкриття потоку

Функція `load_yolov5_model` завантажує модель YOLOv5 для обробки зображень з використанням бібліотеки PyTorch та пакету `ultralytics/yolov5`. Вона приймає шлях до збереженої моделі як аргумент і повертає завантажену YOLOv5-модель.

Функція `open_webcam` створює об'єкт відеозахоплення за допомогою бібліотеки OpenCV, приймаючи індекс вебкамери як аргумент. Вона повертає цей об'єкт, що може бути використаний для отримання кадрів з вебкамери у реальному часі для подальшого аналізу чи обробки.

Далі виконуються функції виявлення об'єктів в кадрі і функція відправки повідомлення у послідовний порт. Фрагмент коду, що демонструє роботу даних функцій представлено на рисунку 3.3.

Функція `detect_objects` використовує передану модель YOLO для виявлення об'єктів на кадрі. Параметри функції включають сам кадр `frame`,

завантажену модель `model` та поріг впевненості `confidence_threshold`. Функція повертає відфільтровані результати у вигляді тензору, який містить інформацію про об'єкти, таку як координати, клас об'єкта та його впевненість.

Функція `send_message_to_serial` відправляє передане повідомлення (`message`) на послідовний порт, використовуючи об'єкт `ser` типу `serial.Serial`. Якщо відправка успішна, функція повертає `True`, в іншому випадку виводить повідомлення про помилку та повертає `False`.

```
# Функція для виявлення об'єктів на кадрі за допомогою моделі YOLOv5
def detect_objects(frame, model, confidence_threshold):
    results = model(frame)
    pred = results.pred[0]
    return pred[pred[:, 4] > confidence_threshold]

# Функція для відправлення повідомлення на послідовний порт
def send_message_to_serial(ser, message):
    try:
        ser.write(bytes(message, 'utf-8'))
        return True
    except serial.SerialException as e:
        print(f"Помилка відправлення даних на {ser.port}: {e}")
        return False
```

Рисунок 3.3 – Функції виявлення об'єктів та відправки повідомлення у COM-порт

У даній програмі шлях до навченої моделі YOLO та індекс камери, як і серійний порт повинен вводити користувач під час її запуску.

Після даних функцій виконується головна функція `main()`, в якій зберігаються такі змінні, як поріг впевненості моделі при якому дефекти, які менше ніж даний поріг не будуть враховуватись та поріг тривалості затримки, від значення якої буде залежати скільки об'єкт має утримуватись в кадрі, щоб було надіслано команду екстреної зупинки друку. Також дана функція включає в себе

основний цикл для обробки відеопотоку, в якому використовується попередньо описана функція `detect_objects`, якщо об'єкт було виявлено і його впевненість вища за вказаний поріг то запускається таймер, який перевіряє час утримання об'єкту в кадрі, і якщо він вище за поріг тривалості затримки, то спрацьовує відправка команди M112, а якщо ні то таймер скидається.

Також в цій функції рисуються прямокутники навколо дефектів які були знайдені, виводиться номер класу дефекту, його назва та впевненість моделі. За допомогою бібліотеки OpenCV виводиться оброблений кадр і виконується умова, яка дозволяє завершити програму за допомогою клавіші q. Фрагмент даної функції зображений на рисунку 3.4.

```
# Основний цикл для обробки відео
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    filtered_pred = detect_objects(frame, model, confidence_threshold)

    if len(filtered_pred) > 0 and filtered_pred[0, 4] > confidence_threshold:
        if start_time is None:
            start_time = time.time()
        elif time.time() - start_time > timeout_duration_threshold and not message_sent:
            message_sent = send_message_to_serial(ser, custom_message)
            start_time = None
            if message_sent and not defect_detected:
                defect_detected = True
                print(f'Об'єкт виявлено з достовірністю: {filtered_pred[0, 4]:.2f}')
                print("Команда M112 відправлена. Завершення...")
                ser.close()
                break # Завершення програми після першого виявлення дефекту
    else:
        start_time = None
        message_sent = False
```

Рисунок 3.4 – Фрагмент основної функції

Повний лістинг програми приведений у додатку Б.

3.2 Підготовка набору даних для навчання

Для підготовки набору даних в загальному випадку необхідно спочатку зібрати необхідну кількість зображень кожного класу. Чим більше зображень, тим краще модель буде працювати. Дані зображення можна отримати декількома способами. Перший з них це просто зібрати необхідну кількість фотографій та привести їх до одного розміру. Цей метод дуже складний в реалізації, адже для коректного навчання моделі, необхідно подавати тисячі таких зображень. Кращим способом збору даних у даному випадку буде запис відео з подальшим розбиттям його на зображення покадрово. В даному випадку клас об'єкту буде один та матиме назву fail. В даному проекті будуть використані готові зображення для першого навчання моделі, а потім буде використаний власний набір для її донавчання.

Після того як зображення були зібрані, їх необхідно розмітити. Тобто необхідно вказати, на якій області зображення наявний наступний клас. Для цього можна використовувати різні програми та сервіси для розмітки. У даному проекті для розмітки зображень була використана програма OpenLabeling-master. Це програма написана на мові Python. Вона має примітивний інтерфейс користувача, але для задачі розмітки зображень підходить ідеально (рис. 3.5).

Даний етап дасть змогу алгоритму навчатися і розпізнавати задані класи, пов'язуючи візуальні ознаки в обмежувальних рамках. Правильно промарковані набори даних значно підвищують точність моделі та її здатність до узагальнення на нові, невідомі дані. Крім того, позначені зображення слугують еталоном для оцінювання ефективності моделі на етапах навчання і перевірки.

Деякі зображення можуть не містити дефекти, тому під час розмітки зображень в файл не записуються ніякі дані. Так, під час навчання, модель навчитися розрізняти дефекти від звичайних об'єктів у кадрі.

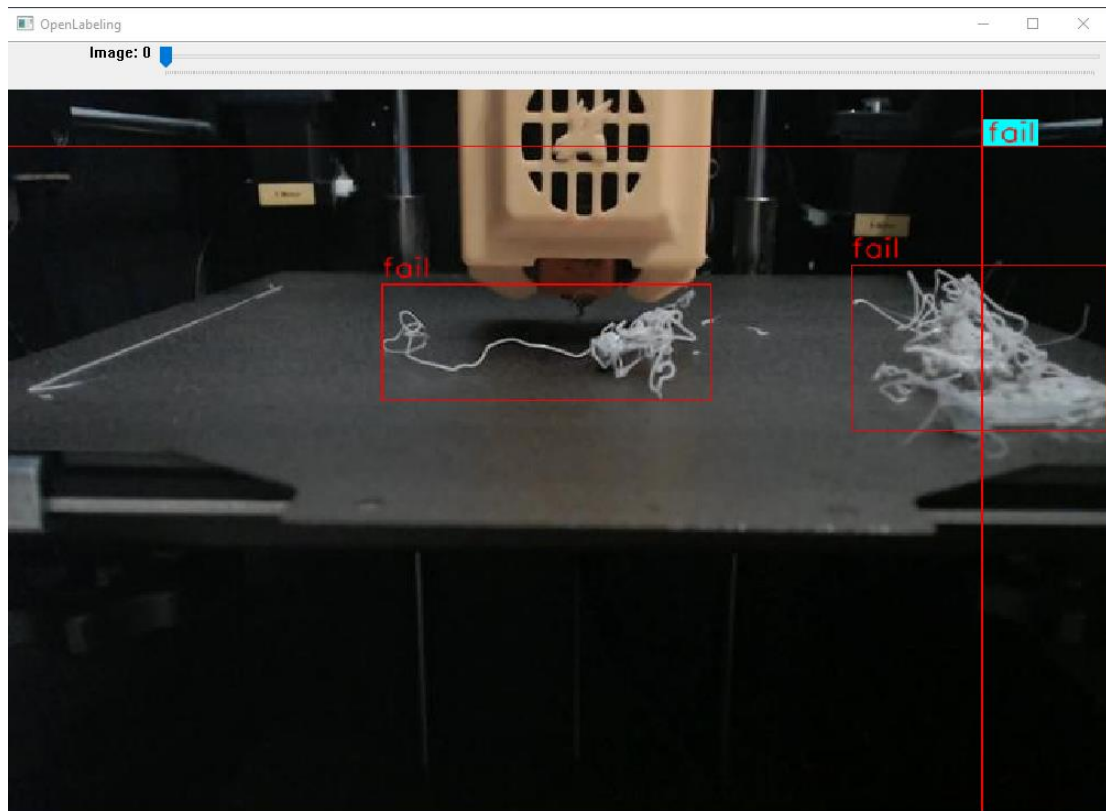


Рисунок 3.5 – OpenLabeling-master

Для розмітки зображень необхідно обводити необхідні класи прямокутниками. При запуску програми в її каталозі output створюються файли з розширенням .txt. Цих файлів буде стільки ж, скільки зображень було подано на вхід, і кожен з цих файлів містить координати певного класу на зображенні. Приклад вмісту такого файлу показано на рисунку 3.6.

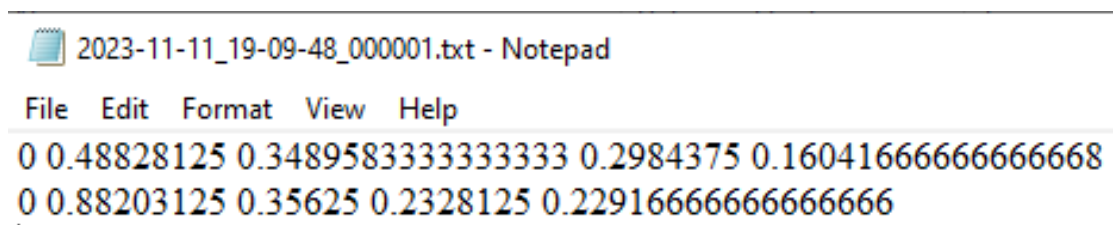


Рисунок 3.6 – Вихідний файл з координатами класу

Кожен рядок містить інформацію про один об'єкт у зображенні, де перше число – це клас об'єкта, а наступні чотири – це числа, які представляють координати обмежувальної рамки об'єкта.

Після підготовки даних для навчання необхідно створити структуру проекту. Структура повинна виглядати наступним чином (рис. 3.7).

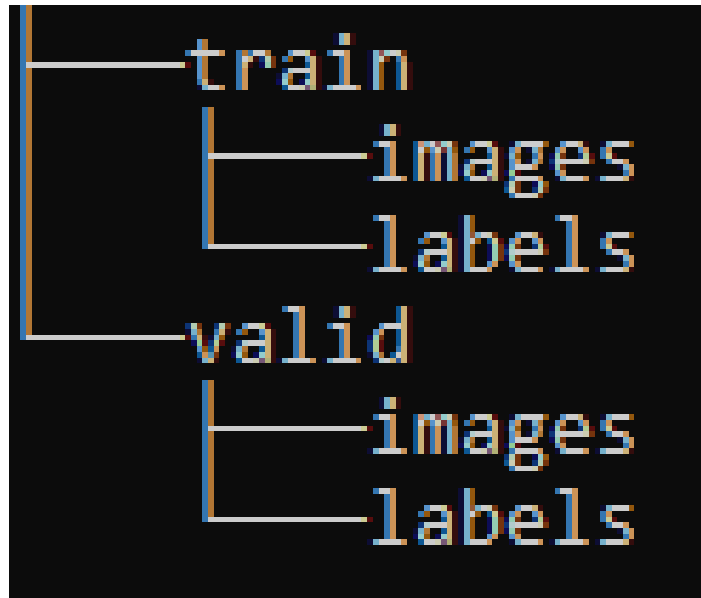


Рисунок 3.7 – Структура проекту

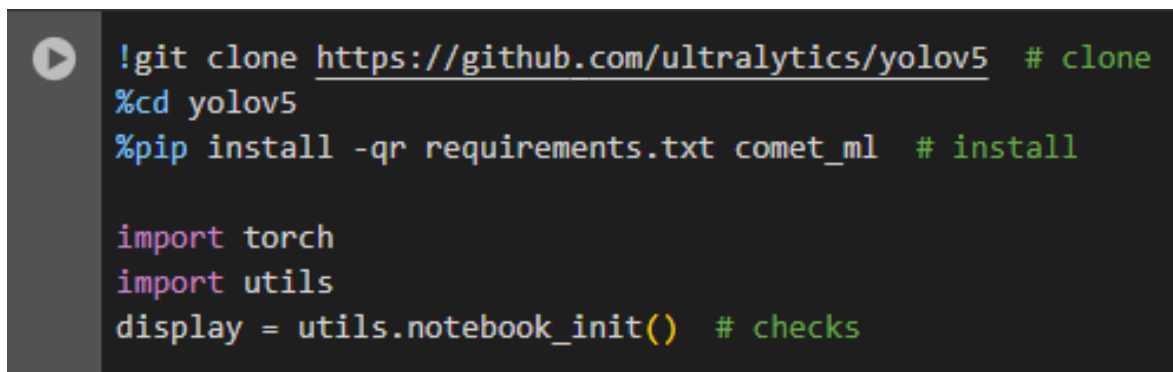
Як можна побачити необхідно створити декілька каталогів. Загальний каталог буде називатися data. Потім необхідно створити ще два каталоги – train і valid, в яких в свою чергу вже створюються каталоги images і labels. Цю структуру вимагає модель YOLO. Каталог train – це дані, які будуть використовуватись під час навчання моделі, каталог valid – це набір даних для перевірки. Для простоти помістимо весь набір даних і в каталог train і в valid. Images – це каталог для зібраних зображень, а labels – це каталог для файлів, які містять координати класів об'єктів.

Після створення структури необхідно архівувати загальний каталог.

3.3 Навчання моделі нейронної мережі

Навчання моделі нейронної мережі відбувалось в середовищі Google Colab. Це хмарне середовище, яке дозволяє використовувати ресурси для навчання моделей нейронних мереж. Завдяки Google Colab навчити модель можна набагато швидше, ніж на звичайному хості, так як там використовуються спеціальні графічні прискорювачі, які були створені для навчання.

Необхідно підключитися до віртуальної машини і створити структуру проекту, після чого необхідно додати дані для навчання і розархівувати їх у каталог проекту (рис. 3.8).



```
!git clone https://github.com/ultralytics/yolov5 # clone
%cd yolov5
%pip install -qr requirements.txt comet_ml # install

import torch
import utils
display = utils.notebook_init() # checks
```

Рисунок 3.8 – Створення структури проекту

Після цього повинно з'явитися стандартне дерево проекту (рис. 3.9).

Необхідно розархівувати дані для навчання і відредагувати файл `soco128.yaml`. Даний файл містить інформацію про вже існуючі класи об'єктів, але так як необхідно створити власну модель зі своїми вагами, то всі класи, що там знаходяться непотрібні. Тому необхідно повністю замінити код в цьому файлі на той, що зображений на рисунку 3.10. Даний файл містить інформацію про шляхи до даних для навчання, кількість та назви класів, також опціонально

можна протестувати навчену модель, для цього необхідно створити ще один каталог з зображеннями для тестування.

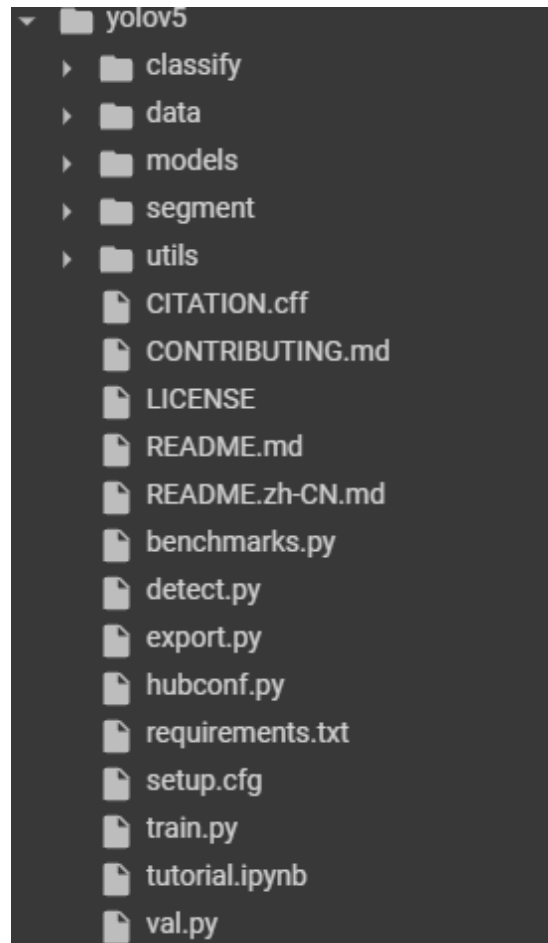


Рисунок 3.9 – Дерево проекту

```
1 train: /content/yolov5/data/images/train
2 val: /content/yolov5/data/images/valid
3
4 # Classes
5 names:
6 | 0: fail
7
```

Рисунок 3.10 – Вміст файлу coco128

Після зміни вмісту цього файлу, можна перейти до навчання моделі. Для навчання необхідно виконати наступну команду:

```
«!python train.py --img 640 --batch 16 --epochs 100 --data coco128.yaml --weights yolov5s.pt --cache»,
```

де `!python train.py` – запуск скрипта `train.py` для навчання,

`--img 640` – аргумент, який задає розмір вхідних зображень для навчання моделі. В даному випадку зображення будуть приведені до розміру 640x640,

`--batch 16` – визначає розмір пакету (batch size), тобто кількість зображень, яка буде оброблена за один раз,

`--epochs 100` – кількість епох навчання,

`--data coco128.yaml` – вказує шлях до файлу даних, який містить конфігурацію для навчання моделі

`--weights yolov5s.pt` – вказує шлях до файлу ваг моделі, які будуть використовуватись в якості початкових ваг для навчання,

`--cache` – дозволяє кешувати дані, що може пришвидшити повторне навчання.

Після того як модель пройде етап навчання, можна буде скачати файл ваг, за допомогою яких і буде працювати програма для детекції дефектів.

Даний етап навчання було запущено з кількістю в 100 епох. Після навчання було отримано результати наведені на рисунку 3.11.

Як видно з цього рисунка, що 100 епох навчання замало для даної моделі, бо вона має доволі низьку впевненість mAP50-95. Тому необхідно перезапустити навчання моделі з кількістю в 300 епох навчання, щоб забезпечити нормальне значення впевненості моделі, а також підготувати більший набір даних для коректної роботи системи контролю.

В даному випадку був обраний набір даних, який складався з зображень конкретної моделі і містив фотографії лише деталей з дефектами. Для того, щоб модель нейронної мережі не розцінювала звичайну деталь як дефект, необхідно

подавати на вхід зображення деталей, на яких не міститься дефектів і, як наслідок, не розмічати дане зображення, а залишати пустим файл з мітками.

```
100 epochs completed in 0.299 hours.
Optimizer stripped from runs/train/exp/weights/last.pt, 14.4MB
Optimizer stripped from runs/train/exp/weights/best.pt, 14.4MB

Validating runs/train/exp/weights/best.pt...
Fusing layers...
Model summary: 157 layers, 7012822 parameters, 0 gradients, 15.8 GFLOPs

```

| | Class | Images | Instances | P | R | mAP50 | mAP50-95: |
|--|-------|--------|-----------|-------|---|-------|-----------|
| | all | 447 | 470 | 0.994 | 1 | 0.995 | 0.811 |

```
Results saved to runs/train/exp
```

Рисунок 3.11 – Результати впевненості після першого навчання

Результат роботи системи після першого навчання моделі нейронної мережі зображений на рисунку 3.12.



Рисунок 3.12 – Результат першого навчання моделі

Результат навчання хоча і доволі непоганий, не дивлячись на впевненість моделі у тому, що це дійсно дефект у 75%, є доволі серйозна проблема у тому, що модель взагалі не може розпізнати дефект (рис. 3.13).

Тобто з рисунка видно, що система не може розпізнати даний дефект, а інколи навіть розпізнає звичайний об'єкт як дефект, що для даної системи є недопустимим, тому було вирішено змінити набір даних, навчити модель на новому наборі, а потім донавчити вже навчену модель на власному наборі даних, що має збільшити впевненість моделі, а також виключити хибні спрацювання.



Рисунок 3.13 – Результат невдалої детекції дефекту

Після проведення другого навчання моделі, в якому було змінено кількість епох навчання, було протестовано її точність визначення дефектів. Результат визначення дефектів приведений на рисунку 3.14.

Слід зазначити, що всі тестування проводилися у режимі реального часу.

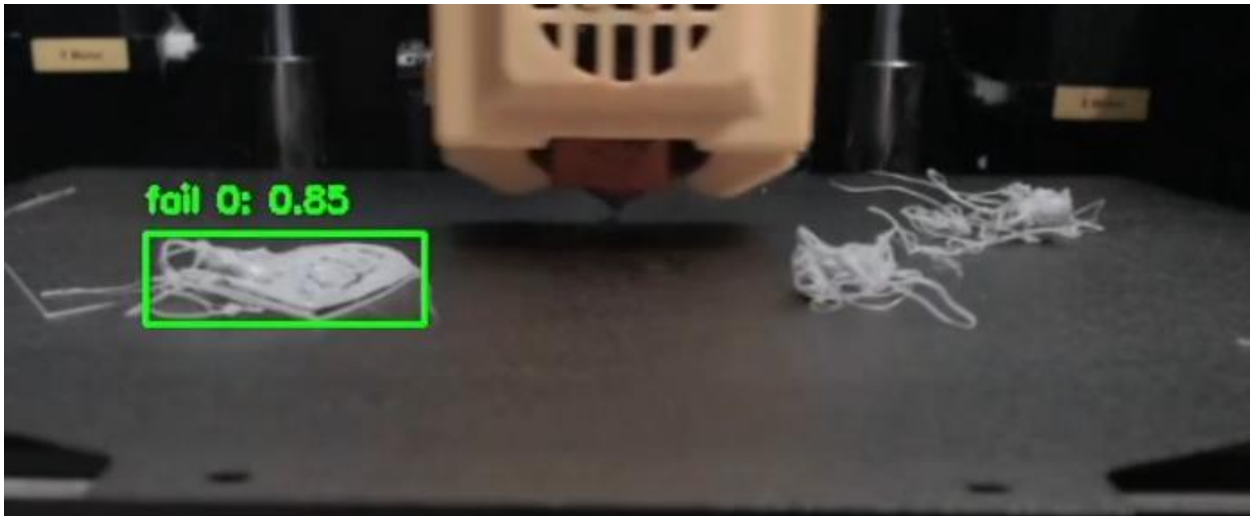


Рисунок 3.14 – Результати детекції після другого навчання зі збільшенням кількості епох

Як можна побачити з рисунку 3.14 – результат досить невдалий, адже об’єкти в кадрі то захоплюються, то припиняють захоплюватись. Точність розпізнавання доволі низька. Відсоток хибних спрацювань досить високий. Це відбувається через те, що набір даних, який використовувався в переважній більшості зображень включав досить різні моделі 3D принтерів. Тому можна зробити висновок, що дана модель нейронної мережі є недонавченою.

Для того щоб донавчити модель необхідно зібрати власний набір даних. В даному проекті для цього було записано 2 відеозаписи, після чого їх покадрово було розбито на 450 зображень. Цього вистачило, щоб довчити модель. Даний набір містить як зображення з дефектами, так і зображення, на яких дефектів немає, що дозволяє моделі навчитися розпізнавати тільки дефекти.

Процес навчання аналогічний тому, що використовувався при першому навчання, за виключенням того, що тепер замість стандартних ваг `yolov5.pt` будуть використовуватись ваги, які були отримані під час першого навчання моделі. Кількість зображень буде меншою, тому навчання пройде набагато швидше.

Після донавчання моделі можна оцінити її точність за допомогою графіків які приводяться разом з файлом ваг.

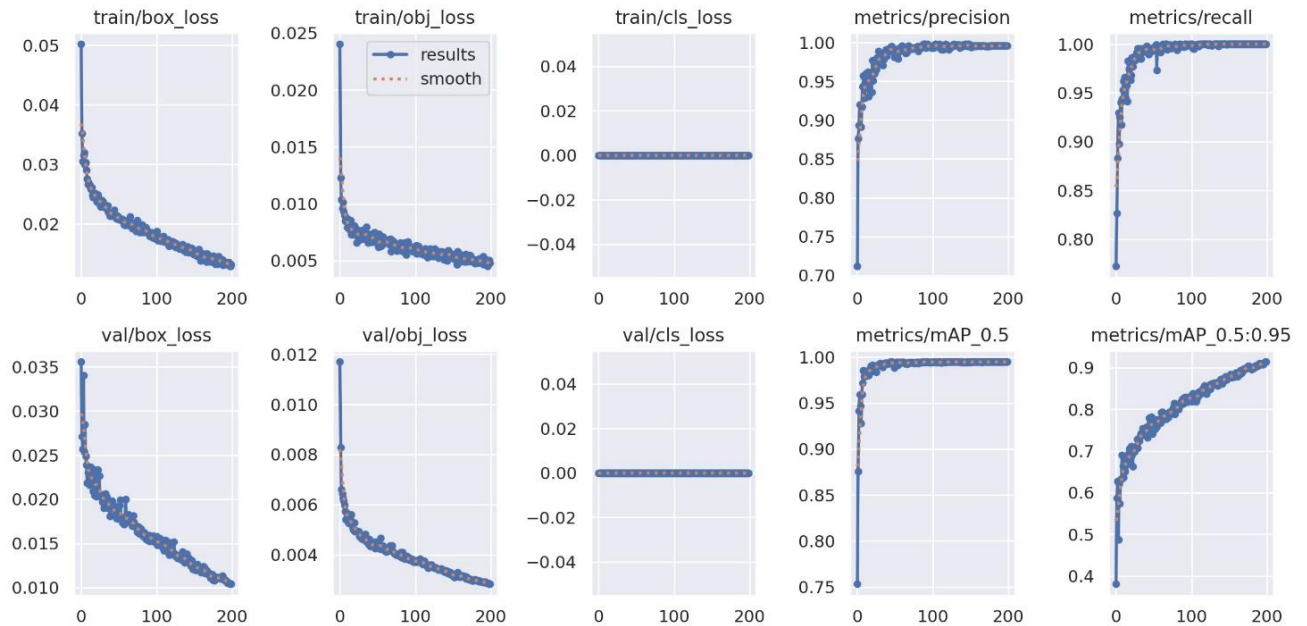


Рисунок 3.15 – Результати навчання моделі

На рисунку 3.15 зображені результати навчання моделі на власному наборі даних, де:

- `train/box_loss` – це графік, який вказує на втрати, пов'язані з розташуванням та розмірами обмежуючих рамок (bounding boxes) під час тренування моделі. Тобто чим нижче даний параметр, тим краще модель може визначити розміри дефекту;

- `train/obj_loss` – ця втрата оцінює, наскільки точно модель виявляє об'єкти в межах обмежуючих рамок на зображенні під час тренування. Мінімізація цієї втрати під час тренування сприяє покращенню здатності моделі виявляти наявність об'єктів на зображенні;

– `train/cls_loss` – визначає точність виявлення класу об'єкту. В даному випадку спостерігається пряма, що свідчить про те, що клас виявляється без будь-яких помилок, але ця величина не несе ніякої користі у даному випадку, адже клас у даному випадку лише один;

– `metrics/precision` – це метрика, яка вимірює пропорцію правильних позитивних прогнозів (TP - true positives) до всіх позитивних прогнозів (TP + FP - false positives). Розрахунок точності представлений формулою (3.1) [25]:

$$Precision = \frac{TP}{TP+FP}, \quad (3.1)$$

де TP (true positives) - кількість правильно визначених позитивних прикладів (правильно виявлені об'єкти або правильно класифіковані позитивні приклади);

FP (false positives) - кількість помилково визначених позитивних прикладів (неправильно виявлені об'єкти або помилково класифіковані позитивні приклади).

В контексті виявлення об'єктів на зображеннях за допомогою моделі YOLO, точність вказує на те, яка частка об'єктів, визначених моделлю як певний клас, дійсно належать до цього класу;

– `metrics/recall` – відноситься до показника, який вимірює співвідношення правильно виявлених позитивних прикладів (True Positives, TP) до загальної кількості дійсно наявних позитивних прикладів у даних (True Positives + False Negatives, TN). Розрахунок повноти представлений формулою (3.2) [25]:

$$Recall = \frac{TP}{TP+FN}, \quad (3.2)$$

де TP (true positives) – кількість правильно визначених позитивних прикладів (об'єкти, які були правильно виявлені моделлю);

FN (False Negatives) – кількість помилково невизначених позитивних прикладів (об'єкти, які були помилково пропущені моделлю).

У виявленні об'єктів на зображеннях за допомогою моделі YOLO, повнота вказує на те, яка частка дійсно наявних об'єктів певного класу була виявлена моделлю. Висока повнота означає, що модель здатна виявити більше реально присутніх об'єктів, але вона може мати більше помилок FP (False Positives);

– metrics/mAP_0.5 – це показник оцінки продуктивності моделі в задачах об'єктного визначення, особливо коли використовується модель для виявлення об'єктів на зображеннях;

– average Precision (AP) – це середнє значення точності (precision) у всіх порогах відсічення для виявлення об'єктів. Поріг відсічення встановлює наскільки впевнено модель визначає об'єкти.

Коли модель виявляє об'єкти на зображеннях, вона видає впевненість у своїх прогнозах. Поріг відсічення визначає, які прогнози вважаються визначенням (наприклад, які об'єкти вважаються виявленими з певною достовірністю). Чим вище поріг відсічення, тим більш впевненими повинні бути прогнози моделі;

– metrics/mAP_0.5:0.95 – вказує на середнє значення Average Precision для різних класів об'єктів, але враховуючи впевненість моделі в межах порогів відсічення від 0,5 до 0,95. Ця метрика дає більш ретельний огляд продуктивності моделі при виявленні об'єктів з різними рівнями впевненості.

3.4 Постановка експерименту

3.4.1 Перевірка деталі з дефектом

Після навчання моделі нейронної мережі необхідно проєкти експеримент для перевірки правильності роботи запропонованого методу на певній моделі, яка

не була задіяна під час навчання, визначити час спрацювання та оцінити наскільки точно модель визначає дефект.

Для досягнення цієї цілі була розроблена спеціальна деталь з явним дефектом (рис. 3.16).

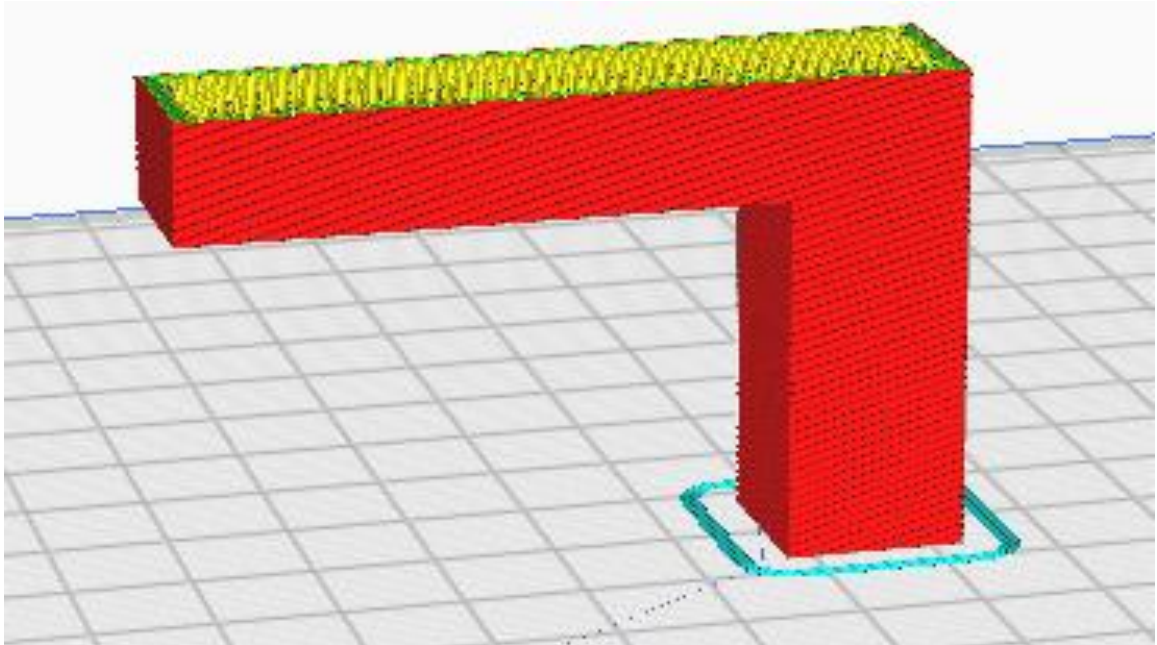


Рисунок 3.16 – Деталь для перевірки системи

Це проста деталь, проте в ній допущена суттєва помилка на етапі підготовки до 3D друку, а саме не розставлені підтримки. Тобто верхня частина деталі почне друкуватися у повітрі, що й спричинить появу дефекту, і, як наслідок, деталь буде зіпсовано. Насправді дану деталь можна розмістити на стіл інакше, щоб підтримки взагалі не знадобились, але для тестування роботи така деталь і її розміщення буде оптимальним, так як дефект з'явиться доволі швидко.

Пластиком для друку виступить PLA. Деталь буде друкуватися при температурі екструдера 215 °C і температурі столу 60 °C.

На рисунку 3.17 показані перші дефекти. На цьому рисунку можна помітити, що верхня частина деталі починає друкуватися у повітрі.

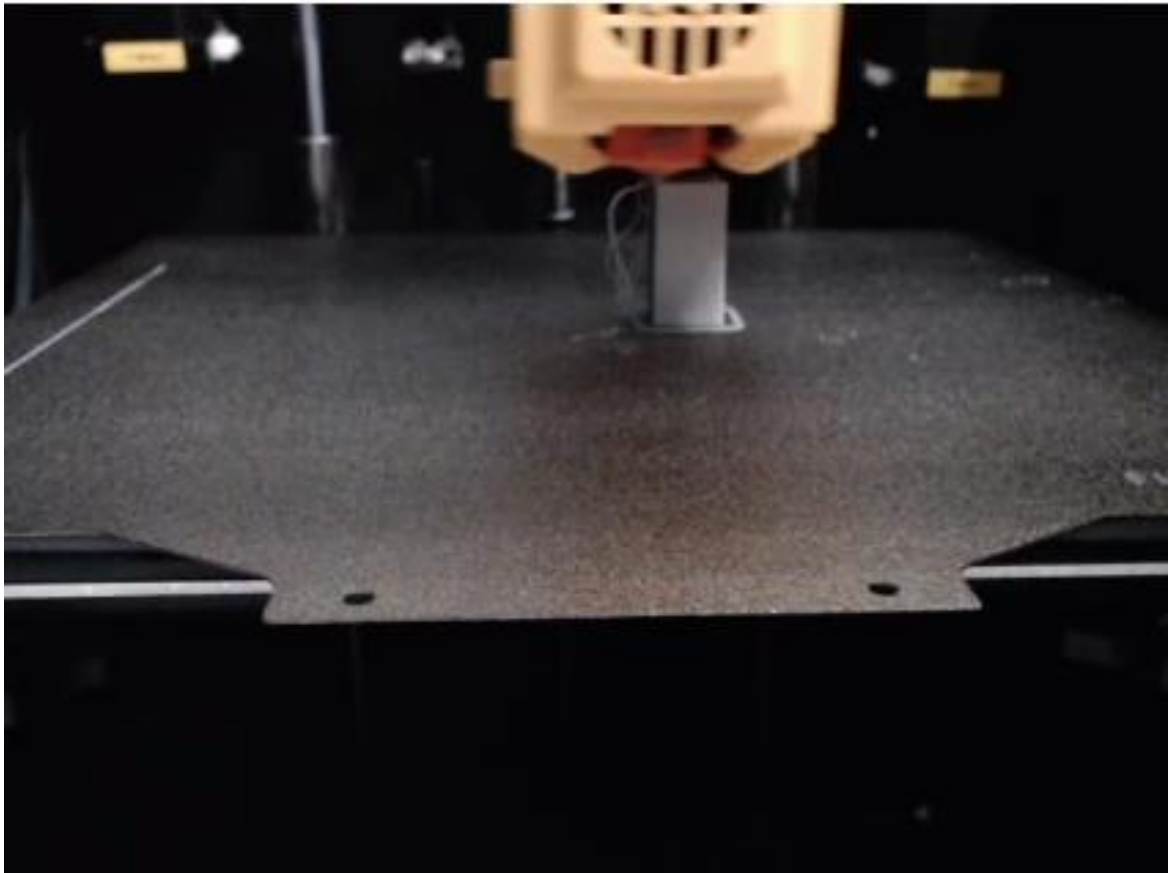


Рисунок 3.17 – Перші дефекти

Через деякий час модель відривається від поверхні, що зображено на рисунку 3.18. Дефект починає розпізнаватися нейромережею з впевненістю 72%. Порогове значення для припинення друку у даний момент встановлено у 70%, при цьому дефект повинен утримуватися в кадрі на протязі 30 секунд. Це зроблено для відкидання хибних спрацювань. Установка порогового значення та умови утримання дефекту в кадрі на протязі 30 секунд допомагає уникнути помилкового припинення друку при тимчасових артефактах або короточасних проблемах. Завдяки умові утримання можна уникнути випадкового припинення друку при невеликих аномаліях, які можуть траплятися в процесі друку.

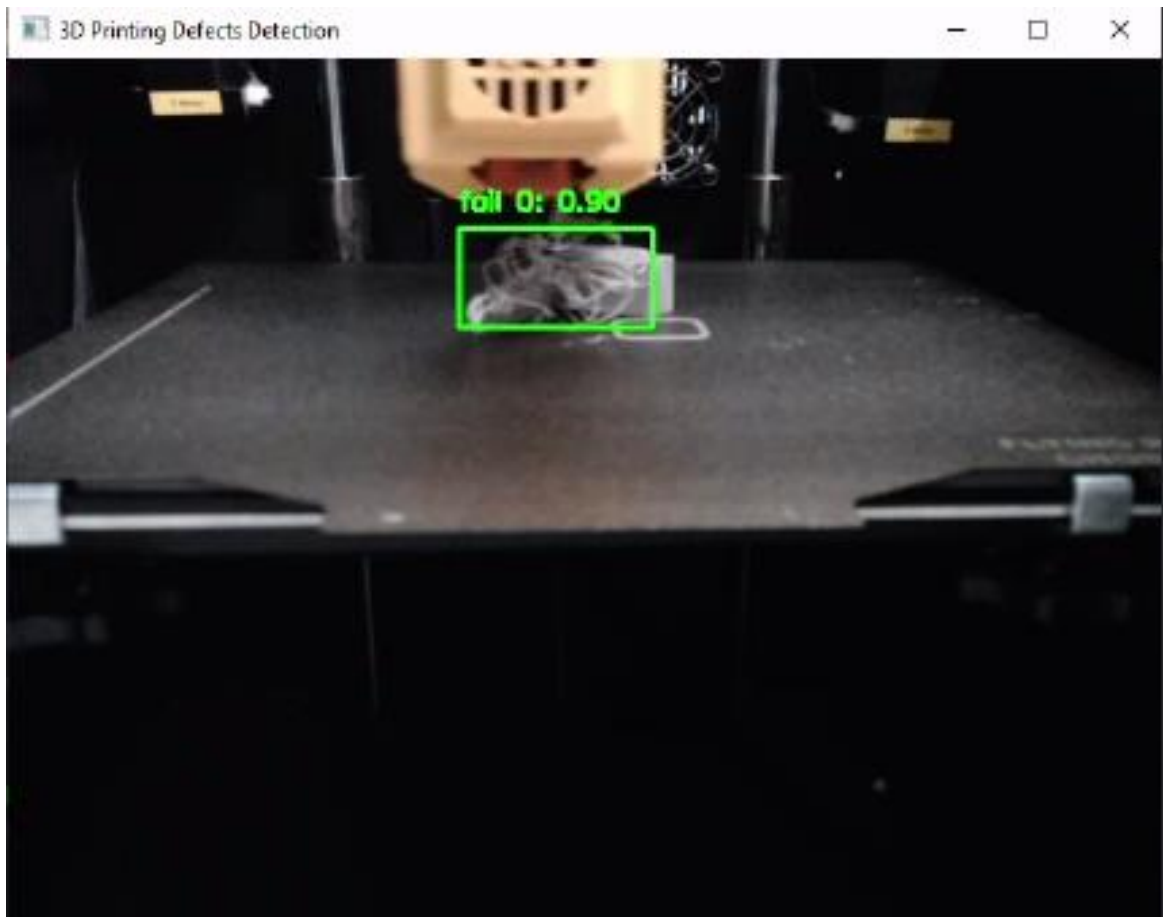


Рисунок 3.20 – Виявлення дефекту із впевненістю в 90%

Після того як дефект визначився, він перейшов перевірку утриманням у кадрі і на принтер було надіслано команду екстреної зупинки друку. Після цього на дисплеї принтера з'явилося повідомлення зображене на рисунку 3.21, були зупинені всі мотори, і принтер перейшов у режим очікування перезавантаження.

Слід зазначити, що під час перевірки системи на даному етапі не було виявлено ніяких помилок. Дефект розпізнавався вірно, хибні спрацювання, через додавання порогу впевненості та порогу затримки були мінімальні і не призводили до зупинки системи через свою появу. Проте, слід зазначити, що такі поняття як відкидання хибних спрацювань та збільшення впевненості моделі можна покращити постійним донавчанням моделі на нових даних.

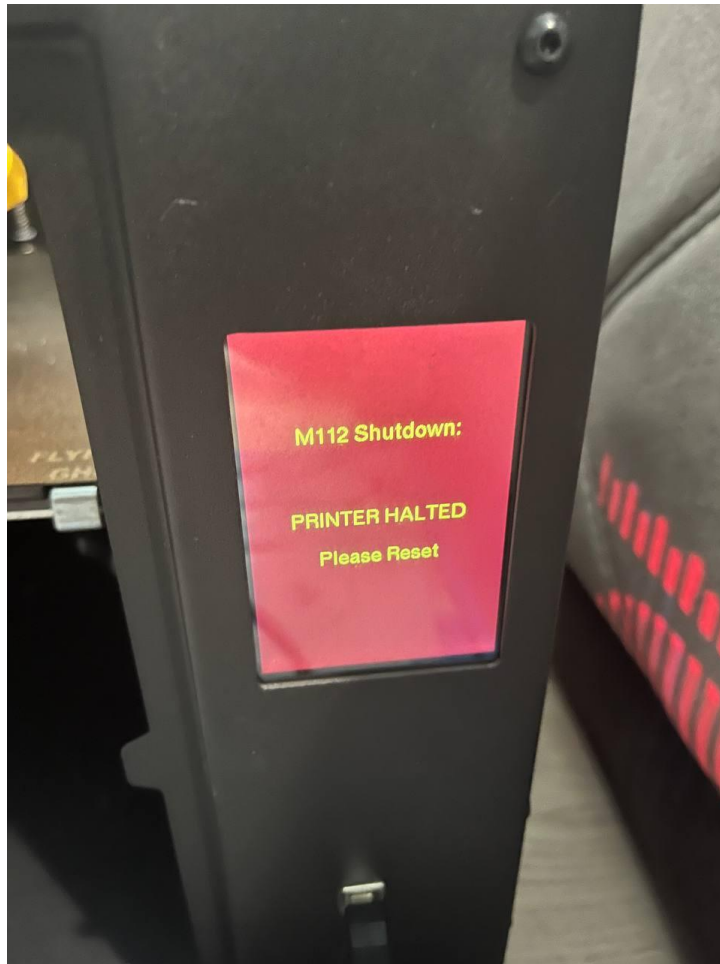


Рисунок 3.21 – Вивід повідомлення про екстрену зупинку друку

3.4.2 Перевірка деталі з відсутністю дефектів

Для того щоб протестувати роботу системи у реальних умовах необхідно не тільки перевіряти деталі, у яких дефект наявний, а й такі які друкуються без дефектів. Це дозволить виключити будь-які помилки на етапі розробки системи і може виключити всі помилки в майбутньому.

Модель, за допомогою якої буде тестуватися система, зображена на рисунку 3.22. Дана модель зазвичай друкується кожен раз, коли необхідно перевірити принтер після калібрування, заміни пластика, підбору температурних параметрів тощо, адже вона має доволі складну геометрію, але друкується доволі швидко.

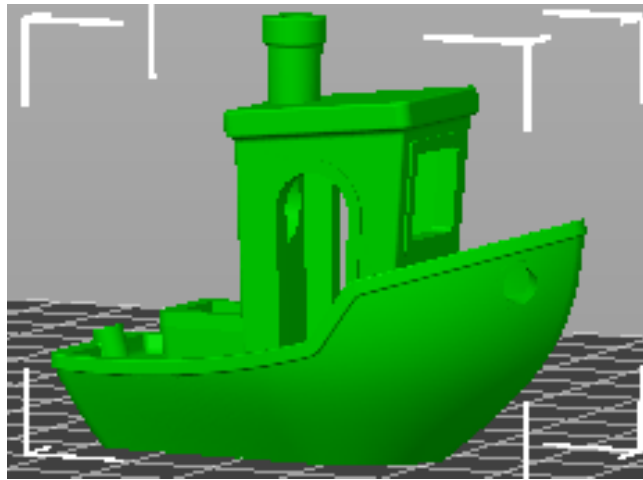


Рисунок 3.22 – Модель без дефектів для тестування

Результати експерименту зображені на рисунках 3.23-3.25.

З цих рисунків видно, що система працює правильно, дефекти не визначаються на різних стадіях друку.

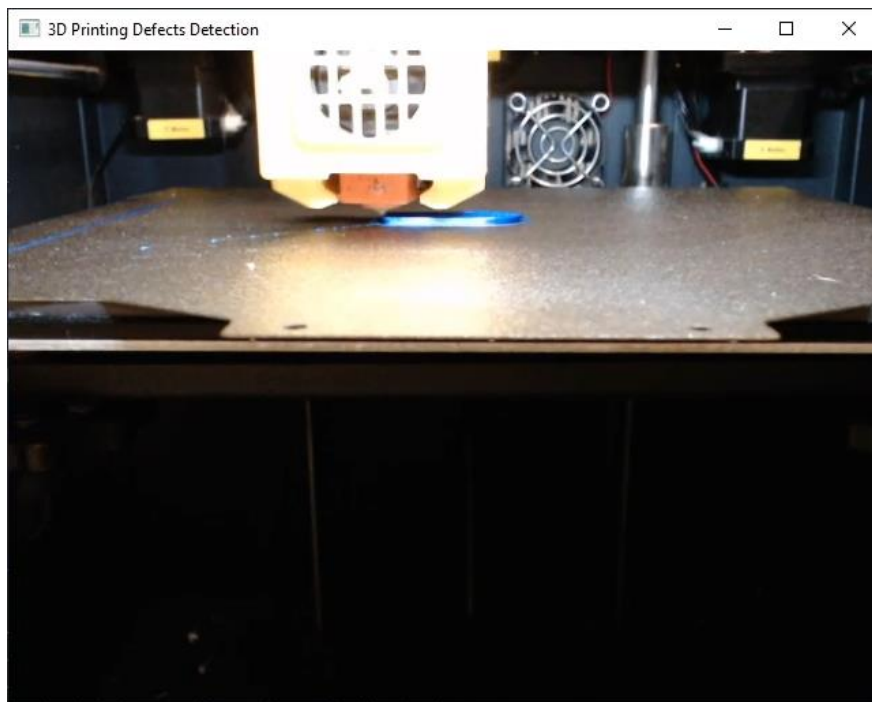


Рисунок 3.23 – Перевірка системи на початковому етапі друку

З даного рисунку видно, що система працює правильно на ранньому етапі друку. Тестова модель не визначається як дефект.

На рисунку 3.24 зображено проміжний етап друку.

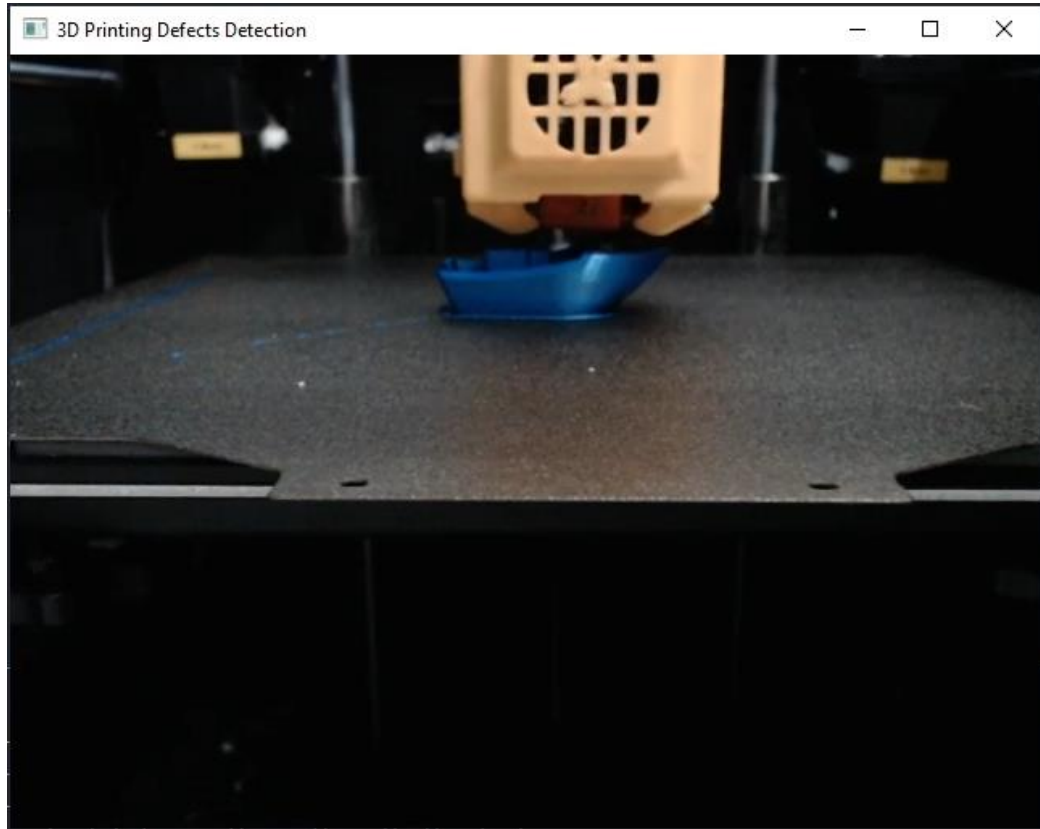


Рисунок 3.24 – Перевірка системи на проміжному етапі друку

З даного рисунку видно, що система поводить себе правильно на проміжному етапі друку. Модель не визначається як дефект.

На рисунку 3.25 зображено кінцевий етап друку моделі, коли друкуюча головка вже від'їхала від моделі. На даному етапі система не видає жодних сповіщень про дефекти. На рисунку 3.26 зображено вивід інформації по завершенню роботи програми.

На рисунку 3.27, як доказ того, що система дійсно працює коректно, на стіл 3D принтера був розташований дефект з попередніх експериментів. Система

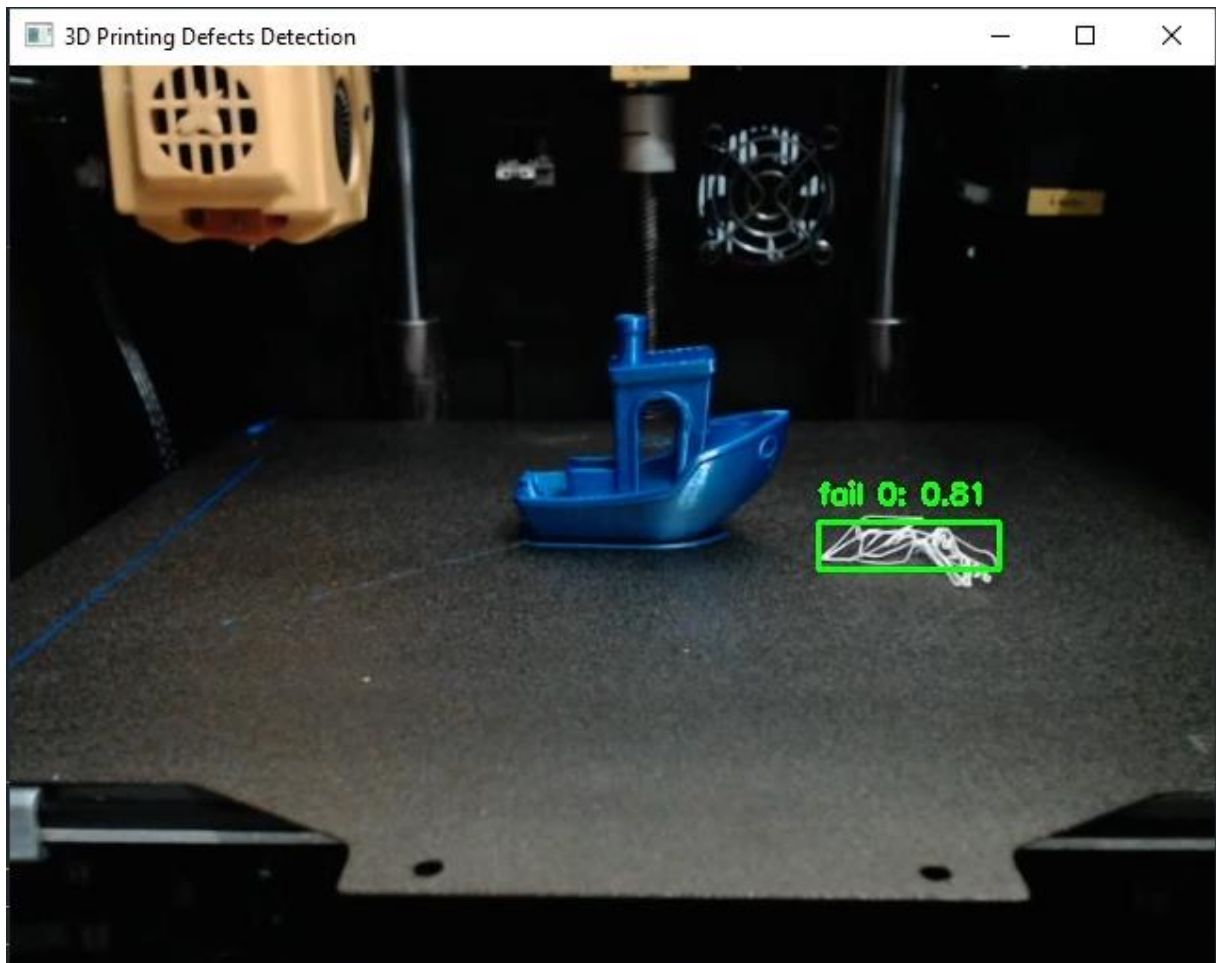


Рисунок 3.27 – Перевірка коректності розпізнавання дефектів

Дана система при перевірці на даній моделі працювала 62 хвилини. За цей час не відбулося жодної зупинки програми. Це є доказом того, що дана система працює стабільно і може використовуватися під час тривалого проміжку часу.

3.5 Охорона праці

Приміщення лабораторії за ступенем небезпеки ураження людей електричним струмом НПАОП 40.1-1-21-98 відноситься до приміщень без підвищеної небезпеки, в яких відсутні умови, що створюють підвищену або особливу небезпеку.

Електропостачання НДЛ здійснюється від трифазної чотирьох мережі з глухозаземленою нейтраллю, струм змінний, частота 50 Гц, напруга 380/220 В. Відповідно до вимог НПАОП 40.1.1-32-01 для забезпечення безпеки необхідно виконати занулення. Час відключення $\leq 0,2$ с.

Для зменшення напруги, прикладеного до тіла людини при випадковому обриві нульового проводу, виконано повторне заземлення нульового провідника.

Проведемо розрахунок занулення в НДЛ.

Визначимо номінальний струм споживання:

$$I_{\text{ном}} = \frac{P \cdot 1000}{U_{\text{М}}} = 9,09 \text{ А}, \quad (3.3)$$

де P – потужність споживання $P = 2 \cdot 10^3$ Вт;

$U_{\text{М}}$ – напруга мережі $U_{\text{М}} = 220$ В.

$$I_{\text{ном}} = \frac{2 \cdot 10^3}{220} = 9,09 \text{ А}. \quad (3.4)$$

З урахуванням запасу вибираємо автоматичний вимикач серії АП50Б з струмом спрацьовування теплового розмикання від $I_p^{\text{тепл}} = 1,6$ А і з струмом спрацьовування електромагнітного розмикача $I_p^{\text{ем}} = 50$ А. Час його спрацьовування не перевищує 0,2 с.

Визначимо очікуване значення струму короткого замикання:

$$I_{\text{кз}}^{\text{оч}} \geq K \cdot I_p^{\text{ем}}, \quad (3.5)$$

де K – коефіцієнт кратності струму, який дорівнює 1,4 для автоматичних вимикачів;

$I_p^{\text{ем}}$ – струм спрацьовування електромагнітного розмикача

$$I_{\text{кз}}^{\text{оч}} = 1,4 \cdot 50 = 70 \text{ А.} \quad (3.6)$$

Зробимо розрахунок перетину алюмінієвого і мідного проводів

$$S_e = \frac{I_{\text{ном}}}{I_{\text{ек}}}, \quad (3.7)$$

де $I_{\text{ном}}$ – розрахунковий струм при нормальній роботі мережі, А;

$I_{\text{ек}}$ – економічна щільність струму, А/мм², яка визначається в залежності від матеріалу і часу використання максимального навантаження.

Отримане перетин провідника округлюють до найбільшого найближчого стандартного перетину.

$$S_e^{\text{Cu}} = \frac{9,09}{3,5} = 2,5 \text{ мм}^2, \quad (3.8)$$

$$S_e^{\text{Al}} = \frac{9,09}{1,9} = 4,78 \text{ мм}^2. \quad (3.9)$$

Знаходимо активний опір повітряної лінії з алюмінієвих провідників:

$$R_{\text{ф}}^{\text{алюм}} = R_{\text{н}}^{\text{алюм}} = p_{\text{алюм}} \cdot \frac{l_n}{S_n}, \quad (3.10)$$

де p – питомий опір провідника, яке для алюмінію одно 0,028 Ом · мм²/м, а для міді 0,018 Ом · мм²/м;

l – довжина провідника, м;

S – переріз провідника, мм²;

$$R_{\phi}^{\text{алюм}} = R_{\text{н}}^{\text{алюм}} = 0,028 \cdot \frac{25}{4} = 0,175 \text{ Ом.} \quad (3.11)$$

Знаходимо активний опір кабельної лінії з мідних провідників:

$$R_{\phi}^{\text{мед}} = R_{\text{н}}^{\text{мед}} = p_{\text{мед}} \cdot \frac{l_n}{S_n}, \quad (3.12)$$

$$R_{\phi}^{\text{мед}} = R_{\text{н}}^{\text{мед}} = 0,018 \cdot \frac{96}{1,5} = 1,152 \text{ Ом.} \quad (3.13)$$

Знаходимо довжину петлі «фаза-нуль»:

$$I_{\text{н}} = I_{\phi}^{\text{алюм}} + I_{\phi}^{\text{мед}} + I_{\text{н}}^{\text{алюм}} + I_{\text{н}}^{\text{мед}}. \quad (3.14)$$

.

Повний опір петлі «фаза-нуль»:

$$Z_{\text{н}} = R_{\phi}^{\text{алюм}} + R_{\text{н}}^{\text{алюм}} + R_{\phi}^{\text{мед}} + R_{\text{н}}^{\text{мед}}, \quad (3.15)$$

$$Z_{\text{н}} = 1,152 + 1,152 + 0,175 + 0,175 = 2,65 \text{ Ом.} \quad (3.16)$$

Потужність живильного трансформатора $P_{\text{тр}} = 160 \text{ ВТА}$, отже його опір дорівнюватиме $Z_{\text{тр}} = 0,047 \text{ Ом}$.

Струм однофазного короткого замикання буде дорівнювати:

$$I_{\text{кз}} = \frac{U_{\phi}}{\left(\frac{Z_{\text{тр}}}{3}\right) + Z_{\text{н}}}, \quad (3.17)$$

$$I_{\text{кз}} = \frac{220}{\left(\frac{0,047}{3}\right) + 2,65} = 82,7 \text{ А.} \quad (3.18)$$

$70 \leq 82,7$ – умова виконується, отже при виникненні короткого замикання всі ПЕОМ та принтера будуть відключені.

Для зменшення напруги, прикладеного до тіла людини при пробі фазі на корпус ПЕОМ або принтера, необхідно виконати повторне заземлення нейтралі. Для цього з'єднані між собою корпусу ПЕОМ та принтера слід заземлити, використовуючи природні і штучні заземлювачі. Опір повторного заземлення не повинен перевищувати 10 Ом.

3.6 Висновки до третього розділу

У даному розділі було розроблено програмне забезпечення, з подальшою інтеграцією моделі нейронної мережі.

Було проведено навчання моделі. Навчання відбувалося у декілька етапів. На початку було обрано готовий набір даних який містив 3000 зображень з різними типами дефектів і на якому були зображені різні 3D принтери з різним розташуванням камер. Перший етап навчання був невдалим, так як було використано малу кількість епох для навчання моделі. Впевненість даної моделі була низька і через це було вирішено збільшити кількість епох навчання до 300 епох. Після навчання моделі її було протестовано і виявлено такий недолік як хибні спрацювання та недостатня впевненість моделі у тому, що перед нею знаходиться дефект. Було вирішено, що дані помилки виникають унаслідок того, що в готовому наборі даних не міститься зображення принтера, який використовувався у макеті, а також ракурс камери був не такий, як у даному макеті. Як наслідок, було вирішено провести донавчання моделі на власному наборі даних. Для цього було записано декілька відеозаписів і розбито їх на зображення покадрово. Після чого дані зображення були розмічені. Модель на даному етапі в якості ваг для навчання використовувала вже власні ваги, які були

отримані на попередньому етапі. Після донавчання моделі було перевірено її роботу.

Перевірка роботи моделі складалась з перевірки на деталі з вимушеним дефектом, яка була розроблена спеціально для тестування і перевірки на деталі без дефектів. Обидві перевірки дана система пройшла успішно. Впевненість моделі під час виявлення дефектів була доволі висока і складала 90%. Під час другої перевірки друкувалася деталь, що не містить дефектів. Перевірка пройшла успішно, хибних спрацювань не було виявлено, система працювала стабільно та доволі тривалий час, що може свідчити про її надійність.

ВИСНОВКИ

Системи контролю, в наш час, є невід'ємними частинами виробництв, так як з їх допомогою можна зменшувати кількість браку, зводити до мінімуму роботу людини в якості контролера і як наслідок зменшувати затрати на виробництва.

Системи контролю 3D друкованих деталей не настільки розповсюджені, хоча вони можуть принести численні переваги в контексті 3D друку. З їхньою допомогою можна контролювати процес друку та відстежувати дефекти у реальному часі.

У результаті виконання першого розділу було проаналізовано системи контролю. Особливу увагу було приділено системам контролю з допомогою машинного зору і її методам. Після чого було вирішено розробити систему з використанням нейронної мережі.

У результаті виконання другого розділу розроблено структурну схему проекту, на якій було візуалізовано всі блоки та зв'язки які використовуються у макеті. Виконано підбір компонентів макету. Розроблено алгоритм роботи програми, а також розглянуто алгоритм роботи моделі нейронної мережі YOLO. Також розроблено підпрограму для зупинки друку.

У результаті виконання третього розділу розроблено програму для розпізнавання дефектів та зупинки друку з допомогою навченої моделі нейронної мережі. Підготовлено набір даних для навчання моделі нейронної мережі. Проведено навчання моделі YOLO у декілька етапів. Після чого систему було протестовано у різних ситуаціях. Модель успішно пройшла всі етапи тестування та виявилася стабільною у роботі, тому можна стверджувати, що її можна інтегрувати у виробництво друкованих деталей задля забезпечення контролю.

На сьогодні є доволі велика кількість 3D студій, які пропонують виготовлення деталей за допомогою технології FDM 3D друку. Дані студії мають великий парк 3D принтерів, задля забезпечення швидкого процесу друку.

Обслуговування всієї техніки, заміна філаменту, запуск друку та контроль відбувається за допомогою оператора. Так як кількість принтерів може бути дуже великою, контролювати процес друку на кожному принтері буде дуже важкою задачею для людини.

Розроблену СК можна інтегрувати у дане виробництво, що з самого початку впровадження замінить оператора в якості контролера повністю. Дана система зменшить відсоток браку, так як на відміну від людини вона може слідувати за процесом друку без зупинок.

ПЕРЕЛІК ПОСИЛАНЬ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.
2. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. – Харків: ХНУРЕ, 2021. – 55 с.
3. Невлюдов І.Ш. Технічні засоби автоматизації: Підручник / І.Ш. Невлюдов, А.О. Андрусевич, О.І. Филипенко, Н.П. Демська, С.П. Новоселов. – Кривий Ріг : Криворізький коледж НАУ, 2019. 366 с.
4. Гриценко Є.М. Аналіз систем контролю виготовлення 3D деталей на потоковому роботизованому виробництві / Автоматизація та Приладобудування («Automation and Development of Electronic Devices» ADED-2023) [Електронний ресурс] : збірник студентських наукових статей / Харківський національний університет радіоелектроніки ; [редкол.: І.Ш. Невлюдов та ін.]. – Харків : ХНУРЕ, 2023. – Вип. 2. – С. 309-318.
5. Савінова В. В. Застосування методів комп'ютерного зору в автомобільній індустрії / В. В. Савінова, В. О. Колесніков // Проблеми та перспективи розвитку автомобільного транспорту : Матеріали V-ї Міжнародної наук.-технічної інтернетконференції (13-14 квітня 2017 р., м. Вінниця). – Вінниця, 2017. – С. 113 -120. – Режим доступу: <http://atmconf.vntu.edu.ua/materialy2017.pdf>.

6. OpenCV Forum [Электронный ресурс] /– Режим доступа: [www](http://www.opencv.org/) / URL: <https://forum.opencv.org/>.
7. Stack overflow [Электронный ресурс] /– Режим доступа: [www](http://www.stackoverflow.com/) / URL: <https://stackoverflow.com/>.
8. Introduction to ORB (Oriented FAST and Rotated BRIEF) [Электронный ресурс] /– Режим доступа: [www](http://www.medium.com/@deepanshut041/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf) / URL: <https://medium.com/@deepanshut041/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf>.
9. Stack overflow. People detection with haar cascade [Электронный ресурс] /– Режим доступа: [www](http://www.stackoverflow.com/questions/29100541/people-detection-with-haar-cascade) / URL: <https://stackoverflow.com/questions/29100541/people-detection-with-haar-cascade>.
10. Scholarpedia [Электронный ресурс] /– Режим доступа: [www](http://www.scholarpedia.org/) / URL: [doi:10.4249/scholarpedia.10491](https://doi.org/10.4249/scholarpedia.10491).
11. An Analysis and Implementation of the Harris Corner Detector [Электронный ресурс] /– Режим доступа: [www](http://www.doi.org/10.5201/ipol.2018.229) / URL: <https://doi.org/10.5201/ipol.2018.229>.
12. Slyusar V. Neural Networks Models based on the tensor-matrix theory. Problems of the development of promising micro– and nanoelectronic systems [Электронный ресурс] /– Режим доступа: [www](http://www.10.31114/2078-7707-2021-2-23-28) / URL: <https://10.31114/2078-7707-2021-2-23-28>.
13. Matusugu, Masakazu; Katsuhiko Mori; Yusuke Mitari; Yuji Kaneda (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network [Электронный ресурс] /– Режим доступа: [www](http://www.doi.org/10.1016/S0893-6080(03)00115-1) / URL: [doi:10.1016/S0893-6080\(03\)00115-1](https://doi.org/10.1016/S0893-6080(03)00115-1).
14. van den Oord, Aaron; Dieleman, Sander; Schrauwen, Benjamin. У Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; Weinberger, K. Q. [Электронный ресурс] /– Режим доступа: [www](http://www.DeepContentBasedMusicRecommendation.com) / URL: Deep content-based music recommendation. Curran Associates, Inc.

15. Collobert, Ronan; Weston, Jason [Электронный ресурс] /– Режим доступа: [www / URL](http://www.url.com): A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. Proceedings of the 25th International Conference on Machine Learning.

16. Rozetka [Электронный ресурс] /– Режим доступа: [www / URL](http://www.url.com): https://rozetka.com.ua/ua/358446246/p358446246/?gad_source=1&gclid=Cj0KCQiAhomtBhDgARIsABcaYynUyjYPQCVRonZCmjQmZIk19TSRLVdSvtiHpVEYoHDUkoaaZrmsg5saAviaEALw_wcB.

17. Flying Bear [Электронный ресурс] /– Режим доступа: [www / URL](http://www.url.com): <https://3dflyingbear.com/products/flying-bear-ghost-6-fdm-3d-printer>.

18. Jianxin Wu. Introduction to Convolutional Neural Networks. [Электронный ресурс] /– Режим доступа: [www / URL](http://www.url.com): <https://cs.nju.edu.cn/wujx/paper/CNN.pdf>.

19. Michael E. Sander, Pierre Ablin, Mathieu Blondel, Gabriel Peyre'. Momentum Residual Neural Networks [Электронный ресурс] /– Режим доступа: [www / URL](http://www.url.com): <https://proceedings.mlr.press/v139/sander21a.html>.

20. Akshay Kapoor, Rishi Shah, Rushi Bhuva, Tejas Pandit. Understanding inception network architecture for image classification [Электронный ресурс] /– Режим доступа: [www / URL](http://www.url.com): https://www.researchgate.net/profile/Akshay-Kapoor-4/publication/339445159_UNDERSTANDING_INCEPTION_NETWORK_ARCHITECTURE_FOR_IMAGE_CLASSIFICATION/links/5e532ddaa6fdcc2f8f5d72a6/UNDERSTANDING-INCEPTION-NETWORK-ARCHITECTURE-FOR-IMAGE-CLASSIFICATION.pdf.

21. Amazon Web Services | What is RNN? [Электронный ресурс] /– Режим доступа: [www / URL](http://www.url.com): <https://aws.amazon.com/what-is/recurrent-neural-network/>.

22. Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, Sivanesan Sangeetha. AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing [Электронный ресурс] /– Режим доступа: [www / URL](http://www.url.com): <https://arxiv.org/abs/2108.05542>.

23. Accurate and automated detection of surface knots on sawn timbers using YOLO-V5 model [Электронный ресурс] /– Режим доступа: [www / URL: https://www.proquest.com/openview/4bb248d49cd5f553f8bb8f9b9bb05e18/1?pq-origsite=gscholar&cbl=5038271](http://www.proquest.com/openview/4bb248d49cd5f553f8bb8f9b9bb05e18/1?pq-origsite=gscholar&cbl=5038271).

24. Gcode | Marlin Firmware [Электронный ресурс] /– Режим доступа: [www / URL: https://marlinfw.org/meta/gcode/](http://www.marlinfw.org/meta/gcode/).

25. mAP (mean Average Precision) for Object Detection [Электронный ресурс] /– Режим доступа: [www / URL: https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173](https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173).